Volume 40 Number 2 June 2016 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Special Issue:

SoICT 2015

Guest Editors:

Luc De Raedt
Yves Deville
Marc Bui
Dieu-Linh Truong

1977

Editorial Boards

Informatica is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Higher Education, Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

Editors' Introduction to the Special Issue on "The Sixth International Symposium on Information and Communication Technology – SoICT 2015"

Editors' Introduction to the Special Issue on the Sixth International Symposium on Information and Communication Technology (SoICT 2015, Hue City, Vietnam)

This Special consists of a selection of the best papers from the 6th International Symposium on Information and Communication Technology - SoICT 2015. Since 2010, SoICT has been organised annually. The symposium provided an academic forum for researchers to share their latest research findings and to identify future challenges in computer science. In 2015, SoICT was held in Hue Royal city, Vietnam, during December 3-4th, 2015. SoICT 2015 was an international symposium that covered four major areas of research including Artificial Intelligence and Big Data, Network and Security, Human-Computer Interaction, Software Engineering and Applied Computing.

In 122 submissions from 18 countries, 49 papers were accepted for presentation at SoICT'2015. Among them, 4 papers were carefully selected, after further extension and additional review, for inclusion in this Special Issue.

The first paper by N.H.T. Dang, S. Dvoenko and S. Dinh "A Mixed Noise Removal Method Based on Total Variation" is about removing noise from biomedical images. The paper proposes a novel method to remove mixed noise using the idea of the total variation of an image intensity (brightness) function.

The second paper by Q.V. Bui, K. Sayadi, M. Bui "A Multi-Criteria Document Clustering Method Based on Topic Modeling and Pseudoclosure Function" addresses the problem of document clustering. The paper proposes a novel unsupervised clustering method based on the structural analysis of the latent semantic space. The authors perform a structural analysis on the latent semantic space using the Pretopology theory that allows them to investigate the role of the number of clusters and the chosen centroids, in the similarity between the computed clusters. Their method has been applied to and evaluated on Twitter data.

The third paper by K.V. Nguyen, P.L. Nguyen, H. Phan is entitled "A Distributed Algorithm For Monitoring An Expanding Hole In Wireless Sensor Networks". Holes in sensor networks are regions that have no operating nodes and may occur due to several reasons, including cases caused by natural obstacles or disaster suffered areas. Determining the location and shape of holes can help to make smart, early routing decisions for circumventing a hole. Many hole determination algorithms have proposed in the literature, however, they consider mainly networks with static holes i.e. with stable boundary nodes. Moreover, most of these are designed in a centralized manner, which is not suitable to the unstable situation of networks with an expanding hole. This paper proposes an algorithmic scheme not only for determining the initial shape but also for monitoring and quickly reporting about the area of a hole gradually expanding.

The last paper by D.L.Truong, E. Ouro and T.C. Nguyen "Protected Elastic-tree topology for Survivable and Energy-efficient Data Center" starts from the observation that data centers currently consume too much energy.

They propose to reduce energy consumption by turning off certain switches in data centers, and study the effect on the connectivity of the network and data center. They then propose to tailor the topology using a path protection method to ensure that all connections in the data center retain survivability upon any single failure.

Luc De Raedt
Yves Deville
Marc Bui
Dieu-Linh Truong

A Mixed Noise Removal Method Based on Total Variation

Dang N.H. Thanh
Tula State University; 92 Lenin Ave., Tula, Russian Federation
Hue College of Industry, 70 Nguyen Hue st., Hue, Vietnam
E-mail: dnhthanh@hueic.edu.vn

Dvoenko Sergey D.
Tula State University; 92 Lenin Ave., Tula, Russian Federation
E-mail: dsd@tsu.tula.ru

Dinh Viet Sang
Hanoi University of Science and Technology; 1 Dai Co Viet st., Hanoi, Vietnam
E-mail: sangdv@soict.hust.edu.vn

Keywords: total variation, ROF model, gaussian noise, poisson noise, image processing, biomedical image, euler-lagrange equation

Received: March 24, 2016

Due to the technology limits, digital images always include some defects, such as noise. Noise reduces image quality and affects the result of image processing. While in most cases, noise has Gaussian distribution, in biomedical images, noise is usually a combination of Poisson and Gaussian noises. This combination is naturally considered as a superposition of Gaussian noise over Poisson noise. In this paper, we propose a method to remove such a type of mixed noise based on a novel approach: we consider the superposition of noises like a linear combination. We use the idea of the total variation of an image intensity (brightness) function to remove this combination of noises.

Povzetek: Članek predlaga izvirno kombinacijo Gaussovega in Poissonovega filtra za filtriranje šuma v slikah.

1 Introduction

Image denoising has attracted a lot of attention in recent years. In order to suppress noise effectively, we need to know its type. There are many types of noises, for example, Gaussian (digital images), Poisson (X-Ray images), Speckle (ultra sonograms) noises and so on.

One of the most famous effective methods is the total variation model [2-4, 10, 12, 17, 18, 22, 26]. The first person who suggested it to solve the denoising problem is Rudin [17]. He used the total variation as a universal tool in image processing. His denoising model is well-known as the ROF model [3, 17]. The ROF model is targeted to efficiently remove Gaussian noise only.

This model is often used to remove not only Gaussian noise, but also other types of noise. For example, the ROF model suppresses Poisson noise not so effectively. Le T. [9] proposed another model, well-known as the modified ROF model to remove Poisson noise only.

Gaussian and Poisson noises both are widespread in real situations, but their combination is important too, for example, in electronic microscopy images [7, 8]. In these images, both types of noises are combined as a superposition. In physical process, Poisson noise usually is added first, before Gaussian noise. Luisier F. with co-authors proposed the theoretically strong PURE-LET method [11] (Poisson-Gaussian unbiased risk estimate – linear expansion of thresholds) to remove this type of combination of noises.

However, such kind of noises usually can be considered as dependent on the image acquisition systems. At the same time, in many papers devoted to the image denoising problem the idea of Poisson-Gaussian noise combination is considered, even though such is not the case.

From other side, many noise reduction approaches have been developed, particularly, wavelet-based transforms, etc. It needs to draw attention, noise reduction approaches that have been developed based on wavelet transform are only for Gaussian or Poisson noise.

In order to remove mixed noise, let us assume that the superposition of noises can be equivalent to some unknown linear combination of them.

We can combine ROF and modified ROF models to suppress the linear combination of noises. The obtained model is supposed to remove the mixed noise better than ROF or modified ROF models separately. Additionally, it is simpler than PURE-LET, because we try to find only the proportion between Poisson and Gaussian noises in the mixed noise.

In experiments, we use images and add noise into them. The image quality is compared with some other denoising methods such as ROF, modified ROF models, and PURE-LET method to remove the superposition of noises. In our paper [19], we proposed to remove the linear combination of Poisson and Gaussian noises and compared results with Wiener [1] and median [23] filters, and with Beltrami method [29]. Our method gives better results for Gaussian and Poisson noises separately, and for the combination of noises too. Hence, in this paper, we do not compare our approach with these methods.

In order to compare image quality after restoration, we use criteria PSNR (Peak Signal-to-Noise Ratio), MSE (Mean Square Error), SSIM (Structure SIMilarity) [24, 25]. The PSNR criterion is the most important, because it is always used to evaluate images and signals quality.

In this paper, we try to represent and discuss only the case limited by the greyscale artificial and real images with artificial noise. According to it, we can use only criteria above based on the full-reference image quality evaluation approach.

In the case of greyscale real images with unknown noises, we need to use the no-reference approach to evaluate the quality of denoising. In general, it is complicated theoretical problem to develop a criterion for it.

Our investigation based on BRISQUE criterion [13] (Blind/Referenceless Image Spatial QUality Evaluator) in this case was discussed in paper [20].

2 Combined denoising model

Let in R^2 space a bounded domain $\Omega \subset \mathrm{R}^2$ be given. Let functions $u(x,y) \in \mathrm{R}$ and $v(x,y) \in \mathrm{R}$, respectively, be ideal (without noise) and observed (noisy) images, $(x,y) \in \Omega$. For smooth function u, its total variation can be defined by $V_T[u] = \int\limits_\Omega |\nabla u|\, dxdy$, where $\nabla u = (u_x, u_y)$ is a gradient, $u_x = \partial u / \partial x$, $u_y = \partial u / \partial y$, $|\nabla u| = \sqrt{u_x^2 + u_y^2}$. In this paper, we consider that the function u always has limited total variation $V_T[u] < \infty$.

According to [2, 3, 17, 18], an image smoothness is characterized by the total variation of an image intensity function. The total variation of the noisy image is always greater than the total variation of the corresponding smooth image. In order to solve the problem $V_T[u] \to \min$, we need to use the following condition

$$\int\limits_\Omega (v-u)^2\, dxdy = const\,.$$

Hence, we obtain the ROF model to remove Gaussian noise in the image [17, 18]:

$$u^* = \arg\min_u \left(\int\limits_\Omega |\nabla u|\, dxdy + \frac{\lambda}{2} \int\limits_\Omega (v-u)^2\, dxdy \right),$$

where $\lambda > 0$ is Lagrange multiplier. This is a solution of the unconstrained optimization problem.

In order to remove Poisson noise, Le T. built another model based on ROF model [9] as the optimization problem $V_T[u] \to \min$ with the following constraint

$$\int\limits_\Omega \ln(p(v\,|\,u))\, dxdy = \int\limits_\Omega (u - v\ln(u))\, dxdy = const\,.$$

This model resulted in the following unconstrained optimization problem

$$u^* = \arg\min_u \left(\int\limits_\Omega |\nabla u|\, dxdy + \beta \int\limits_\Omega (u - v\ln(u))\, dxdy \right),$$

where $\beta > 0$ is a coefficient of regularization. This is the known modified ROF model to remove Poisson noise.

In order to build a model for removing the mixed Poisson-Gaussian noise, we also solve the same optimization problem $V_T[u] \to \min$, but with a different constraint as follows.

This constraint is very similar to constraints above. We consider, the noise variance is unchangeable (Poisson noise is not changed and Gaussian noise only depends on noise variance):

$$\int\limits_\Omega \ln(p(v\,|\,u))\, dxdy = const\,, \qquad (1)$$

where $p(v\,|\,u)$ is a conditional probability of the real image v with the ideal image u given.

The probability density function of Gaussian noise is

$$p_1(v\,|\,u) = \exp\left(-\frac{(v-u)^2}{2\sigma^2} \right) / (\sigma\sqrt{2\pi})\,,$$

and the probability distribution of Poisson noise is

$$p_2(v\,|\,u) = \exp(-u)u^v / v!\,.$$

We have to notice that intensity functions of images u and v are integer (for example, for 8-bits greyscale image the range of intensity is from 0 to 255).

In order to combine Gaussian and Poisson noises, we consider the following linear combination

$$\ln(p(v\,|\,u)) = \lambda_1 \ln(p_1(v\,|\,u)) + \lambda_2 \ln(p_2(v\,|\,u))\,,$$
$$\lambda_1 > 0,\ \lambda_2 > 0,\ \lambda_1 + \lambda_2 = 1\,.$$

According to (1), we obtain the denoising problem as a constrained optimization problem

$$\begin{cases} u^* = \arg\min\limits_u \int\limits_\Omega |\nabla u|\, dxdy \\ \int\limits_\Omega \left(\dfrac{\lambda_1}{2\sigma^2}(v-u)^2 + \lambda_2 (u - v\ln(u)) \right) dxdy = \kappa\,, \end{cases}$$

where κ is a constant value. We transform this problem into unconstrained optimization problem by using Lagrange functional

$$L(u,\tau) = \int\limits_\Omega |\nabla u|\, dxdy + \tau \left(\frac{\lambda_1}{2\sigma^2} \int\limits_\Omega (v-u)^2\, dxdy + \right.$$
$$\left. \lambda_2 \int\limits_\Omega (u - v\ln(u))\, dxdy - \kappa \right)$$

to find the solution as

$$(u^*, \tau^*) = \arg\min_{u,\,\tau} L(u,\tau) \qquad (2)$$

where $\tau > 0$ is Lagrange multiplier.

If $\lambda_1 = 0$ and $\beta = \lambda_2 \tau$, we obtain the modified ROF model to remove Poisson noise. If $\lambda_2 = 0$ and $\lambda = \lambda_1 \tau / \sigma^2$, we obtain the ROF model to remove Gaussian noise. If $\lambda_1 > 0, \lambda_2 > 0$, we obtain our model to remove mixed Poisson-Gaussian noise.

3 Discrete denoising model

The problem (2) can be solved by using Lagrange multipliers method [5, 16, 28].

We use Euler-Lagrange equation [28]. Let a function $f(x,y)$ be defined in a limited domain $\Omega \subset \mathrm{R}^2$ and be second-order continuously differentiated by x and y, where $(x,y)\in\Omega$. Let $F(x,y,f,f_x,f_y)$ be a convex functional, where $f_x = \partial f / \partial x$, $f_y = \partial f / \partial y$. Then the solution of the following optimization problem

$$\int_{\Omega} F(x,y,f,f_x,f_y)dxdy \to \min$$

satisfies the following Euler-Lagrange equation

$$F_f(x,y,f,f_x,f_y) - \frac{\partial}{\partial x}F_{f_x}(x,y,f,f_x,f_y) - \frac{\partial}{\partial y}F_{f_y}(x,y,f,f_x,f_y) = 0,$$

where $F_f = \partial F / \partial f$, $F_{f_x} = \partial F / \partial f_x$, $F_{f_y} = \partial F / \partial f_y$.

We use the above result to solve the obtained model. Then the solution of the problem (2) satisfies the following Euler-Lagrange equation

$$-\frac{\lambda_1}{\sigma^2}(v-u) + \lambda_2(1-\frac{v}{u}) - \mu\frac{\partial}{\partial x}\left(\frac{u_x}{\sqrt{u_x^2+u_y^2}}\right) - \mu\frac{\partial}{\partial y}\left(\frac{u_y}{\sqrt{u_x^2+u_y^2}}\right) = 0, \qquad (3)$$

where $\mu = 1/\tau$. We rewrite (3) in the form

$$\frac{\lambda_1}{\sigma^2}(v-u) - \lambda_2(1-\frac{v}{u}) + \mu\frac{u_{xx}u_y^2 - 2u_xu_yu_{xy} + u_x^2u_{yy}}{(u_x^2+u_y^2)^{3/2}} = 0 \qquad (4)$$

$$u_{xx} = \frac{\partial^2 u}{\partial x^2}, \quad u_{yy} = \frac{\partial^2 u}{\partial y^2}, \quad u_{xy} = \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial y}\right) = \frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x}\right) = u_{yx}.$$

In order to obtain the discrete form of the model (4), we add an artificial time parameter and consider the function $u = u(x,y,t)$ in the following diffusion equation

$$u_t = \frac{\partial u}{\partial t} = \frac{\lambda_1}{\sigma^2}(v-u) - \lambda_2(1-\frac{v}{u}) + \mu\frac{u_{xx}u_y^2 - 2u_xu_yu_{xy} + u_x^2u_{yy}}{(u_x^2+u_y^2)^{3/2}}. \qquad (5)$$

Then the discrete form of the equation (5) is

$$u_{ij}^{k+1} = u_{ij}^k + \xi\left(\frac{\lambda_1}{\sigma^2}(v_{ij} - u_{ij}^k) - \lambda_2(1-\frac{v_{ij}}{u_{ij}^k}) + \mu\varphi_{ij}^k\right), \qquad (6)$$

$$\varphi_{ij}^k = \frac{\nabla_{xx}(u_{ij}^k)(\nabla_y(u_{ij}^k))^2}{((\nabla_x(u_{ij}^k))^2 + (\nabla_y(u_{ij}^k))^2)^{3/2}} + \frac{-2\nabla_x(u_{ij}^k)\nabla_y(u_{ij}^k)\nabla_{xy}(u_{ij}^k) + (\nabla_x(u_{ij}^k))^2\nabla_{yy}(u_{ij}^k)}{((\nabla_x(u_{ij}^k))^2 + (\nabla_y(u_{ij}^k))^2)^{3/2}},$$

$$\nabla_x(u_{ij}^k) = \frac{u_{i+1,j}^k - u_{i-1,j}^k}{2\Delta x},$$

$$\nabla_y(u_{ij}^k) = \frac{u_{i,j+1}^k - u_{i,j-1}^k}{2\Delta y}, \nabla_{xx}(u_{ij}^k) = \frac{u_{i+1,j}^k - 2u_{ij}^k + u_{i-1,j}^k}{(\Delta x)^2},$$

$$\nabla_{yy}(u_{ij}^k) = \frac{u_{i,j+1}^k - 2u_{ij}^k + u_{i,j-1}^k}{(\Delta y)^2},$$

$$\nabla_{xy}(u_{ij}^k) = \frac{u_{i+1,j+1}^k - u_{i+1,j-1}^k - u_{i-1,j+1}^k + u_{i-1,j-1}^k}{4\Delta x\Delta y},$$

$$u_{0j}^k = u_{1j}^k;\ u_{N_1+1,j}^k = u_{N_1,j}^k;\ u_{i0}^k = u_{i1}^k;\ u_{i,N_2+1}^k = u_{i,N_2}^k;$$

$$i = 1,\dots,N_1;\ j = 1,\dots,N_2;$$

$$k = 0,1,\dots,K;\ \Delta x = \Delta y = 1;\ 0 < \xi < 1,$$

where K is enough great number, $K = 500$.

4 Optimal model parameters

In practice, parameters $\lambda_1,\lambda_2,\mu,\sigma$ in procedure (6) are usually unknown. We have to change λ_1,λ_2,μ into $\lambda_1^k,\lambda_2^k,\mu^k$ to evaluate them on every iteration step k.

4.1 Optimal parameters λ1 and λ2

Let (u,τ) be a solution of problem (2). Then we obtain the following condition $\partial L(u,\tau)/\partial u = 0$. This condition give us optimal λ_1 and λ_2:

$$\lambda_1 = \frac{\int_{\Omega}(1-\frac{v}{u})dxdy}{\frac{1}{\sigma^2}\int_{\Omega}(v-u)dxdy + \int_{\Omega}(1-\frac{v}{u})dxdy}, \ \lambda_2 = 1-\lambda_1.$$

The discrete form for $k = 0,1,\dots,K$ is

$$\lambda_1^k = \frac{\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}(1-\frac{v_{ij}}{u_{ij}^k})}{\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}(\frac{v_{ij}-u_{ij}^k}{\sigma^2} + 1 - \frac{v_{ij}}{u_{ij}^k})}, \ \lambda_2^k = 1-\lambda_1^k.$$

4.2 Optimal parameter μ

In order to find the optimal μ, we multiply (3) by $(v-u)$ and integrate by parts over domain Ω. Finally, we obtain the formula to find the optimal

$$\mu = \frac{\int_{\Omega}(-\frac{\lambda_1}{\sigma^2}(v-u)^2 - \lambda_2\frac{(v-u)^2}{u})dxdy}{\int_{\Omega}(\sqrt{u_x^2+u_y^2} - \frac{u_xv_x + u_yv_y}{\sqrt{u_x^2+u_y^2}})dxdy}.$$

The discrete form is

$$\mu^k = \frac{\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}(-\frac{\lambda_1^k}{\sigma^2}(v_{ij}-u_{ij}^k)^2 - \lambda_2^k\frac{(v_{ij}-u_{ij}^k)^2}{u_{ij}^k})}{\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\eta_{ij}^k},$$

$$\eta_{ij}^k = \sqrt{(\nabla_x(u_{ij}^k))^2 + (\nabla_y(u_{ij}^k))^2} - \frac{\nabla_x(u_{ij}^k)\nabla_x(v_{ij}) + \nabla_y(u_{ij}^k)\nabla_y(v_{ij})}{\sqrt{(\nabla_x(u_{ij}^k))^2 + (\nabla_y(u_{ij}^k))^2}},$$

$$\nabla_x(u_{ij}^k) = \frac{u_{i+1,j}^k - u_{i-1,j}^k}{2\Delta x},\ \nabla_y(u_{ij}^k) = \frac{u_{i,j+1}^k - u_{i,j-1}^k}{2\Delta y},$$

$$\nabla_x(v_{ij}^k) = \frac{v_{i+1,j}^k - v_{i-1,j}^k}{2\Delta x} v,\ \nabla_y(v_{ij}^k) = \frac{v_{i,j+1}^k - v_{i,j-1}^k}{2\Delta y},$$

$$u_{0j}^k = u_{1j}^k;\ u_{N_1+1,j}^k = u_{N_1,j}^k;\ u_{i0}^k = u_{i1}^k;\ u_{i,N_2+1}^k = u_{i,N_2}^k;$$

$$v_{0j} = v_{1j};\ v_{N_1+1,j} = v_{N_1 j};\ v_{i0} = v_{i1};\ v_{i,N_2+1} = v_{i,N_2};$$

$$i = 1,...,N_1;\ j = 1,...,N_2;\ k = 0,1,...,K;\ \Delta x = \Delta y = 1.$$

4.3 Optimal parameter σ

The parameter σ is calculated at the first step of the iteration process. We use the method of Immerker [6]:

$$\sigma = \frac{\sqrt{\pi/2}}{6(N_1-2)(N_2-2)}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}|u_{ij} * \Lambda|,$$

with the mask $\Lambda = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$ for convolution operator *,

$$u_{ij} * \Lambda = u_{i-1,j-1}\Lambda_{33} + u_{i,j-1}\Lambda_{32} + u_{i+1,j-1}\Lambda_{31} + u_{i-1,j}\Lambda_{23} + u_{ij}\Lambda_{22} + u_{i+1,j}\Lambda_{21} + u_{i-1,j+1}\Lambda_{13} + u_{i,j+1}\Lambda_{12} + u_{i+1,j+1}\Lambda_{11},$$

$$i = 1,...,N_1;\ j = 1,...,N_2;$$

$u_{ij} = 0$, if $i = 0$, or $j = 0$, or $i = N_1 + 1$, or $j = N_2 + 1$.

4.4 Initial solution

In the iteration procedure (6), the result depends on initial parameters $\lambda_1^0, \lambda_2^0, \mu^0$. If $\lambda_1^0, \lambda_2^0, \mu^0$ are given first, then its unsuitable values define not so good solution u_{ij} and later, not so good evaluation of a probability distribution parameters. If $\lambda_1^0, \lambda_2^0, \mu^0$ are randomized, the result is unacceptable too, because of the additional noise added in the image.

Of course, initial values of $\lambda_1^0, \lambda_2^0, \mu^0$ need to be closed to required values. We evaluate $\lambda_1^0, \lambda_2^0, \mu^0$ as average values of neighbour pixels of the image, for example, by the method of Immerker.

5 Image quality evaluation

In order to evaluate the image quality after denoising, we use criteria PSNR, MSE and SSIM [24, 25]:

$$\text{Q}_{\text{MSE}} = \frac{1}{N_1 N_2}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}(v_{ij} - u_{ij})^2,\ \text{Q}_{\text{PSNR}} = 10\lg\left(\frac{L^2}{\text{Q}_{\text{MSE}}}\right),$$

$$\text{Q}_{\text{SSIM}} = \frac{(2\bar{u}\,\bar{v} + C_1)(2\sigma_{uv} + C_2)}{(\bar{u}^2 + \bar{v}^2 + C_1)(\sigma_u^2 + \sigma_v^2 + C_2)},$$

where

$$\bar{u} = \frac{1}{N_1 N_2}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} u_{ij},\ \bar{v} = \frac{1}{N_1 N_2}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} v_{ij}.$$

$$\sigma_u^2 = \frac{1}{N_1 N_2 - 1}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}(u_{ij} - \bar{u})^2,$$

$$\sigma_v^2 = \frac{1}{N_1 N_2 - 1}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}(v_{ij} - \bar{v})^2,$$

$$\sigma_{uv} = \frac{1}{N_1 N_2 - 1}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}(u_{ij} - \bar{u})(v_{ij} - \bar{v}),$$

$$C_1 = (K_1 L)^2, C_2 = (K_2 L)^2; K_1 << 1; K_2 << 1.$$

For example, $K_1 = K_2 = 10^{-6}$, L is an image intensity with $L = 255$ for 8-bits greyscale image.

The greater the value of Q_{PSNR}, the better the image quality. If the value of Q_{PSNR} belongs to the interval from 20 to 25, then the image quality is acceptable, for example, for wireless transmission [21].

The Q_{MSE} is a mean squared error and is used to evaluate the difference between two images. The lower the value of Q_{MSE}, the better the result of restoration. The value of Q_{MSE} directly related to the value of Q_{PSNR}.

The value of Q_{SSIM} is used to evaluate an image quality by comparing the similarity of two images. This value is between -1 and 1. The greater the value of Q_{SSIM}, the better the image quality.

6 Experiments and discussion

In this paper, we consider cases as in [19] and additionally the superposition of noises. The image size is changed from 300x300 pixels to 256x256 pixels specified in PURE-LET method [11]. We process the artificial image with artificial noise and the real image with artificial noise. The artificial image is noise-free and we need to add noise with high intensity (the image to be very noisy) to reduce its quality. Therefore, we specify 0.6 for proportion of Gaussian noisy image and 0.4 for proportion of Poisson noisy image. The real image (captured by a digital device) already includes some noise. We specify 0.5 for proportion of Gaussian noisy image and 0.5 for proportion of Poisson noisy image too.

We need to point the attention in the case of Gaussian noise our method sometimes can be better than ROF, because the method to evaluate the variance of Gaussian noise can be better than one included in the original ROF model in many cases. In the case of superposition of noises, our method sometimes can be better than PURE-LET, because parameters of our method are usually more optimal than in original model too.

6.1 Artificial image with artificial noise

We use artificial image with artificial mixed noise for the first test. The image includes eight bars (Fig. 1a). Other images (Fig. 1b-j) show noisy and denoised images and zoomed out part of them.

Artificial noise is generated by linear combination, and by superposition of Poisson and Gaussian noises.

For both cases, we consider Poisson noise first. Its probability density is $p_2(v\,|\,u)$, and variance is $\sigma_2 = \sqrt{u_{i\,j}}$

at every pixel (i, j), $i = 1,..., N_1$, $j = 1,..., N_2$. Poisson noise variance is an average value $\bar{\sigma}_2 = 11.7939$. If the grey value of a pixel after adding of Poisson noise is out of the interval from 0 to 255, it needs to be reset to $v^{(2)}_{i\,j} = u_{i\,j}$. For this image, there are no pixels out of this interval. Next, we consider the variance of Gaussian noise is four times greater than the variance of Poisson noise $\sigma_1 = 4\bar{\sigma}_2 = 47.1757$.

For the linear combination of noises, we denote the intensity function of Gaussian noisy image as $v^{(1)}$. As above, values of $v^{(1)}$ need to be between 0 and 255. If the grey value of a pixel after adding of Gaussian noise is out of the interval from 0 to 255, it needs to be reset to $v^{(1)}{}_{i\,j} = u_{i\,j}$. In this case, there are 1075 pixels out of this

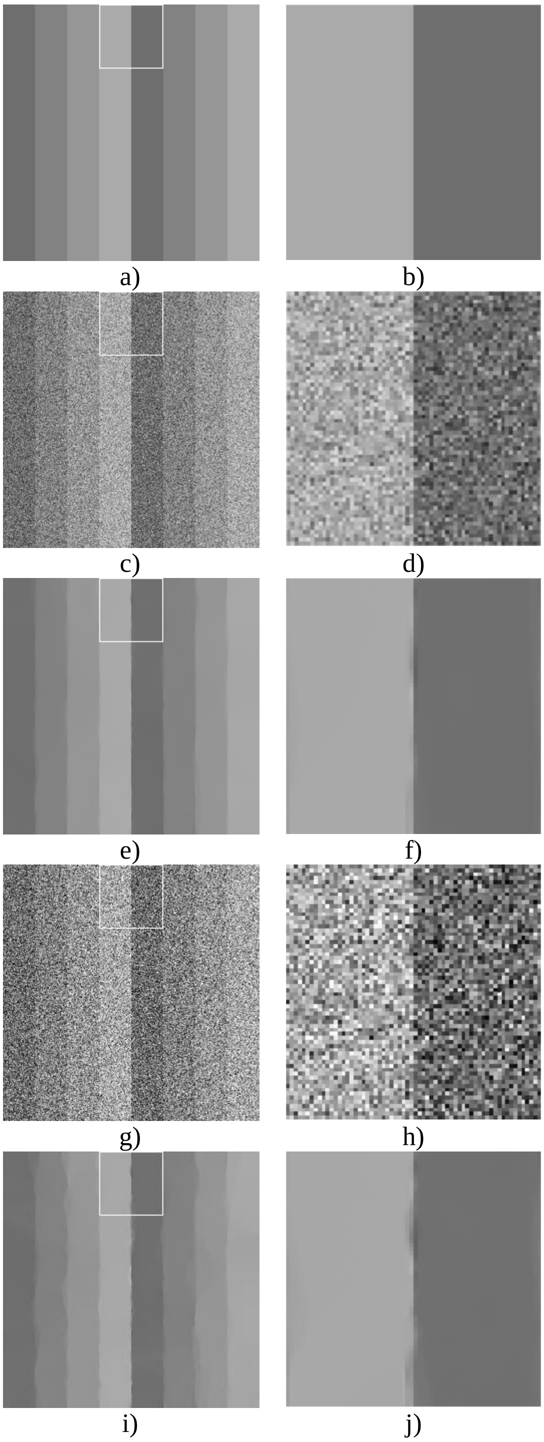


Figure 1: Denoising of the artificial image: a)-b) original image, c)-d) noisy image with linear combination of noises, e)-f) denoised image (c), g)-h) noisy image with superposition of noises, i)-j) denoised image (g).

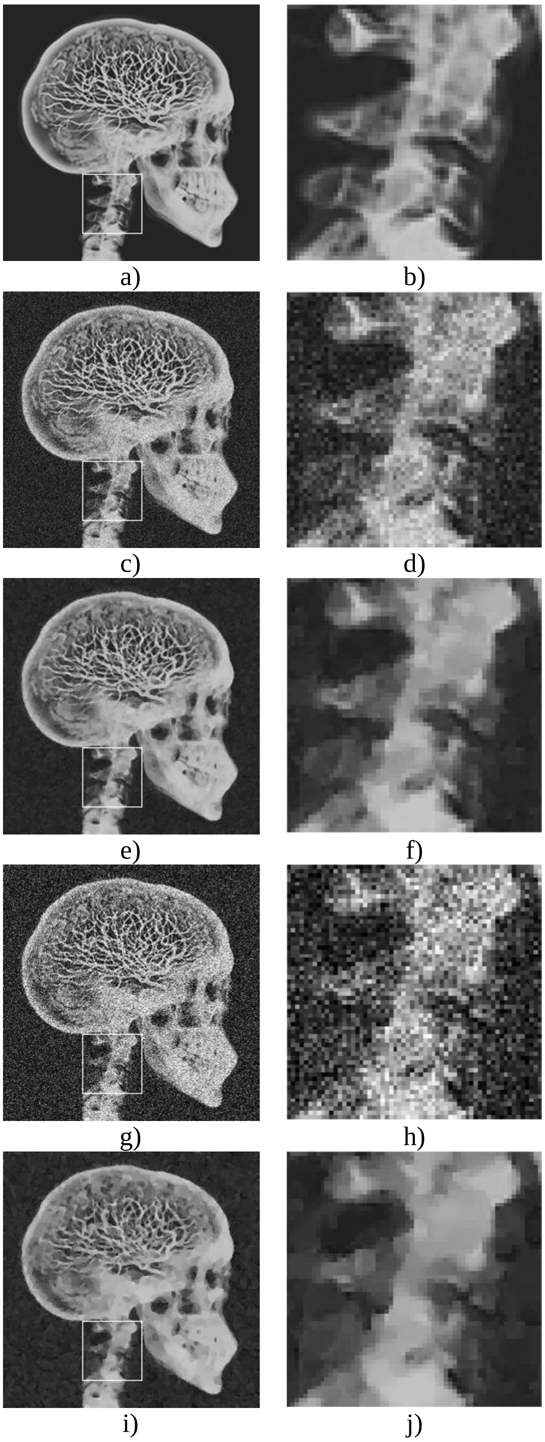


Figure 2: Denoising of the real image: a)-b) original image, c)-d) noisy image with linear combination of noises, e)-f) denoised image (c), g)-h) noisy image with superposition of noises, i)-j) denoised image (g).

interval (1.64%).

The final noisy image (linear combination of noises in Fig. 1c) is created with proportion 0.6 for Gaussian noisy image $v^{(1)}$ and proportion 0.4 for Poisson noisy image $v^{(2)}$: $v = 0.6v^{(1)} + 0.4v^{(2)}$.

Then we define proportion for linear combination as $\lambda_1 / \lambda_2 = (0.6\times 47.1757)/(0.4\times 11.7939) = 6/1$. Coefficients of linear combination are defined as λ_1 = 6/7 = 0.8571, λ_2 = 1/7 = 0.1429.

	Q_{PSNR}	Q_{SSIM}	Q_{MSE}
Noisy	19.4291	0.1073	741.5963
ROF	34.1236	0.8978	25.1606
Modified ROF	32.4315	0.8703	37.8791
PURE-LET	33.0309	0.9277	32.3587
Proposed method λ_1=0.8571, λ_2=0.1429, μ = 0.5003, σ = 47.1757	**41.1209**	**0.9841**	**4.9905**
Proposed method with automatically defined parameters λ_1=0.8414, λ_2=0.1586, μ = 0.5112, σ = 41.0314	41.0998	0.9840	5.0478

Table 1: Quality of noise removing for the artificial image with linear combination of noises.

	Q_{PSNR}	Q_{SSIM}	Q_{MSE}
Noisy	15.1406	0.0457	1990.8
ROF	31.4797	0.8364	21.2502
Modified ROF	28.4591	0.7871	27.5694
PURE-LET	28.9451	0.7986	25.9883
Proposed method λ_1=1, λ_2=0, μ = 0.3033, σ = 47.1757	**35.8011**	**0.9598**	**16.8122**
Proposed method with automatically defined parameters λ_1=0.9715, λ_2=0.0285, μ = 0.3021, σ = 46.0314	35.7589	0.9596	17.2658

Table 2: Quality of noise removing for the artificial image with Gaussian noise.

Values of Q_{PSNR}, Q_{MSE}, and Q_{SSIM} of the noisy image (linear combination of noises) are, respectively, 19.4291, 741.5963, and 0.1073.

In the case of the image with superposition of noises, we add Gaussian noise over Poisson noisy image. The intensity function of this Gaussian noisy image is $v^{(1)}$ too. As above, the grey values of $v^{(1)}$ need to be between 0 and 255. If the grey value of a pixel after adding of Gaussian noise is out of the interval from 0 to 255, it needs to be reset to $v^{(1)}_{ij} = v^{(2)}_{ij}$.

There are 1220 pixels out of this interval (1.86%). The noisy image (superposition of noises, Fig. 1g) is also Gaussian noisy image $v = v^{(1)}$. In this case, we don't know λ_1 and λ_2, therefore we use the algorithm with automatically defined parameters.

Values of Q_{PSNR}, Q_{MSE}, and Q_{SSIM} of the noisy image are, respectively, 14.9211, 2093.9827, and 0.0439.

	Q_{PSNR}	Q_{SSIM}	Q_{MSE}
Noisy	26.6776	0.3640	139.7396
ROF	36.4958	0.9381	14.5715
Modified ROF	**44.6347**	**0.9897**	**2.2001**
PURE-LET	37.4485	0.9404	10.5692
Proposed method λ_1=0, λ_2=1, μ = 0.8012, σ = 0.0001	44.6343	0.9897	2.2014
Proposed method with automatically defined parameters λ_1=0.0524, λ_2=0.9476, μ = 0.7923, σ = 2.0544	44.6156	0.9896	2.2466

Table 3: Quality of noise removing for the artificial image with Poisson noise.

	Q_{PSNR}	Q_{SSIM}	Q_{MSE}
Noisy	14.9211	0.0439	2093.983
ROF	31.2913	0.8346	48.3008
Modified ROF	30.5471	0.8232	56.5601
PURE-LET	33.9889	0.9298	25.9534
Proposed method with automatically defined parameters λ_1=0.8014, λ_2=0.1986, μ = 0.4812, σ = 40.0314	**37.3366**	**0.9677**	**12.0066**

Table 4: Quality of noise removing for the artificial image with superposition of noises.

Tables 1 – 4 show results for linear combination of noises, Gaussian noise, Poisson noise, and superposition of noises for the artificial image.

6.2 Real image with artificial noise

The artificial noise is generated by linear combination and superposition of Poisson and Gaussian noises.

For both cases, we consider Poisson noise first. Poisson noise variance is an average value $\bar{\sigma}_2 = 9.0882$. If the grey value of a pixel after adding of Poisson noise is out of the interval from 0 to 255, it needs to be reset to

$v_{ij}^{(2)} = u_{ij}$. Here there are no pixels out of this interval.

For Gaussian noise, we consider the variance of Gaussian noise is four times greater than the variance of Poisson noise $\sigma_1 = 4\bar{\sigma}_2 = 36.3529$. The real image is a human skull [14] (Fig. 2a). Others (Fig. 2b-j) show noisy and denoised images and zoomed out part of them.

For the case of linear combination of noises, we denote the intensity function of Gaussian noisy image as $v^{(1)}$. As above, the grey values of intensity function $v^{(1)}$ also need to be between 0 and 255. If the grey value of a pixel after adding of Gaussian noise is out of the interval from 0 to 255, it needs to be reset to $v_{ij}^{(1)} = u_{ij}$. In this case, there are 5355 pixels out of this interval (8.17%). The final image (linear combination of noises, Fig. 2c) is created with proportion 0.5 for Gaussian noisy image $v^{(1)}$ and proportion 0.5 for Poisson noisy image $v^{(2)}$: $v = 0.5v^{(1)} + 0.5v^{(2)}$. The proportion for linear combination is: $\lambda_1 / \lambda_2 = (0.5\times 36.3529)/(0.5\times 9.0882) = 4/1$.

Hence, coefficients of linear combination are defined as λ_1 = 4/5 =0.8, λ_2 = 1/5 = 0.2. Values of Q_{PSNR}, Q_{MSE}, and Q_{SSIM} of final noisy image are, respectively, 23.6878, 278.1619, and 0.5390.

For superposition of noises, we add Gaussian noise over Poisson noisy image. We denote the intensity function of Gaussian noisy image as $v^{(1)}$. As above, grey values of $v^{(1)}$ need to be between 0 and 255. If the grey value after adding of Gaussian noise is out of the interval from 0 to 255, it needs to be reset to $v_{ij}^{(1)} = v_{ij}^{(2)}$. In this case, there are 5621 pixels out of this interval (8.58%). The final noisy image (superposition of noises, Fig. 2g) is also the Gaussian noisy image $v = v^{(1)}$.

In this case, we don't know λ_1 and λ_2, therefore we use the algorithm to find them. Values of Q_{PSNR}, Q_{MSE}, and Q_{SSIM} of the final noisy image (superposition) are, respectively, 17.8071, 1077.3831, and 0.3242.

	Q_{PSNR}	Q_{SSIM}	Q_{MSE}
Noisy	23.6878	0.5390	278.1619
ROF	27.3974	0.8295	118.3975
Modified ROF	25.5644	0.7513	197.5403
PURE-LET	25.7781	0.8105	191.0341
Proposed method λ_1=0.8, λ_2=0.2, μ = 0.0524, σ = 36.3529	**27.6641**	**0.8331**	**110.9451**
Proposed method with automatically defined parameters λ_1=0.7804, λ_2=0.2196, μ = 0.0512, σ = 34.2311	27.6039	0.8325	112.8984

Table 5: Quality of noise removing for the real image with linear combination of noises.

	Q_{PSNR}	Q_{SSIM}	Q_{MSE}
Noisy	18.0693	0.3337	1014.3
ROF	24.0246	0.7299	257.4095
Modified ROF	23.2511	0.7019	311.8742
PURE-LET	23.8712	0.7989	265.6153
Proposed method λ_1=1, λ_2=0, μ = 0.0956, σ = 36.3529	**24.2011**	**0.8029**	**242.5101**
Proposed method with automatically defined parameters λ_1=0.9538, λ_2=0.0462, μ = 0.0902, σ = 35.0633	24.1882	0.8028	247.8894

Table 6: Quality of noise removing for the real image with Gaussian noise.

	Q_{PSNR}	Q_{SSIM}	Q_{MSE}
Noisy	28.4991	0.7625	91.8683
ROF	31.0567	0.9457	50.9818
Modified ROF	**31.1992**	**0.9022**	**48.9375**
PURE-LET	30.8955	0.8678	53.1066
Proposed method λ_1=0, λ_2=1, μ = 0.0541, σ = 0.0001	31.1334	0.8986	49.7922
Proposed method with automatically defined parameters λ_1=0.0491, λ_2=0.9509, μ = 0.0567, σ = 4.2012	31.1316	0.8986	50.1094

Table 7: Quality of noise removing for the real image with Poisson noise.

	Q_{PSNR}	Q_{SSIM}	Q_{MSE}
Noisy	17.8077	0.3242	1077.383
ROF	23.1936	0.7062	311.6856
Modified ROF	23.0413	0.7033	319.3831
PURE-LET	23.6278	0.7072	282.0349
Proposed method with automatically defined parameters λ_1=0.7704, λ_2=0.2296, μ = 0.1102, σ = 36.3412	**23.7292**	**0.7094**	**275.5229**

Table 8: Quality of noise removing for the real image with superposition of noises.

Tables 5 – 8 show results for linear combination of noises, Gaussian noise, Poisson noise, and superposition of noises for the real image.

6.3 About of initial solution

In order to create the initial image, we use the convolution operator. The table 9 shows the dependency of restored result for the initial image, where:

(a) Initial parameters $\lambda_1^0 = 0, \lambda_2^0 = 1, \mu = 1$;

(b) Initial parameters $\lambda_1^0 = \lambda_2^0 = 0.5, \mu = 1$;

(c) Initial solution u^0 is given as a randomized matrix;

(d) Initial solution $u^0 = \Lambda * v$ is given as an average value of neighbour pixels by the convolution operator with the mask $\Lambda = (1/9)$ of the size 3x3.

Table 9 shows the best result of denoising is (d) by criteria PSNR and MSE.

The result (c) by SSIM looks different in contract to ones in Tables 1-8. It illustrates incorrectness of a randomized initial solution (accidental and not stable, if a probability distribution is unknown).

Next, we have to notice that the non-optimal result (a) has been used in experiments for Table 5. It appears to be enough for the good result with automatically defined model parameters.

	(a)	(b)	(c)	(d)
λ_1	0.7804	0.8094	0.8733	0.8032
λ_2	0.2196	0.1906	0.1267	0.1968
μ	0.0512	0.0573	0.0653	0.0565
σ	34.2311			
Q_{PSNR}	27.6039	27.2214	26.5611	**27.6523**
Q_{MSE}	112.8984	120.4355	132.0264	**107.5431**
Q_{SSIM}	0.8325	0.8317	**0.8395**	0.8392

Table 9: Dependency of denoising on initial solution.

At last, the variant (b) initially looks better than (a) for kind of better assumption of $\lambda_1^0 = \lambda_2^0 = 0.5$ to process the real image. Nevertheless, our assumption about $\mu = 1$ is very far from the good one, and evidently the limit of the number of steps $K = 500$ is insufficient in this case.

As a result, the variant (d) is the best idea for initial solution.

7 Conclusion

In this paper, we proposed a novel method that can effectively remove the mixed Poisson-Gaussian noise. Furthermore, our proposed method can be also used to remove Gaussian or Poisson noise separately. This method is based on the variational approach.

The denoising result strongly depends on values of coefficients of linear combination λ_1 and λ_2. These values can be set manually or can be defined automatically. When processing real images, we can use the proposed method with automatically defined parameters.

Although our method concentrates on removing the linear combination of noise, but it also efficiently removes the superposition of noises. In this case, we consider the superposition of noises is equivalent to some linear combination of them with coefficients found in iteration process.

In this paper we show that our simple model “feels” well the wide range of proportion of two types of noises. As a result, it appears to be the good basis for removing superposition of such noises.

It is evident, the iteration process (6) used here is insufficiently effective in comparing with other possible computational schemes. In this paper, we try to compare our approach to image denoising with PURE-LET method only in possible reduction of our model complexity, not in others.

We would like to express our great thanks to developers of PURE-LET method for kindly granted us the original executable module of it.

8 Acknowledgements

This work is partially supported by Russian Foundation for Basic Research, under grants 13-07-00529, 14-07-00964, 15-07-02228, 16-07-01039, and by Ministry of Education and Training, Vietnam, under grant number B2015–01–90.

References

[1] Abe C., Shimamura T. Iterative Edge-Preserving adaptive Wiener filter for image denoising. ICCEE, 2012, Vol. 4, No. 4, P. 503-506.

[2] Chan T.F., Shen J. *Image processing and analysis: Variational, PDE, Wavelet, and stochastic methods*. SIAM, 2005.

[3] Chen K. Introduction to variational image processing models and application. *International journal of computer mathematics*, 2013, Vol. 90, No.1, P.1-8.

[4] Getreuer P. *Rudin-Osher-Fatemi total variation denoising using split Bregman*. IPOL 2012: http://www.ipol.im/pub/art/2012/g-tvd/.

[5] Gill P.E., Murray W. *Numerical methods for constrained optimization*. Academic Press Inc., 1974.

[6] Immerker J. Fast noise variance estimation. *Computer vision and image understanding*, 1996, Vol. 64, No.2, P. 300-302.

[7] Jezierska A. An EM approach for Poisson-Gaussian noise modelling. *EUSIPCO 19th*, 2011, Vol. 62, Is. 1, P. 13-30.

[8] Jezierska A. Poisson-Gaussian noise parameter estimation in fluorescence microscopy imaging. *IEEE International Symposium on Biomedical Imaging 9th*, 2012, P. 1663-1666.

[9] Le T., Chartrand R., Asaki T.J. A variational approach to reconstructing images corrupted by Poisson noise. *Journal of mathematical imaging and vision*, 2007, Vol. 27, Is. 3, P. 257-263.

[10] Li F., Shen C., Pi L. A new diffusion-based variational model for image denoising and segmentation. *Journal mathematical imaging and vision*, 2006, Vol. 26, Is. 1-2, P. 115-125.

[11] Luisier F., Blu T., Unser M. Image denoising in mixed Poisson-Gaussian noise. *IEEE transaction on Image processing*, 2011, Vol. 20, No. 3, P. 696-708.

[12] Lysaker M., Tai X. Iterative image restoration combining total variation minimization and a second-order functional. *International journal of computer vision*, 2006, Vol. 66, P. 5-18.

[13] Mittal A., Moorthy A.K., and Bovik A.C. No reference image quality assessment in the spatial domain, *IEEE Trans. Image Processing* 21 (12), 4695–4708 (2012).

[14] Nick V. *Getty images*: http://well.blogs.nytimes.com/2009/09/16/what-sort-of-exercise-can-make-you-smarter/.

[15] Rankovic N., Tuba M. Improved adaptive median filter for denoising ultrasound images. *Advances in computer science,* 2012, P.169-174.

[16] Rubinov A., Yang X. *Applied Optimization: Lagrange-type functions in constrained non-convex optimization*. Springer, 2003.

[17] Rudin L.I., Osher S., Fatemi E. Nonlinear total variation based noise removal algorithms. *Physica D.*, 1992, Vol. 60, P. 259-268.

[18] Scherzer O. *Variational methods in Imaging*. Springer, 2009.

[19] Thanh N. H. Dang, Dvoenko Sergey D., Dinh Viet Sang. *A Denoising Method Based on Total Variation.* Proc. of 6th Intern. Symposium on Information and Communication Technology (SoICT-2015). P. 223-230. ACM, NY, USA.

[20] Thanh D.N.H., Dvoenko S.D. A method of total variation to remove the mixed Poisson-Gaussian noise. *Pattern Recognition and Image Analysis*, 26 (2), 285-293 (2016).
DOI: 10.1134/S1054661816020231.

[21] Thomos N., Boulgouris N.V., Strintzis M.G. Optimized Transmission of JPEG2000 streams over Wireless channels. *IEEE transactions on image processing*, 2006, Vol. 15, No.1, P .54-67.

[22] Tran M.P., Peteri R., Bergounioux M. Denoising 3D medical images using a second order variational model and wavelet shrinkage. *Image analysis and recognition*, 2012, Vol. 7325, P. 138-145.

[23] Wang C., Li T. An improved adaptive median filter for Image denoising. ICCEE, 2012, Vol. 53, No. 2.64, P. 393-398.

[24] Wang Z. Image quality assessment: From error visibility to structural similarity. *IEEE transaction on Image processing*, Vol. 13, No. 4, P. 600-612. 2004.

[25] Wang Z., Bovik A.C. *Modern image quality assessment*. Morgan & Claypool Publisher, 2004.

[26] Xu J., Feng X., Hao Y. A coupled variational model for image denoising using a duality strategy and split Bregman. *Multidimensional systems and signal processing*, 2014, Vol. 25, P. 83-94.

[27] Zhu Y. Noise reduction with low dose CT data based on a modified ROF model. *Optics express*, 2012, Vol. 20, No. 16, P. 17987-18004.

[28] Zeidler E. *Nonlinear functional analysis and its applications: Variational methods and optimization*. Springer, 1985.

[29] Zosso D., Bustin A. A Primal-Dual Projected Gradient Algorithm for Efficient Beltrami Regularization. Computer Vision and Image Understanding, 2014: http://www.math.ucla.edu/~zosso/.

A Multi-Criteria Document Clustering Method Based on Topic Modeling and Pseudoclosure Function

Quang Vu Bui
CHArt Laboratory EA 4004, Ecole Pratique des Hautes Etudes, PSL Research University, 75014, Paris, France
Hue University of Sciences, Vietnam
E-mail: quang-vu.bui@etu.ephe.fr

Karim Sayadi
CHArt Laboratory EA 4004, Ecole Pratique des Hautes Etudes, PSL Research University, 75014, Paris, France
Sorbonne University, UPMC Univ Paris 06, France
E-mail: karim.sayadi@upmc.fr

Marc Bui
CHArt Laboratory EA 4004, Ecole Pratique des Hautes Etudes, PSL Research University, 75014, Paris, France
E-mail: marc.bui@ephe.sorbonne.fr

Keywords: pretopology, pseudoclosure, clustering, k-means, latent Dirichlet allocation, topic modeling, Gibbs sampling

Received: July 1, 2016

We address in this work the problem of document clustering. Our contribution proposes a novel unsupervised clustering method based on the structural analysis of the latent semantic space. Each document in the space is a vector of probabilities that represents a distribution of topics. The document membership to a cluster is computed taking into account two criteria: the major topic in the document (qualitative criterion) and the distance measure between the vectors of probabilities (quantitative criterion). We perform a structural analysis on the latent semantic space using the Pretopology theory that allows us to investigate the role of the number of clusters and the chosen centroids, in the similarity between the computed clusters. We have applied our method to Twitter data and showed the accuracy of our results compared to a random choice number of clusters.

Povzetek: Predstavljena metoda grupira dokumente glede na semantični prostor. Eksperimenti so narejeni na podatkih s Twitterja.

1 Introduction

Classifying a set of documents is a standard problem addressed in machine learning and statistical natural language processing [13]. Text-based classification (also known as text categorization) examines the computer-readable ASCII text and investigates linguistic properties to categorize the text. When considered as a machine learning problem, it is also called statistical Natural Language Processing (NLP) [13]. In this task, a text is assigned to one or more predefined class labels (i.e category) through a specific process in which a classifier is built, trained on a set of features and then applied to label future incoming texts. Given the labels, the task is performed within the supervised learning framework. Several Machine Learning algorithms have been applied to text classification (see [1] for a survey): Rocchio's Algorithm, N-Nearest Neighbors, Naive Bayes, Decision tree, Support Vector Machine (SVM).

Text-based features are typically extracted from the so-called word space model that uses distributional statistics to generate high-dimensional vector spaces. Each document is represented as a vector of word occurrences. The set of documents is represented by a high-dimensional sparse matrix. In the absence of predefined labels, the task is referred as a clustering task and is performed within the unsupervised learning framework. Given a set of keywords, one can use the angle between the vectors as a measure of similarity between the documents. Depending on the algorithm, different measures are used. Nevertheless, this approach suffers from the curse of dimensionality because of the sparse matrix that represents the textual data. One of the possible solutions is to represent the text as a set of topics and use the topics as an input for a clustering algorithm.

To group the documents based on their semantic content, the topics need to be extracted. This can be done using one of the following three methods. (i) LSA [10] (Latent Semantic Analysis) uses the Singular Value Decomposition methods to decompose high-dimensional sparse matrix to three matrices: one matrix that relates words to topics, another one that relates topics to documents and a diagonal matrix of singular value. (ii) Probabilistic LSA [8] is a probabilistic model that treats the data as a set of observations coming from a generative model. The generative model includes hidden variables representing the probability distribution of the words in the topics and the probability distribution of the topics in the words. (iii) Latent Dirichlet Allocation [4] (LDA) is a Bayesian extension of probabilistic LSA. It defines a complete generative model with a uniform prior and full Bayesian estimator.

LDA gives us three latent variables after computing the posterior distribution of the model; the topic assignment, the distribution of words in each topic and the distribution of the topics in each document. Having the distribution of topics in documents, we can use it as the input for clustering algorithms such as k-means, hierarchical clustering.

K-means uses a distance measure to group a set of data points within a predefined random number of clusters. Thus, to perform a fine-grained analysis of the clustering process we need to control the number of clusters and the distance measure. The Pretopology theory [3] offers a framework to work with categorical data, to establish a multi-criteria distance for measuring the similarity between the documents and to build a process to structure the space [11] and infer the number of clusters for k-means. We can then tackle the problem of clustering a set of documents by defining a family of binary relationships on the topic-based contents of the documents. The documents are not only grouped together using a measure of similarity but also using the pseudoclosure function built from a family of binary relationships between the different hidden semantic contents (i.e topics).

The idea of using Pretopology theory for k-means clustering has been proposed by [16]. In this paper, the authors proposed the method to find automatically a number k of clusters and k centroids for k-means clustering by results from structural analysis of minimal closed subsets algorithm [11] and also proposed to use pseudoclosure distance constructed from the relationships family to examine the similarity measure for both numeric and categorical data. The authors illustrated the method with a toy example about the toxic diffusion between 16 geographical areas using only one relationship.

For the problem of classification, the authors of this work [2] built a vector space with Latent Semantic Analysis (LSA) and used the pseudoclosure function from Pretopological theory to compare all the cosine values between the studied documents represented by vectors and the documents in the labeled categories. A document is added to a labeled categories if it has a maximum cosine value.

Our work differs from the work of [2] and extends the method proposed in [16] with two directions: first, we exploited this idea in document clustering and integrated structural information from LDA using the pretopological concepts of pseudoclosure and minimal closed subsets introduced in [11]. Second, we showed that Pretopology theory can apply to multi-criteria clustering by defining the pseudo distance built from multi-relationships. In our paper, we clustered documents by using two criteria: one based on the major topic of document (qualitative criterion) and the other based on Hellinger distance (quantitative criterion). The clustering is based on these two criteria but not on multicriteria optimization [5] for clustering algorithms.Our application on Twitter data also proposed a method to construct a network from the multi-relations network by choosing the set of relations and then applying strong or weak Pretopology.

We present our approach in a method that we named the Method of Clustering Documents using Pretopology and Topic Modeling (MCPTM). MCPTM organizes a set of unstructured entities in a number of clusters based on multiple relationships between each two entities. Our method discovers the topics expressed by the documents, tracks changes step by step over time, expresses similarity based on multiple criteria and provides both quantitative and qualitative measures for the analysis of the document.

1.1 Contributions

The contributions of this paper are as follows.

1. We propose a new method to cluster text documents using Pretopology and Topic Modeling.

2. We investigate the role of the number of clusters inferred by our analysis of the documents and the role of the centroids in the similarity between the computed clusters.

3. We conducted experiments with different distances measures and show that the distance measure that we introduced is competitive.

1.2 Outline

The article is organized as follows: Section 2, 3 present some basic concepts such as Latent Dirichlet Allocation (Section 2) and the Pretopology theory (Section 3), Section 4 explains our approach by describing at a high level the different parts of our algorithm. In Section 5, we apply our algorithm to a corpus consisting of microblogging posts from Twitter.com. We conclude our work in Section 6 by presenting the obtained results.

2 Topic modeling

Topic Modeling is a method for analyzing large quantities of unlabeled data. For our purposes, a topic is a probability distribution over a collection of words and a topic model is a formal statistical relationship between a group of observed and latent (unknown) random variables that specifies a probabilistic procedure to generate the topics [4, 8, 6, 15]. In many cases, there exists a semantic relationship between terms that have high probability within the same topic – a phenomenon that is rooted in the word co-occurrence patterns in the text and that can be used for information retrieval and knowledge discovery in databases.

2.1 Latent Dirichlet allocation

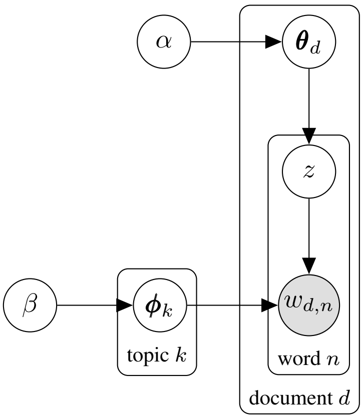


Figure 1: Bayesian Network (BN) of Latent Dirichlet Allocation.

Latent Dirichlet Allocation (LDA) by Blei et al. [4] is a generative probabilistic model for collections of grouped discrete data. Each group is described as a random mixture over a set of latent topics where each topic is a discrete distribution over the vocabulary collection. LDA is applicable to any corpus of grouped discrete data. In our work we refer to the standard Natural Language Processing (NLP) use case where a corpus is a collection of documents, and the discrete data are represented by the occurrence of words.

LDA is a probabilistic model for unsupervised learning, it can be seen as a Bayesian extension of the probabilistic Latent Semantic Analysis (pLSA) [8]. More precisely, LDA defines a complete generative model which is a full Bayesian estimator with a uniform prior while pLSA provides a Maximum Likelihood (ML) or Maximum a Posterior (MAP) estimator. For more technical details we refer to the work of Gregor Heinrich [7]. The generative model of LDA is described with the probabilistic graphical model [9] in Fig. 1.

In this LDA model, different documents d have different topic proportions θ_d. In each position in the document, a topic z is then selected from the topic proportion θ_d. Finally, a word is picked from all vocabularies based on their probabilities ϕ_k in that topic z. θ_d and ϕ_k are two Dirichlet distributions with α and β as hyperparameters. We assume symmetric Dirichlet priors with α and β having a single value.

The hyperparameters specify the nature of the priors on θ_d and ϕ_k. The hyperparameter α can be interpreted as a prior observation count of the number of times a topic z is sampled in document d [15]. The hyper hyperparameter β can be interpreted as a prior observation count on the number of times words w are sampled from a topic z [15].

The advantage of the LDA model is that interpreting at the topic level instead of the word level allows us to gain more insights into the meaningful structure of documents since noise can be suppressed by the clustering process of words into topics. Consequently, we can use the topic proportion in order to organize, search, and classify a collection of documents more effectively.

2.2 Inference with Gibbs sampling

In this subsection, we specify a topic model procedure based on the Latent Dirichlet Allocation (LDA) and Gibbs Sampling.

The key problem in Topic Modeling is posterior inference. This refers to reversing the defined generative process and learning the posterior distributions of the latent variables in the model given the observed data. In LDA, this amounts solving the following equation:

$$p(\theta, \phi, z|w, \alpha, \beta) = \frac{p(\theta, \phi, z, w|\alpha, \beta)}{p(w|\alpha, \beta)} \tag{1}$$

Unfortunately, this distribution is intractable to compute [7]. The normalization factor in particular, $p(w|\alpha, \beta)$, cannot be computed exactly. However, there are a number of approximate inference techniques available that we can apply to the problem including variational inference (as used in the original LDA paper [4]) and Gibbs Sampling that we shall use.

For LDA, we are interested in the proportions of the topic in a document represented by the latent variable θ_d, the topic-word distributions $\phi^{(z)}$, and the topic index assignments for each word z_i. While conditional distributions - and therefore an LDA Gibbs Sampling algorithm - can be derived for each of these latent variables, we note that both θ_d and $\phi^{(z)}$ can be calculated using just the topic index assignments z_i (i.e. z is a sufficient statistic for both these distributions). Therefore, a simpler algorithm can be used if we integrate out the multinomial parameters and simply sample z_i. This is called a collapsed Gibbs sampler [6, 15].

The collapsed Gibbs sampler for LDA needs to compute the probability of a topic z being assigned to a word w_i, given all other topic assignments to all other words. Somewhat more formally, we are interested in computing the following posterior up to a constant:

$$p(z_i \mid z_{-i}, \alpha, \beta, w) \tag{2}$$

where z_{-i} means all topic allocations except for z_i.

Equation 3 shows how to compute the posterior distribution for topic assignment.

$$P(z_i = j|z_{-i}, w) \propto \frac{n^{w_i}_{-i,j} + \beta}{n^{(\cdot)}_{-i,j} + V\beta} \frac{n^{d_i}_{-i,j} + \alpha}{n^{d_i}_{-i,\cdot} + K\alpha} \tag{3}$$

where $n^{w_i}_{-i,j}$ is the number of times word w_i was related to topic j. $n^{(\cdot)}_{-i,j}$ is the number of times all other words

Algorithm 1 The LDA Gibbs sampling algorithm.

```
Require: words $w \in$ corpus $\mathcal{D} = (d_1, d_2, \ldots, d_M)$
procedure LDA-GIBBS($w$, $\alpha$, $\beta$, $T$)
    randomly initialize z and increment counters
    loop for each iteration
        loop for each word w in corpus $\mathcal{D}$
            Begin
                word $\leftarrow w[i]$
                tp $\leftarrow z[i]$
                $n_{d,tp}- = 1; n_{word,tp}- = 1; n_{tp}- = 1$
                loop for each topic j $\in \{0, \ldots, K-1\}$
                    compute $P(z_i = j | z_{-i}, w)$
                $tp \leftarrow sample \quad from \quad p(z|.)$
                $z[i] \leftarrow tp$
                $n_{d,tp}+ = 1; n_{word,tp}+ = 1; n_{tp}+ = 1$
            End
    Compute $\phi^{(z)}$
    Compute $\theta_d$
    return $z, \phi^{(z)}, \theta_{\mathcal{D}}$                    ▷ Output
end procedure
```

were related with topic j. $n^{d_i}_{-i,j}$ is the number of times topic j was related with document d_i. The number of times all other topics were related with document d_i is annotated with $n^{d_i}_{-i,.}$. Those notations were taken from the work of Thomas Griffiths and Mark Steyvers [6].

$$\hat{\phi}^{(w)}_j = \frac{n^{(w)}_j + \beta}{n^{(\cdot)}_j + V\beta} \tag{4}$$

$$\hat{\theta}^{(d)}_j = \frac{n^{(d)}_j + \alpha}{n^{(d)}_. + K\alpha} \tag{5}$$

Equation 4 is the Bayesian estimation of the distribution of the words in a topic. Equation 5 is the estimation of the distribution of topics in a document.

3 Pretopology theory

The Pretopology is a mathematical modeling tool for the concept of proximity. It was first developed in the field of social sciences for analyzing discrete spaces [3]. The Pretopology establishes powerful tools for conceiving a process to structure the space and infer the number of clusters for example. This is made possible by ensuring a follow-up of the process development of dilation, alliance, adherence, closed subset and acceptability [16, 12].

3.1 Pseudoclosure

Let consider a nonempty set E and $\mathcal{P}(E)$ which designates all the subsets of E.

Definition 1. *A pseudoclosure $a(.)$ is a mapping from $\mathcal{P}(E)$ to $\mathcal{P}(E)$, which satisfies following two conditions:*

$$a(\emptyset) = \emptyset; \forall A \subset E, A \subset a(A) \tag{6}$$

A pretopological space (E, a) is a set E endowed with a pseudoclosure function $a.()$.

Subset $a(A)$ is called the pseudoclosure of A. As $a(a(A))$ is not necessarily equal to a(A), a sequential appliance of pseudoclosure on A can be used to model expansions: $A \subset a(A) \subset a(a(A)) = a^2(A) \subset \ldots \subset a^k(A)$

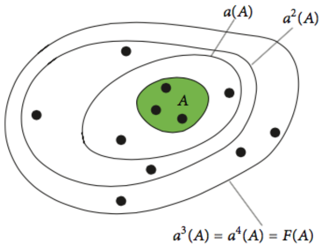


Figure 2: Iterated application of the pseudoclosure map leading to the closure.

Definition 2. *Let (E, a) a pretopological space, $\forall A, A \subset E$. A is a closed subset if and only if $a(A) = A$.*

Definition 3. *Given a pretopological space (E, a), call the closure of A, when it exists, the smallest closed subset of (E, a) which contains A. The closure of A is denoted by $F(A)$.*

Remark:

- $F(A)$ is the intersection of all closed subsets which contain A. In the case where (X, a) is a "general" pretopological space, the closure may not exist.
- Closure is very important because of the information it gives about the "influence" or "reachability" of a set, meaning, for example, that a set A can influence or reach elements into $F(A)$, but not further (see Figure 2).

Hence, it is necessary to build a pretopological spaces in which the closure always exists. V-type pretopological spaces are the most interesting cases.

Definition 4. *A Pretopology space (E, a) is called $\mathcal{V}$-type space if and only if*

$$\forall A \subset E, \forall B \subset E, (A \subset B) \Rightarrow (a(A) \subset a(B)) \tag{7}$$

Proposition 1. *In any pretopological space of type V, given a subset A of E, the closure of A always exists.*

The other reason why we use the spaces of type V is that we can build them from a family of reflexive binary relations on the finite set E. That thus makes it possible to take various points of view (various relations) expressed in a qualitative way to determine the pretopological structure placed on E. So, it can be applied on multi-criteria clustering or multi-relations networks.

3.2 Pretopology and binary relationships

Suppose we have a family $(R_i)_{i=1,\ldots,n}$ of binary reflexive relationships on a finite set E. Let us consider $\forall i = 1, 2, \ldots, n, \forall x \in E, V_i(x)$ defined by:

$$V_i(x) = \{y \in E | x \, R_i \, y\} \tag{8}$$

Then, the pseudoclosure $a_s(.)$ is defined by:

$$a_s(A) = \{x \in E | \forall i = 1, 2, \ldots, n, V_i(x) \cap A \neq \emptyset\} \tag{9}$$

Pretopology defined on E by $a_s(.)$ using the intersection operator is called the strong Pretopology induced by the family $(R_i)_{i=1,\dots,n}$.

Similarly, we can define weak Pretopology from $a_w(.)$ by using the union operator:

$$a_w(A) = \{x \in E | \exists i = 1, 2, \dots, n, V_i(x) \cap A \neq \emptyset\} \quad (10)$$

Proposition 2. *$a_s(.)$ and $a_w(.)$ determine on E a pretopological structure and the spaces (E, a_s), (E, a_w) are of V-type.*

3.3 Minimal closed subsets

We denote $\mathcal{F}_e$ as the family of elementary closed subsets, the set of closures of each singleton $\{x\}$ of $P(E)$. So in a V-type pretopological space, we get:

- $\forall x \in E, \exists F_x$: closure of $\{x\}$.
- $\mathcal{F}_e = \{F_x | x \in E\}$

Definition 5. *F_{min} is called a minimal closed subset if and only if F_{min} is a minimal element for inclusion in $\mathcal{F}_e$.*

We denote $\mathcal{F}_m = \{F_{m_j}, j = 1, 2, \dots, k\}$, the family of minimal closed subsets, the set of minimal closed subsets in $\mathcal{F}_e$.

4 Our approach

In our approach, we build The Method of Clustering Documents using Pretopology and Topic Modeling (MCPTM) which clusters documents via Topic Modeling and pseudoclosure. MCPTM can be built by:

1. Defining the topic-distribution of each document d_i in corpus $\mathcal{D}$ by document structure analysis using LDA.
2. Defining two binary relationships: R_{MTP} based on major topic and R_{d_H} based on Hellinger distance.
3. Building the pseudoclosure function from two binary relationships R_{MTP}, R_{d_H}.
4. Building the pseudoclosure distance from pseudoclosure function.
5. Determining initial parameters for the k-means algorithm from results of minimal closed subsets.
6. Using the *k-means* algorithm to cluster sets of documents with initial parameters from the result of minimal closed subsets, the pseudoclosure distance to compute the distance between two objects and the inter-pseudoclosure distance to re-compute the new centroids.

4.1 Document structure analysis by LDA

A term-document matrix is given as an input to LDA and it outputs two matrices:

- The document-topic distribution matrix θ.
- The topic-term distribution matrix ϕ.

The topic-term distribution matrix $\phi \in \mathbf{R}^{K \times V}$ consists of K rows, where the i-th row $\phi_i \in \mathbf{R}^V$ is the word distribution of topic i. The terms with high ϕ_{ij} values indicate that they are the representative terms of topic i. Therefore, by looking at such terms one can grasp the meaning of each topic without looking at the individual documents in the cluster.

In a similar way, the document-topics distributions matrix $\theta \in \mathbf{R}^{M \times K}$ consists of M rows, where the i-th row $\theta_i \in \mathbf{R}^K$ is the topic distribution for document i. A high probability value of θ_{ij} indicates that document i is closely related to topic j. In addition, documents with low θ_{ij} values over all the topics are noisy documents that belong to none of the topics. Therefore, by looking at the θ_{ij} values, one can understand how closely the document is related to the topic.

4.2 Defining binary relationships

By using LDA, each document may be characterized by its topic distribution and also be labeled by the topic with the highest probability. In this subsection, we use this information to define the relations between two documents based on the way we consider the "similarity" between them.

4.2.1 Based on major topic

Firstly, based on the label information, we can consider connecting the documents if they have the same label. However, in some cases such as noisy documents, the probability of label topic is very small and it is not really good if we use this label to represent a document. Hence, we just use the label information if its probability is higher than threshold p_0. We define the major topic of each document as:

Definition 6. *$MTP(d_i)$ is the major topic of document d_i if $MTP(d_i)$ is the topic with highest probability in the topic distribution of document d_i and this probability is greater than threshold p_0, $p_0 \geq 1/K$, K is the number of topic.*
$MTP(d_i) = \{k | \theta_{ik} = max_j \theta_{ij} \quad and \quad \theta_{ik} \geq p_0\}$.

Considering two documents d_m,d_n with their major topic $MTP(d_m)$, $MTP(d_n)$, we see that document d_m is close to document d_n if they have the same major topic. So, we proposed a definition of binary relationship R_{MTP} of two documents based on their major topic as:

Definition 7. *Document d_m has binary relationship R_{MTP} with document d_n if d_m and d_n have the same major topic.*

4.2.2 Based on Hellinger distance

Secondly, we can use the topic distributions of documents to define the relation based the similarity between two real number vectors or two probability distributions. If we consider a probability distribution as a vector, we can choose some distances or similarity measures related to the vector distance such as Euclidean distance, Cosine Similarity, Jaccard Coefficient, Pearson Correlation Coefficient, etc. But, it is better if we choose distances or similarity measures related to the probability distribution such as Kullback-Leibler Divergence, Bhattacharyya distance, Hellinger distance, etc. We choose the Hellinger distance because it is a metric for measuring the deviation between two probability distributions, easily to compute and especially limited in $[0, 1]$.

Definition 8. *For two discrete probability distributions $P = (p_1, \dots, p_k)$ and $Q = (q_1, \dots, q_k)$, their Hellinger distance is defined as*

$$d_H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2}, \tag{11}$$

The Hellinger distance is directly related to the Euclidean norm of the difference of the square root vectors, i.e.

$$d_H(P, Q) = \frac{1}{\sqrt{2}} \left\| \sqrt{P} - \sqrt{Q} \right\|_2.$$

The Hellinger distance satisfies the inequality of $0 \leq d_H \leq 1$. This distance is a metric for measuring the deviation between two probability distributions. The distance is 0 when $P = Q$. Disjoint P and Q shows the maximum distance of 1.

The lower the value of the Hellinger distance, the smaller the deviation between two probability distributions. So, we can use the Hellinger distance to measure the similarity between two documents d_m,d_n. We then define the binary relationship R_{d_H} between two documents as:

Definition 9. *Document d_m has binary relationship R_{d_H} with document d_n if $d_H(d_m, d_n) \leq d_0, 0 \leq d_0 \leq 1$, d_0 is the accepted threshold.*

4.3 Building pseudoclosure function

Based on two binary relationships R_{MTP} and R_{d_H}, we can build the neighborhood basis (see. Algorithm 2) and then build the pseudoclosures (see Algorithm 3) for strong (with intersection operator) and weak (with union operator) Pretopology.

4.4 Building pseudoclosure distance

In standard *k-means*, the centroid of a cluster is the average point in the multidimensional space. Its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster which are not effective with categorical data analysis. On the other hand, the pseudoclosure distance is used to examine the similarity using both numeric and categorical data. Therefore, it can contribute to improving the classification with k-means.

Algorithm 2 Neighborhood Basis Using Topic Modeling.

```
Require: document-topic distribution matrix $\theta$, corpus $\mathcal{D}$
Require: $R_{MTP}$, $R_{d_H}$: family of relations.
procedure NEIGHBORHOOD-TM($\mathcal{D}, \theta, R_{MTP}, R_{d_H}$)
    loop for each relation $R_i \in \{R_{MTP}, R_{d_H}\}$
        loop for each document $d_m \in \mathcal{D}$
            loop for each document $d_n \in \mathcal{D}$
                If $R_i(d_m, d_n)$ then
                    $B_i[d_m].append(d_n)$
    return $B = [B_1, B_2]$                                    ▷ Output
end procedure
```

Algorithm 3 Pseudoclosure using Topic Modeling.

```
Require: $B = (B_1, B_2), \mathcal{D} = \{d_1, \dots, d_M\}$
procedure PSEUDOCLOSURE($A, B, \mathcal{D}$)
    aA=A
    loop for each document $d_n \in \mathcal{D}$
        If $(A \cap B_1[d_n] \neq \emptyset \quad or \quad A \cap B_2[d_n] \neq \emptyset)$ then
            $aA.append(d_n)$
    return $aA$                                                ▷ Ouput
end procedure
```

Definition 10. *We define $\delta(A, B)$ pseudoclosure distance between two subsets A and B of a finite set E:*

$$k_0 = \min(\min\{k | A \subset a^k(B)\}, \infty)$$

$$k_1 = \min(\min\{k | B \subset a^k(A)\}, \infty)$$

$$\delta(A, B) = \min(k_0, k_1)$$

where $a^k(.) = a^{k-1}(a(.))$

Definition 11. *We call $D_A(x)$ interior-pseudo-distance of a point x in a set A:*

$$D_A(x) = \frac{1}{|A|} \sum_{y \in A} \delta(x, y).$$

In case where A and B are reduced to one element x and y, we get the distance $\delta(x, y)$. For clustering documents with k-means algorithm, we use the pseudoclosure distance $\delta(x, y)$ to compute distance between two documents (each document represented by its topic distribution is a point $x \in E$) and the interior-pseudo-distance $D_A(x)$ to compute centroid of A (x_0 is chosen as centroid of A if $D_A(x_0) = min_{x \in A} D_A(x)$).

4.5 Structure analysis with minimal closed subsets

The two limits of the standard *k-means* algorithm are the number of clusters which must be predetermined and the randomness in the choice of the initial centroids of the clusters. Pretopology theory gives a good solution to omit these limits by using the result from minimal closed subsets. The algorithm to compute minimal closed subset is presented in algorithm 4.

Algorithm 4 Minimal closed subsets algorithm.

```
Require: corpus $\mathcal{D}$, pseudoclosure $aA()$
procedure MINIMAL-CLOSED-SUBSETS($\mathcal{D}$, $aA()$)
    compute family of elementary closed subsets $\mathcal{F}_e$
    $\mathcal{F}_m = \emptyset$
    loop until $\mathcal{F}_e = \emptyset$
        Begin
            Choose $F \subset \mathcal{F}_e$
            $\mathcal{F}_e = \mathcal{F}_e - F$
            minimal = True
            $\mathcal{F} = \mathcal{F}_e$
            loop until $\mathcal{F} = \emptyset$ and not minimal
                Begin
                    Choose $G \in \mathcal{F}$
                    If $G \subset F$ then
                        minimal=False
                    Else
                    If $F \subset G$ then
                        $\mathcal{F}_e = \mathcal{F}_e - \{G\}$
                    $\mathcal{F} = \mathcal{F} - G$
                End
        End
    If minimal =True $\&\& F \notin \mathcal{F}_m$ then
        $\mathcal{F}_m = \mathcal{F}_m \cup F$
    return $\mathcal{F}_m$                                    ▷ Ouput
end procedure
```

By performing the minimal closed subset algorithm, we get the family of minimal closed subsets. This family, by definition, characterizes the structure underlying the data set E. So, the number of minimal closed subsets is a quite important parameter: it gives us the number of clusters to use in the *k-means* algorithm. Moreover, the initial centroids for starting the *k-means* process can be determined by using the interior-pseudo-distance for each minimal closed subset $F_{m_j} \in \mathcal{F}_m$ (x_0 is chosen as centroid of F_{m_j} if $D_{F_{m_j}}(x_0) = min_{x \in F_{m_j}} D_{F_{m_j}}(x)$).

4.6 MCPTM algorithm

In this subsection, we present The Method of Clustering Documents using Pretopology and Topic Modeling (MCPTM) which clusters documents via the Topic Modeling and pseudoclosure. At first, an LDA Topic Modeling is learned on the documents to achieve topic-document distributions. The major topic and Hellinger probability distance are used to define relations between documents and these relations are used to define a pretopological space which can be employed to get preliminarily clusters of a corpus and determine the number of clusters. After that, k-means clustering algorithm is used to cluster the documents data with pseudodistance and inter-pseudodistance. The MCPTM algorithm is presented in algorithm 5.

4.7 Implementation in python of the library AMEUR

In this part, we briefly present our *AMEUR* library written in python. *AMEUR* is a project connecting the tools that come from the framework of Pretopology, Topic Modeling, multi-relations networks analysis and semantic relationship. The library is composed of the following modules: *Pretopology*, *topicmodeling* and *nlp*.

Algorithm 5 The MCPTM algorithm: clustering documents using Pretopology and Topic Modeling.

```
Require: $\mathcal{D}$: corpus from set of documents
procedure MCPTM($\mathcal{D}$)
    $\theta_{\mathcal{D}} \leftarrow$ LDA-GIBBS($\mathcal{D}, \alpha, \beta, T$)
    $B \leftarrow$ NEIGHBORHOOD-TM($\mathcal{D}, \theta_{\mathcal{D}}, R_{MTP}, R_{d_H}$)
    $aA \leftarrow$ pseudoCLOSURE($B$)
    $\mathcal{F}_m \leftarrow$ MIMINAL-CLOSED-SUBSETS($\mathcal{D}, aA()$)
    $k = |\mathcal{F}_m|$: number of clusters
    $M = \{m_i\}_{,i=1,\dots,k}$ , $m_i = Centroid(F_{m_i})$
    while clusters centroids changed do
        for each x $\in E - M$ do
            compute $\delta(x, m_i), i = 1, \dots, k$
            find $m_0$ with $\delta(x, m_0) = min\delta(x, m_i)_{i=1,\dots,k}$
            $F_{m_0} = F_{m_0} \cup \{x\}$
        end for
        Recompute clusters centroids M.
    end while
    return $Clusters = \{F_1, F_2, \dots, F_k\}$              ▷ Output
end procedure
```

The *Pretopology* module implements the functions described in section III. The implementation of the Pretopology in the *AMEUR* library allows us to ensures the follow-up of step-by-step processes like dilatation, alliance, pseudoclosure, closure, family of minimal closed subsets, MCPTM and acceptability in multi-relations networks.

The *topicmodeling* module implements generative models like the Latent Dirichlet Allocation, LDA Gibbs Sampling that allows us to capture the relationships between discrete data. This module is used within the *AMEUR* library for querying purposes e.g to retrieve a set of documents that are relevant to a query document or to cluster a set of documents given a latent topic query. These computations of these queries are ensured by the connection between the *topicmodeling* module and the *Pretopology* module.

The *nlp* (natural language processing) module implements the necessary functions for getting unstructured text data of different sources from web pages or social medias and preparing them as proper inputs for the algorithms implemented in other modules of the library.

5 Application and Evaluation

The microblogging service Twitter has become one of the major micro-blogging websites, where people can create and exchange content with a large audience. In this section, we apply the MCPTM algorithm for clustering a set of users around their interests. We have targeted 133 users and gathered their tweets in 133 documents. We have cleaned them and run the *LDA Gibbs Sampling* algorithm to define the topics distribution of each document and words distribution of each topic. We have used then, the *MCPTM* algorithm to automatically detect the different communities for clustering users. We present in the following, the latter steps in more details.

Table 1: Words - Topic distribution ϕ and the related users from the θ distribution

Topic 3					Topic 10				
Words	**Prob.**	**Users**	**ID**	**Prob.**	**Words**	**Prob.**	**Users**	**ID**	**Prob.**
paris	0.008	GStephanopoulos	42	0.697	ces	0.010	bxchen	22	0.505
charliehebdo	0.006	camanpour	23	0.694	people	0.007	randizuckerberg	102	0.477
interview	0.006	AriMelber	12	0.504	news	0.006	NextTechBlog	88	0.402
charlie	0.005	andersoncooper	7	0.457	media	0.006	lheron	71	0.355
attack	0.005	brianstelter	20	0.397	tech	0.006	LanceUlanoff	68	0.339
warisover	0.004	yokoono	131	0.362	apple	0.006	MarcusWohlsen	74	0.339
french	0.004	piersmorgan	96	0.348	facebook	0.005	marissamayer	76	0.334
today	0.004	maddow	72	0.314	yahoo	0.005	harrymccracken	43	0.264
news	0.004	BuzzFeedBen	21	0.249	app	0.005	dens	33	0.209
police	0.003	MichaelSteele	81	0.244	google	0.004	nickbilton	89	0.204

Table 2: Topics - document distribution θ

User ID 02		User ID 12		User ID 22		User ID 53		User ID 75		User ID 83	
Topic	**Prob.**	**Topic**	**Prob.**	**Topic**	**Prob.**	**Topic**	**Prob.**	**Topic**	**Prob.**	**Topic**	**Prob.**
10	0.090	3	0.504	10	0.506	17	0.733	19	0.526	8	0.249
16	0.072	19	0.039	3	0.036	1	0.017	2	0.029	0	0.084
12	0.065	10	0.036	19	0.034	18	0.016	3	0.029	11	0.06
18	0.064	15	0.035	14	0.031	13	0.016	5	0.028	7	0.045
0	0.058	13	0.032	4	0.03	11	0.015	105	0.028	12	0.043

5.1 Data collection

Twitter is a micro-blogging social media website that provides a platform for the users to post or exchange text messages of 140 characters. Twitter provides an API that allows easy access to anyone to retrieve at most a 1% sample of all the data by providing some parameters. In spite of the 1% restriction, we are able to collect large data sets that contain enough text information for Topic Modeling as shown in [14].

The data set contains tweets from the 133 most famous and most followed public accounts. We have chosen these accounts because they are characterized by the heterogeneity of the tweets they posts. The followers that they aim to reach comes from different interest areas (i.e. politics, technology, sports, art, etc..). We used the API provided by Twitter to collect the messages of 140 characters between January and February 2015. We gathered all the tweets from a user into a document.

5.2 Data pre-processing

Social media data and mainly Twitter data is highly unstructured: typos, bad grammar, the presence of unwanted content, for example, humans expressions (happy, sad, excited, ...), URLs, stop words (the, a, there, ...). To get good insights and to build better algorithms it is essential to play with clean data. The pre-processing step gets the textual data clean and ready as input for the MCPTM algorithm.

Table 3: Classifying documents based on their major topic

Major Topic	prob ≥ 0.3	$0.15 <$ prob < 0.3
Topic 0	112,85,104	-
Topic 1	44,129,114	61
Topic 2	101,108,91	90
Topic 3	42,23,12,7,20, 131,96,72	21,81,93,10
Topic 4	125,36,123,0	-
Topic 5	82,126	62
Topic 6	127,37,26	92
Topic 7	118,106,32	70,4
Topic 8	113	83,55,59
Topic 9	67,122	111,100
Topic 10	22,102,88,71,74, 68,76	43,89,33,65
Topic 11	54,51,121	29,94
Topic 12	50	12
Topic 13	16,35	38
Topic 14	31,98	-
Topic 15	66,73,34,	48
Topic 16	99	-
Topic 17	53,30	-
Topic 18	47,128,1,124,5	78,115
Topic 19	14,80,39,75,18,103	-
None	**remaining users (probability** < 0.15)	

5.3 Topic modeling results

After collecting and pre-processing data, we obtained data with 133 documents, 158,578 words in the corpus which averages 1,192 words per document and 29,104 different words in the vocabulary. We run LDA Gibbs Sampling from algorithm 1 and received the output with two matrices: the document-topic distribution matrix θ and the distribution of terms in topics represented by the matrix ϕ. We present in Table 1 two topics from the list of 20 topics that we have computed with our LDA implementation. A topic is presented with a distribution of words. For each topic, we have a list of users. Each user is identified with an ID from 0 to 132 and is associated with a topic by an order of probabilities. The two lists of probabilities in topic 3, 10 are extracted respectively from θ and ϕ distributions. The topic 3 and topic 10 are of particular interest due to the important number of users that are related to them. Topic 3 is about the terrorist attack that happened in Paris and topic 10 is about the international Consumer Electronics Show (CES). Both events happened at the same time that we collected our data from Twitter. We note that we have more users for these topics than from other ones. We can conclude that these topics can be considered as hot topics at this moment.

Due to the lack of space, we could not present in details all the topics with their distribution of words and all topic distributions of documents. Therefore, we presented six topic distributions θ_i (sorted by probability) of six users in the table 2. A high probability value of θ_{ij} indicates that document i is closely related to topic j. Hence, user ID 12 is closely related to topic 3, user ID 22 closely related to topic 10, etc. In addition, documents with low θ_{ij} values over all the topics are noisy documents that belong to none of the topics. So, there is no major topic in user ID 02 (the max probability < 0.15).

We show in Table 3 clusters of documents based on their major topics in two levels with their probabilities. The documents with the highest probability less than 0.15 are considered noisy documents and clustered in the same cluster.

5.4 Results from the k-means algorithm using Hellinger distance

After receiving the document-topic distribution matrix θ from LDA Gibbs Sampling, we used the k-means algorithm with Hellinger distance to cluster users. The table 4 presents the result from the k-means algorithm using Hellinger distance with a number of clusters k=13 and random centroids. Based on the mean value of each cluster, we defined the major topic related to the clusters and attached these values in the table. We notice that different choices of initial seed sets can result in very different final partitions.

Table 4: Result from k-means algorithm using Hellinger distance

Cluster	Users	Major Topic
1	67, 111, 122	TP 9 (0.423)
2	34, 48, 66, 73	TP 15 (0.315)
3	10, 22, 33, 43, 65, 68, 71, 74, 76, 88, 89, 98, 102	TP 10 (0.305)
4	26, 92	TP 6 (0.268)
5	16, 35, 44, 90, 91, 101, 108, 114, 129	TP 2 (0.238)
6	4, 32, 70, 106, 118	TP 7 (0.345)
7	37, 127	TP 6 (0.580)
8	14, 18, 39, 75, 80, 103	TP 19 (0.531)
9	1, 5, 47, 78, 124, 128	TP 18 (0.453)
10	30, 53	TP 17 (0.711)
11	7, 12, 20, 21, 23, 42, 72, 81, 93, 96, 131	TP 3 (0.409)
12	0, 31, 36, 82, 123, 125	TP 4 (0.310)
13	remaining users	None

5.5 Results from the MCPTM algorithm

After getting the results (e.g table 2) from our LDA implementation, we defined two relations between two docements, the first based on their major topic R_{MTP} and the second based their Hellinger distance R_{d_H}. We then built the weak pseudoclosure with these relations and applied it to compute pseudoclosure distance and the minimal closed subsets. With this pseudoclosure distance, we can use the MCPTM algorithm to cluster sets of users with multi-relationships.

Figure 4 shows the number of elements of minimal closed subsets with different thresholds p_0 for R_{MTP} and d_0 for R_{d_H}. We used this information to choose the number of clusters. For this example, we chose $p_0 = 0.15$ and $d_0 = 0.15$ i.e user i connects with user j if they have the same major topic (with probability ≥ 0.15) or the Hellinger distance $d_H(\theta_i, \theta_j) \leq 0.15$. From the network (figure 3) for 133 users built from the weak pseudoclosure, we chose the number of clusters $k = 13$ since the network has 13 connected components (each component represents an element of the minimal closed subset). We used inter-pseudoslosure distance to compute initial centroids and received the result:

$$\{0, 52, 4, 14, 26, 29, 30, 31, 34, 44, 85, 90, 99\}$$

Table 5 presents the results of the MCPTM algorithm and the *k-means* algorithm using Hellinger distance. We notice that there is almost no difference between the results from two methods when using the number of clusters k and initial centroids above.

We saw that the largest connected component in the users network (fig. 3) has many nodes with weak ties. This component represents the cluster 13 with 89 elements. It contains the 8 remaining topics that were nonsignificant

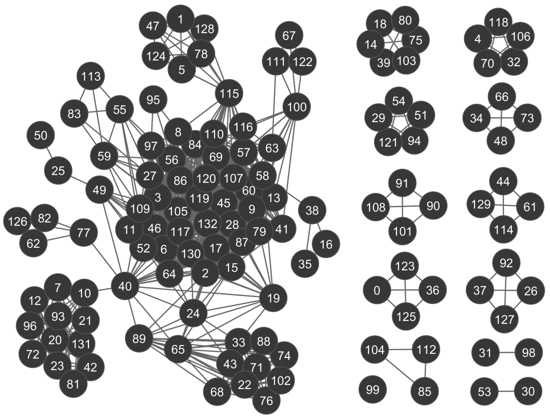


Figure 3: Network for 133 users with two relationships based on Hellinger distance ($distance \leq 0.15$) and Major topic (probability ≥ 0.15).

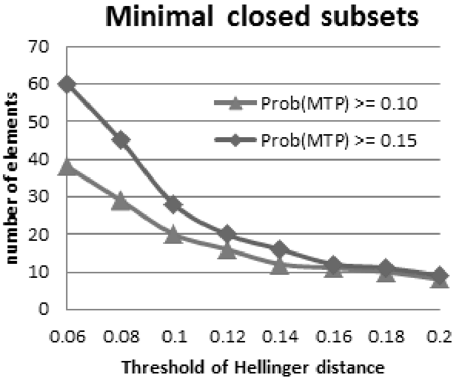


Figure 4: Number of elements of Minimal closed subsets with difference thresholds p_0 for R_{MTP} and d_0 for R_{d_H}.

Table 5: Result from k-means algorithm using Hellinger distance and MCPTM

	K-means & Hellinger		MCPTM Algorithm	
Cluster	**Users**	**Topic**	**Users**	**Topic**
1	0,36,123,125	TP 4 (0.457)	0,36,123,125	TP 4
2	4,32,70,10,118	TP 7 (0.345)	4,32,70,10,118	TP 7
3	14,18,39,75,80,103	TP 19 (0.531)	14,18,39,75,80,103	TP 19
4	26,37,92,127	TP 6 (0.424)	26,37,92,127	TP 6
5	29,51,54,94,121	TP 11 (0.345)	29,51,54,94,121	TP 11
6	30,53	TP 17 (0.711)	30,53	TP 17
7	31	TP 14 (0.726)	31,98	TP 14
8	34,48,66,73	TP 15 (0.315)	34,48,66,73	TP 15
9	44,61,114,129	TP 1 (0.413)	44,61,114,129	TP 1
10	85,104,112	TP 0 (0.436)	85,104,112	TP 0
11	67,90,91,101,108	TP 2 (0.407)	90,91,101,108	TP 2
12	99	TP 16 (0.647)	99	TP 16
13	remaining users	None	remaining users	None

or contains noisy documents without major topics. Hence, we used the *k-means* algorithm with Hellinger distance for clustering this group with number of clusters $k = 9$, centroids:

$$\{23, 82, 113, 67, 22, 50, 16, 47, 2\}$$

and showed the result in the table 6.

Table 6: Result from k-means algorithm using Hellinger distance for cluster 13 (89 users)

Cluster	Users	Major Topic
13.1	7, 12, 20, 21, 23, 42, 72, 81, 93, 96, 131	TP 3 (0.409)
13.2	62, 77, 82, 126	TP 5 (0.339)
13.3	27, 55, 59, 83, 113	TP 8 (0.218)
13.4	67, 111, 122	TP 9 (0.422)
13.5	22, 33, 43, 65, 68, 71, 74, 76, 88, 89, 102	TP 10 (0.330)
13.6	50	TP 12 (0.499)
13.7	16, 35	TP 13 (0.576)
13.8	1, 5, 47, 78, 124, 128	TP 18 (0.453)
13.9	remaining users	None

5.6 Evaluation

In this part of the article, we conducted an evaluation of our algorithm by comparing similarity measure of MCPTM (using the pseudocloseure distance with information from results of minimal closed subsets) and k-means with random choice. The evaluation is performed as follows: we firstly discovered the similarity measure of k-means using three distances: Euclidean distance, Hellinger distance and pseudoclosure distance; we then compared similarity measures among three distances and the similarity measure when we use the number of clusters and the initial centroids from the result of minimal closed subsets. We used the similarity measure proposed by [17] to calculate the similarity between two clusterings of the same dataset produced by two different algorithms, or even the same K-means algorithm. This measure allows us to compare different sets of clusters without reference to external knowledge and is called internal quality measure.

5.6.1 Similarity measure

To identify a suitable tool and algorithm for clustering that produces the best clustering solutions, it becomes necessary to have a method for comparing the different results in the produced clusters. To this matter, we used in this article the method proposed by [17].

To measure the "similarity" of two sets of clusters, we define a simple formula here: Let $C = \{C_1, C_2, \ldots, C_m\}$ and $D = \{D_1, D_2, \ldots, D_n\}$ be the results of two clustering algorithms on the same data set. Assume C and D are "hard" or exclusive clustering algorithms where clusters produced are pair-wise disjoint, i.e., each pattern from the dataset belongs to exactly one cluster. Then the similarity matrix for C and D is an $m \times n$ matrix $S_{C,D}$.

$$S_{C,D} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & \ldots & S_{1n} \\ S_{21} & S_{22} & S_{23} & \ldots & S_{2n} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ S_{m1} & S_{m2} & S_{m3} & \ldots & S_{mn} \end{bmatrix} \tag{12}$$

where $S_{ij} = \frac{p}{q}$, which is Jaccard's Similarity Coefficient with p being the size of the intersection and q being the size of the union of cluster sets C_i and D_j. The similarity of clustering C and clustering D is then defined as

$$Sim(C, D) = \frac{\sum_{1 \le i \le m, 1 \le i \le m} S_{ij}}{\max(m, n)} \tag{13}$$

5.6.2 Discussion

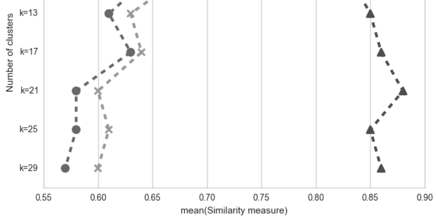


Figure 5: Illustration of the similarity measure where we have the same initial centroids. The appreviation E stands for Euclidean distance, H for Hellinger distance an P for the pseudoclosure distance.

Table 7: The results of the clustering similarity for K-means with different distance measures. The abbreviation E stands for Euclidean distance, H for Hellinger distance (see definition 8) and P for the pseudoclosure distance (see definition 10 and 11).

	Same algorithm			**Same centroids**			**Different centroids**			**Inter-pseudo centroids**		
k	**E**	**H**	**P**	**E vs H**	**E vs P**	**H vs P**	**E vs H**	**E vs P**	**H vs P**	**E vs H**	**E vs P**	**H vs P**
5	0.423	0.454	0.381	0.838	0.623	0.631	0.434	0.373	0.383	-	-	-
9	0.487	0.544	0.423	0.831	0.665	0.684	0.495	0.383	0.447	-	-	-
13	0.567	0.598	0.405	0.855	0.615	0.633	0.546	0.445	0.469	**0.949**	**0.922**	**0.946**
17	0.645	0.658	0.419	0.861	0.630	0.641	0.641	0.493	0.518	-	-	-
21	0.676	0.707	0.445	0.880	0.581	0.604	0.687	0.478	0.491	-	-	-
25	0.736	0.720	0.452	0.856	0.583	0.613	0.715	0.519	0.540	-	-	-
29	0.723	0.714	0.442	0.864	0.578	0.600	0.684	0.4885	0.511	-	-	-
mean	0.608	0.628	0.423	0.855	0.611	0.629	0.600	0.454	0.480	**0.949**	**0.922**	**0.946**

We have compared the similarity measure between three k-means algorithms with different initializations of the centroids and different numbers of clusters k. We plotted the similarity measure between the clusters computed with the three *k-means* algorithms with the same initial centroid in Figure 5 and the three k-means algorithms with different initial centroids in Figure 6.

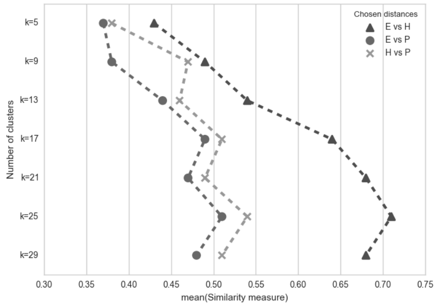


Figure 6: Illustration of the similarity measure where we have diffirent initial centroids. The appreviation E stands for Euclidean distance, H for Hellinger distance an P for the pseudoclosure distance.

We notice that in the both figures, the Euclidean Distance and the Hellinger distance have higher similarity measure. This is due to the fact that both distances are similar. In Figure 5, we see a big gap between the clusters of Euclidean distance, Hellinger distance and the clusters from Pseuoclosure distance. This gap is closing in Figure 6 and starts opening again from $k = 17$. With a different initial centroids the pseudoclosure distance closed the gap between the k-means algorithms using Euclidean and Hellinger distance. But, when $k > 13$, the number of closed subsets, the gap between the pseudoclosure and the other distances starts opening again. In table 7 where we applied the same algorithm twice, the similarity measure between two clusters results from k-means is low for all three distances: Euclidean, Hellinger, pseudoclosure distance. The different choices of initial centroids can result in very different final partitions.

For k-means, choosing the initial centroids is very important. Our algorithm MCPTM offers a way to compute the centroids based on the analysis of the space of data (in this case text). When we use the centroids computed from the results of minimal closed subsets that we present in Table 5, we have the higher similarity: 0,949 for Euclidean vs Hellinger; 0,922 for Euclidean vs pseudocloure and 0,946 for Hellinger vs pseudoclosure. It means that the results from k-means using the centroids $\{0, 52, 4, 14, 26, 29, 30, 31, 34, 44, 85, 90, 99\}$ is very similar with all three distances Euclidean, Hellinger, pseudoclosure. We can conclude that the result that we obtained from our MCPTM algorithm is a good result for clustering with this Twitter dataset.

6 Conclusion

The major finding in this article is that the number of clusters and the chosen criterias for grouping the document is closely tied to the accuracy of the clustering results. The method presented here can be considered as a pipeline where we associate Latent Dirichlet Allocation (LDA) and pseudoclosure function. LDA is used to estimate the topic-distribution of each document in corpus and the pseudoclosure function to connect documents with multi-relations built from their major topics or Hellinger distance. With this method both quantitative data and categorical data are used, allowing us to have multi-criteria clustering. We have presented our contribution by applying it on microblogging posts and have obtained good results. In future works, we want to test these results on large scale and more conventional benchmark datasets. And we intend also to parallelize the developed algorithms.

References

[1] C. C. Aggarwal and C. Zhai. A survey of text classification algorithms. In *Mining text data*, pages 163–222. Springer, 2012.

[2] M. Ahat, B. Amor S., M. Bui, S. Jhean-Larose, and G. Denhiere. Document Classification with LSA and Pretopology. *Studia Informatica Universalis*, 8(1), 2010.

[3] Z. Belmandt. *Basics of Pretopology*. Hermann, 2011.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[5] A. Ferligoj and V. Batagelj. Direct multicriteria clustering algorithms. *Journal of Classification*, 9(1):43–61, 1992.

[6] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.

[7] G. Heinrich. Parameter estimation for text analysis. Technical report, 2004.

[8] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[9] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[10] T. K. Landauer and S. T. Dutnais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, pages 211–240, 1997.

[11] C. Largeron and S. Bonnevay. A pretopological approach for structural analysis. *Information Sciences*, 144(1–4):169 – 185, 2002.

[12] V. Levorato and M. Bui. Modeling the complex dynamics of distributed communities of the web with pretopology. *Proceedings of the 7th International Conference on Innovative Internet Community Systems*, 2007.

[13] C. D. Manning and P. Raghavan. An Introduction to Information Retrieval, 2009.

[14] F. Morstatter, J. Pfeffer, H. Liu, and K. M. Carley. Is the sample good enough? comparing data from twitter's streaming API with twitter's firehose. *arXiv:1306.5204 [physics]*, June 2013. arXiv: 1306.5204.

[15] M. Steyvers and T. Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

[16] N. K. Thanh Van Le and M. Lamure. A clustering method associating pretopological concepts and k-means algorithm. *The 12th International Conference on Applied Stochastic Models and Data Analysis*, 2007.

[17] G. J. Torres, R. B. Basnet, A. H. Sung, S. Mukkamala, and B. M. Ribeiro. A similarity measure for clustering and its applications. *Int J Electr Comput Syst Eng*, 3(3):164–170, 2009.

Appendix

List of Notations:

Notation	Meaning
K	number of topics
V	number of words in the vocabulary
M	number of documents
M	number of documents
$\mathcal{D}$	corpus
$\phi_{j=1,\ldots,K}$	distribution of words in topic j
$\theta_{d=1,\ldots,M}$	distribution of topics in document d
$a(A)$	pseudoclosure of A
$F(A)$	closure of A
$\mathcal{F}_e$	family of elementary closed subset
$\mathcal{F}_m$	family of minimal closed subset
$\delta(A, B)$	pseudodistance between A,B
$D_A(x)$	interior-pseudodistance of x in A
$MTP(d)$	major topic of document d
$d_H(P, Q)$	Hellinger distance between P, Q
R_{MTP}	relationships based on major topic
R_{d_H}	relationships based on Hellinger distance
k	number of clusters

A Distributed Algorithm for Monitoring an Expanding Hole in Wireless Sensor Networks

Khanh-Van Nguyen
Hanoi University of Science and Technology, Vietnam
E-mail: vannk@soict.hust.edu.vn

Phi Le Nguyen
Department of Informatics, The Graduate University for Advanced Studies, Japan
E-mail: nguyenle@nii.ac.jp

Hau Phan
Hanoi University of Science and Technology, Vietnam
E-mail: pvhau93@gmail.com

Trong D. Nguyen
Iowa State University, USA
E-mail: trong@iastqae.edu

Keywords: wireless sensor networks, hole detection, expanding hole

Received: April 15, 2016

Holes in sensor networks are regions that have no operating nodes and that may occur due to several reasons, including cases caused by natural obstacles or disaster suffered areas. Determining the location and shape of holes can help to monitor these disasters (such as volcano, tsunami, etc.) or help to make smart, early routing decisions for circumventing a hole. There are many hole determination algorithms proposed in the literature, however, these only consider the networks with static holes i.e. with stable boundary nodes. Moreover, most of these are designed in a centralized manner which is not suitable to the unstable situation of networks with an expanding hole. In this paper, we propose an algorithmic scheme not only for determining the initial shape but also for monitoring and quickly reporting about the area of a hole gradually expanding. Our algorithms are designed in a distributed manner and our initial simulation results show that our protocol is lightweight and feasible with monitoring sensor networks with an expanding hole.

Povzetek: Razvita je nova metoda za obravnavo lukenj v brezžičnih omrežjih.

1 Introduction

Let start by considering an example scenario of a forest wherein fire rather frequently occurs in summer. Since it is very large and difficult to access and monitor by human, one thinks of using helicopters to disseminate thousands of wireless sensor nodes through out this forest area for deploying a fire monitor system. Due to the cost limitation, these pieces are equipped with just a temperature and humidity sensing device, a very simple processing unit and a small radio communication module that allows them to communicate with the others in their communication range. The mission of these sensor nodes is to monitor the environment, detect the fire and report to a central host via some special sink nodes (with more powerful resource). Paradoxically possibly when the fire occurs, the sensor nodes in the area of an on-going fire can be destroyed and thus cannot perform their monitor task and fail to report about the fire. To solve this problem, a common approach is letting the alive nodes surrounding the fire areas detect the boundary of the fire area fast enough and report to the central host. This scenario belongs to a well-known problem in sensor network which is called *hole detection and determination.*

The hole detection-determination (HDD) problem is not only used to detect the events that are being monitored as described above but also is, and in fact known more as, an important technical issue in geographic routing. Geographic routing [1][2] which exploits the local geographical information at the sensors is popular for its simplicity and efficiency. The traditional geographic routing strategy works well and can achieve the near-optimal routing path length in networks without holes, however with the occurrence of network holes, the path length can grow as much as $\theta(c^2)$ where c is the optimal path length [3]. In order to solve the problem of path enlargement, a common approach is to determine the hole boundary, describe it by a simpler shape whose information is disseminated to the surrounding area. This information will help to establish a hole awareness and mechanism to find short detour routes [4][5][6][7][8].

There are many HDD algorithms proposed however, these just consider the problem of detecting a typically static hole (hole whose boundary never changes). Moreover, almost of these algorithms are conducted in the centralization manner and is time ineffective. A very common approach is based on the possibility of *having a monitor packet to travel a full route around the hole, which could be unlikely possible in our problem scenario – the network hole can be expanding fast enough that could destroy any attempt to send a packet around in one simple loop*.

In this paper, we propose a novel approach to detect and regularly report about the hole boundary, that to the best of our knowledge is *the first one that targets expanding holes* (whose boundary change continually). Especially, we aim at scenarios where holes can occur and grow rapidly; i.e., we aim to design fast algorithms for detecting and monitoring hole boundaries. Our main idea is actually to distribute this task of capturing the dynamic hole perimeter into the hands of several boundary nodes and network cells that are carefully selected to take this joint responsibility.

In our protocol, the network is divided into small grid squares by a given grid. The squares (cells) are to be divided into three groups: black, gray and white. A black square is one which already belongs to or is heavily affected by a hole, a gray square is not yet touched by any hole but is detected being close enough and getting soon affected by an expanding hole, and a white is a currently safe one. A white square can later get turned into a gray and then a black square when facing an expanding hole, which can be seen as a set of the black squares. Each square in the black and the gray group is monitored by a local node, called a pivot, which is elected from the sensors in that square. These pivots monitor the status of the sensors and the squares and help to manage the square status changes.

The contribution of this paper are threefold:

- We propose a distributed hole detection algorithm which can determine a hole boundary efficiently with respect to time and energy saving.
- We propose a distributed hole updating algorithm which monitors the expansion of the hole boundary and reports regularly to the central host.
- We conduct initial experiments by simulation to analyze our protocol and evaluate its feasibility and advantages.

The remainder of the paper is organized as follows. Our hole detecting and updating protocols are proposed in section 2. Section 3 evaluates the performance of our protocol using simulation experiments. We also discuss further on related work in section 4 and finally conclude the paper in section 5 with further concerns about open issues.

A preliminary version of this paper appears in [12]. Besides a significant revision effort for improving the clarity, formality and preciseness we also add new substantial elements to this full version. That is we reshape and adjust some important parts in our proposed algorithm and thus make it more efficient; as a result, we redo the evaluation task but in a larger scale for obtaining a more insightful about the performance of our algorithm. Below we briefly mention the most important adjustments. In section 2.1, for the task of detecting a hole boundary we justify our decision to choose the approach used in [10] over the one in [26]. In section 2.2 we optimize the way we define and use Bitmap Presentation for reducing incurred communication overhead and delay. In 2.3 we tune the process of forwarding the hole boundary info towards the sink(s): the pivots fully involve in this process that increases the efficiency of the whole mechanism.

Compared to the previous version, moreover, we also extend our evaluation work to a significant deeper level where we redesign a new, significant larger set of simulation scenarios, inspired by new observations and thus obtain new findings. Most notably, our 3 main simulation settings (for studying the effects of 3 system parameters: Dead Node Threshold, Notification Threshold & Report Threshold) are all extended by our new deployment where we use two separate scripts to simulate an expanding hole: the Fast Expansion and the Slow Expansion scripts. We focus more on the Fast Expansion scenario (a hole expands fast for simulating a forest fire) and we identify some value region of the Dead Node Threshold that could optimize our algorithm. We also extend our study on these main parameters with some initial consideration of the relationship with another variable that is the grid cell size. In spite of this rather extensive evaluation analysis there still remain many unknowns that can be challenging enough for good results in future work.

2 Our hole monitor scheme

2.1 Scheme overview

Our goal is to detect and update the hole as fast as possible so that our hole monitor scheme can beat the expanding speed of a hole. Therefore, we aim to determine the approximate shape of the hole boundary rather than its exact shape. The approximate shape of the hole boundary is determined via a set of the unit squares of a given grid (with certain predefined unit length) which are affected by the hole, i.e. ether intersecting the hole boundary or staying inside the hole.

Our hole monitor scheme consists of two protocols: the hole boundary detection protocol, denoted by the HBD protocol, and the hole boundary update protocol, denoted by the HBU protocol. Let us discuss the basics of the HBD protocol first. The main idea is to have some nodes on the hole boundary detect that they are on a hole's boundary. This initial awareness helps to start the process of learning about the whole hole shape by arranging a special monitor packet to travel around the hole: this packet is being forwarded from one boundary node to another as a neighbor.

Technically, we know of two main approaches in detecting a hole boundary, one from the research works on geographic routing in WSNs e.g. in [10], and one from the research work on coverage hole e.g. [26]. We have done certain probing experiments and found out that the former approach appears more efficient (we show our experimental evaluation on this in section 3) and thus, we choose to use this approach for designing our HBD protocol where we aim at approximating the hole shape in a distributed manner.

Based on the above mentioned approach, our HBD protocol is conducted by using the *stuck nodes* introduced in [10]: by definition, a node u is considered a stuck node if there exists a position w outside u's transmission range such that u is closer to w than all the neighbor nodes of u [1]. Note that a stuck node always stays on a hole boundary but not all the boundary nodes are stuck node. Each stuck node can detect its status as being stuck (using the TENT rule [10]) and then creates a hole-monitor packet, which contains a HBA message (denoted for Hole Boundary Approximation), and forwards it to the next neighbor, boundary node (determined by the right hand rule [10]). The mission of this HBA message is to record the approximate shape of the path it has gone through: at each boundary node, the HBA gets updated about the new, just traveled edge. Upon reaching the next stuck node, a HBA message has fulfilled its mission by capturing the approximate shape of the boundary segment between this pair of successive stuck nodes, which will be forwarded to the nearest sink node (via a cell pivot node that we will introduce later). Note that each boundary segment (a chunk between two successive stuck nodes) is determined and approximated by using a separate HBA message. The sink node combines information received from all these stuck nodes (via the pivot nodes) and sends the whole info to the central node.

We now discuss the HBU protocol. Because the holes can enlarge quickly we want to have each update action performed as fast as possible to reflect well with the changes. Therefore, instead of determining the changes after happening we predict and update the changes of the hole boundary based on the prediction. Specifically, we monitor the status of the nodes (i.e. of being alive or dead) in the unit grid squares (also called cells) around the hole boundary. When the amount of the dead nodes of a given cell exceeds a predefined threshold, this cell is predicted to belong the hole soon and thus the hole area gets updated by adding that cell. To make the mechanism distributed, the task of monitoring within each unit square is performed by a so-called pivot node of this cell. The cell pivot is a sensor node (by default, the one closest to the center of the square) that is elected from all the nodes locating inside the cell.

During the HBD protocol, when a HBA message is about to fulfill its mission (and then stop being forwarded), the receiving stuck node can initiate the election of the pivot node of its cell if not selected yet, then forward the HBA info to this pivot.

2.2 Bitmap representation

The main idea of our approximation mechanism is to describe the status of the cells (unit grid squares) in the network area by a bitmap representation where each bit would reflect if a corresponding cell belongs to a hole or not. More specifically, let $m\&n$ denote the width and length of the network; we use a grid with the edge length (of the unit squares) a that divides the network into $([\frac{m}{a}]+1)\times([\frac{n}{a}+1])$ cells. We use a two dimension array $bmp[.][.]$ to represent the status of the cells, where $bmp[i][j]$ corresponds to the unit grid square whose center has the coordinates $(1/2+i)a;\ (1/2+j)a$ and $bmp[i][j]=1$ if the corresponding cell belongs to a hole (i.e. this cell stays inside this hole or intersects its boundary) or else, $bmp[i][j]=0$.

During the execution of the HBD protocol, the chunks of a hole's boundary (boundary segments between two successive stuck nodes) are approximated and these approximation info pieces are recorded into the HBA messages, which then are later forwarded to the sink(s) for being combined. To facilitate this computing mechanism, below we formally define bitmap representations as data structures to keep data at the HBA messages as well as at the pivots and the sink(s).

Definition 1 (Bitmap representation)**.** *For a line segment l on the plane, the bitmap representation of l is defined as a two dimension array $bmp_l[.][.]$, where $bmp_l[i][j]=1$ if and only if l intersects the unit grid square whose center's coordinates are $((1/2+i)a,(1/2+j)a)$; $bmp_l[i][j]=0$, otherwise.*

We call the bitmap representation of the whole network a two dimension array bmp, where $bmp[i][j]=1$ if and only if the unit grid square whose center's coordinates $((1/2+i)a,(1/2+j)a)$ belongs to a hole and $bmp[i][j]=0$, otherwise.

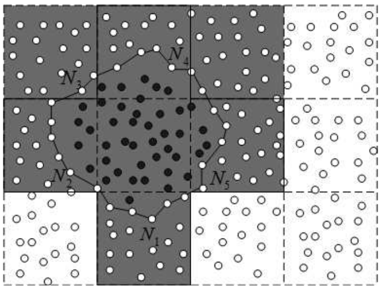


Figure 1: Illustration of bitmap representation

Fig. 1 illustrates a network with a big hole. The unit grid squares which intersect the hole are colored black. The bitmap representation of line segment N_1N_2 is

[1] Thus, a task of forwarding to destination w would gets 'stuck' at node u if greedy routing is being used

$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ and the bitmap representation of the network is $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$.

In the HBD algorithm, the bitmap representations of the boundary segments are determined by using HBA messages and sent to the local cell pivots, which forward these info pieces to the sinks. During the HBU algorithm, the pivots of the newly alarmed cells (who have been alarmed about the expanding hole after the HBD's execution) keep monitoring the hole and reports its status to the sinks as a black cell (belonging to the hole) when the number of the dead nodes inside exceeds a predefined threshold. The sinks maintain their bitmap representations (as about the network from their viewpoints) and periodically report them to the central node, which can combine these pieces into the bitmap representation of the whole network by simple bitmap XOR operations.

2.2.1 Forwarding mechanism with compact hole info

The main goal of our scheme is to monitor the hole(s) and update the sink(s) with any new status of the network shape. Our algorithm scheme is fully distributed where several nodes independently and concurrently capture info about pieces of a hole's boundary and forward towards the sink(s) via the cell pivots – the intermediate hubs. The bitmap presentations of the network are only formed at the sink(s), however "pieces" of that being formed and forwarded at node- and pivot- levels, at which sending a hole bitmap would be too much a luxury to afford (by a piece we mean a chunk of adjacent nodes on the hole boundary). In fact, the bitmap presentation of a line segment, or even a full hole's boundary (a collection of segments) can be compactly described, i.e. specifically, stored into memory as follows. Starting with an end vertex of the line segment (or any of the hole boundary polygon) we store the full coordinates (indexes) of its cell and then continually check the adjacent vertices for newly separate cells, per each of which store just two bits for describing its relative position to the preceding cell [2]. By this mechanism, even a complex full hole boundary can be compactly described (as for reflecting the bitmap presentation of the network) by just the coordinates of a cell plus a rather short binary string (just a few bytes) as illustrated in 2.

2.3 Hole boundary detection algorithm

This section discusses further details of our HBD algorithm.

At the initial, all the network nodes detect if each is a stuck node by using the TENT rule described in [10]. Each stuck node then creates its HBA message and sends it to the left boundary neighbor node (using the right hand rule [10]) which then forwards it towards the next stuck node. Here we use the forwarding mechanism as suggested in [10] but the HBA message is customized to reflect bmp_s – the bitmap representation of the boundary segment between the two successive stuck nodes. When a node N on the hole boundary receives the HBA message, it determines the cells intersecting the line segment connecting N and the previous boundary node by the algorithm suggested in [11] and updates bmp_s and the HBA message accordingly. The details are described in algorithm 1.

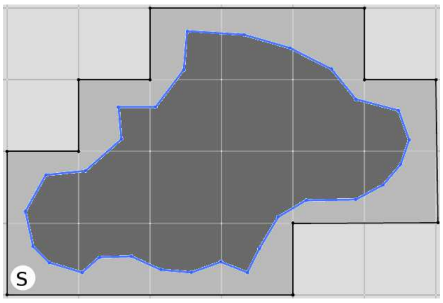


Figure 2: Compact Hole Info
This hole can be reflected by the coordinate of cell S plus a string of 24 bits to locate the remaining 12 cells at the hole boundary.

In the deployment of our HBD algorithm, to reduce size of data transferred between nodes, we do not store in the HBD packet this full bitmap representation but instead use the compact form as discussed in section 2.2.1, which would take just a few bytes in a HBA packet.

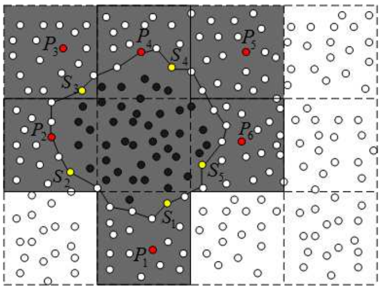


Figure 3: Illustration of hole boundary detecting algorithm

To prepare for the hole boundary update (HBU) phase, the HBD algorithm not only detects the hole boundary but also determines the pivots of the cells intersecting the hole boundary. Such a pivot is determined by the following algorithm. Each node u receiving a HBA message determines the cell(s) which intersect the line segment connecting u and the previous boundary node v but does not contain v; there may exist none, one, or more such cell; for each such a cell if exists, u sends a pivot election message towards this cell's center. Note that, such a message then stops at the node w where none of its neighbors is nearer to the cell center than w, i.e w can see it as elected as the cell pivot. Each pivot then broadcasts a pivot announcement

[2] It only needs two bits to describe 4 possible directions that may involves left, right, up and down

Algorithm 1 Algorithm to update compression form of bitmap representation bmp_{comp}

```
Require: P: HBA packet; $N_c(x_C, y_C)$: current node and its coordinate; $N_P(x_P, y_P)$: previous node and its coordinate
Ensure: updated $bmp_{comp}$
$bmp_{comp} \leftarrow$ the compression form from packet $P$
$tmp \leftarrow$ transfering $bmp_{comp}$ into bitmap
$U \leftarrow$ set of the unit grid squares which intersect $N_C N_P$
for all the unit grid square $U_i(x_i, y_i)$ in $U$ do
    $tmp[\lfloor \frac{x_i}{a} \rfloor][\lfloor \frac{y_i}{a} \rfloor] \leftarrow 1$
    if $(\lfloor \frac{x_i}{a} \rfloor \neq \lfloor \frac{x_P}{a} \rfloor)$ $and$ $(\lfloor \frac{y_i}{a} \rfloor \neq \lfloor \frac{y_P}{a} \rfloor)$ then
        Send a election packet to $U_i(x_i, y_i)$.
    end if
end for
$bmp_{comp} \leftarrow$ transfer $tmp$ into compression form
update $P$ with new $bmp_{comp}$
```

message to all the nodes in the cell.

Fig. 3 illustrates the algorithm. In the figure, the yellow nodes $S_1, S_2, \ldots, S_5$ found themselves as stuck nodes using the TENT rule. They initiate and forward the HBA messages along the boundary (using the right hand rule to determine the next boundary neighbor) until each arriving at the next stuck node. So, the HBA message initiated by S_i is forwarded until arriving at S_{i+1} wherein this message captures the bitmap representation of the line segment of the boundary between S_i and S_{i+1}. For example, the bitmap representation of the segment between S_1 and S_2 is $\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$. After the HBD algorithm finishes, the approximate shape of the hole as well as the pivots of the cells surrounding the hole are determined. In Fig. 3, the approximate shape of the hole is colored black and the pivots are colored red.

2.4 Hole boundary update algorithm

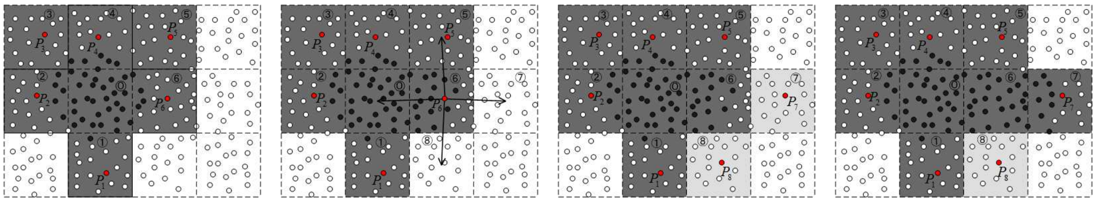
We discuss further details on the HBU protocol. As mentioned above, se call an unit grid square, i.e. a network cell, belonging to a hole (i.e. staying inside the hole or intersecting its boundary) a *black* one. After the HBD phase finishes, each black cell has a pivot elected. All nodes in these black cells notify their status (i.e. whether alive or not) to its cell's pivot periodically.

When the ratio of the number of the dead nodes (over the total) of this cell exceeds a predefined threshold, which we call *Notification Threshold*(NTT), the situation with this expanding hole is seen serious. Thus, the cell's pivot sends an alert message to each of the four neighbor cells which share an edge with it to notify about the high possibility that they would soon get affected by the hole, i.e. these four are considered *gray* (becoming black). Note that some of these neighbors may have already been black cells wherein this alert simply gets ignored. Also, the destination of such an alert is set to be the center of the targeted gray cell and thus, the last node receiving the alert is elected as the pivot of this gray cell.

When the ratio of the number of dead nodes in a black cell exceeds a predefined threshold, which we call *Report Threshold* (RPT), the cell is considered severely damaged by the hole and thus, this dangerous state gets reported to the nearest sink by the cell pivot.

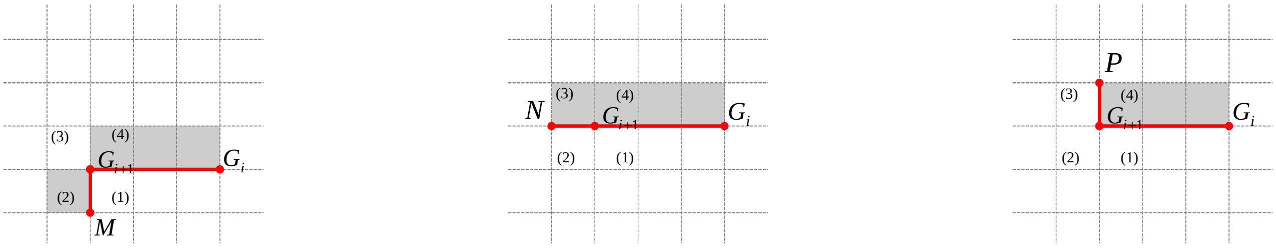
Similarly to the pivot of a black cell, the pivot of a gray cell also monitors the status of all the nodes within the cell. When the ratio of the number of the dead nodes of this cell exceeds the NTT, the cell situation gets serious and thus, it turns into a black one. At this same time the cell's pivot sends an alert to its four neighbor cells (notification) to make any white remainder of them to become *gray*. The details of these algorithmic operations executed at the white, gray, and black cells are described in algorithms 2a, 2b and 2c, respectively.

Fig. 4 illustrates an example. In this figure, the network is divided into 4×3 unit grid squares. The black nodes are dead and others are alive. After conducting the HBD algorithm, the hole boundary has been detected and the cells that belong the hole are colored black as shown in Fig. 4(a). The red nodes $P_1, P_2, \ldots, P_6$ represent the pivots of the black cells. Suppose that, the hole is expanding to the right, then after sometime some nodes in square (6) die, which is enough to make pivot P_6 send an alert to the neighbor squares as shown in Fig. 4(b). Among these neighbors, (0) and (5) are already black which ignore the alert; the others i.e. squares (7) or (8) are still white, so become gray. The cell center nodes P_7 and P_8 then become the pivots of (7) and (8), respectively – Fig. 4(a). When the ratio of the number of dead nodes in (7) exceeds NTT, it is considered black and P_7's pivot sends alerts to the neighbor cells – Fig. 4(d).



(a) (b) (c) (d)

Figure 4: An example of hole boundary updating process



(a) a(2) = 1 (b) a(2) = 0 and a(3) =1 (c) a(2) = 0 and a(3) = 0

Figure 5: Algorithm to determine hole boundary from bitmap representation

Algorithm 2a HBU process at nodes in a white cell ($state = WHITE$) when receives a message

```
Require: $P$: received message, $C$: the current node
Ensure:
if $P$ is an alert message then
    $state \leftarrow GRAY$
    if $C$ is the closest node of center then
        $C$ takes responsibility of the cell pivot
        Broadcast a notification to all nodes in the cell
    else
        if $C$ has not received a pivot notification then
            Forward $P$ to the center of cell (Vote the closest
            node of center as the pivot node)
        end if
    end if
else if $P$ is a notification message then
    Be informed that the source of message as the
    elected pivot
else
    Send HELLO packet
end if
```

Algorithm 2b HBU work at nodes in a gray cell ($state = GRAY$)

```
Require: $C$: current node
Ensure:
loop periodically
    if $C$ is a pivot then
        $d \leftarrow$ the number of dead nodes in the cell
        if $d > NTT$ then
            $state \leftarrow BLACK$
            Send alert packet to 4 neighbor cells
        end if
    else
        Send HELLO packet to update node's state
    end if
end loop
```

Algorithm 2c HBU work at nodes in a black cell ($state = BLACK$)

```
Require: $C$: current node
Ensure:
loop periodically
    if $C$ is a pivot then
        $d \leftarrow$ the number of dead nodes in the cell
        if $d > RPT$ then
            Report to the nearest sink
        end if
    else
        Send HELLO packet to update node's state
    end if
end loop
```

2.5 Hole information combination

At the central node, we obtain the bitmap representation of the whole network. Although this bitmap gives us a view of location of the holes, in some applications (e.g. in geographic routing) geometric expressions (i.e. polygonal shapes) of the holes are required. We now describe an algorithm to determine the boundary of a hole approximate shape as a polygon, called an A-polygon, which satisfies the following condition:

- This A-polygon vertices are the vertices of the grid squares which intersect the hole
- All the centers of the unit grid squares belonging to the hole stay inside this A-polygon

We will compute the (coordinates of) the vertices of this A-polygon in the clockwise order. The following fact helps to show the process during each step for obtaining the next vertex; this fact is quite simple so we omit the proof.

In fact 1, we consider the scenario where we have just determined that G_i and G_{i+1} are two consecutive vertices of the A-polygon (in the clockwise order), and we need to determine the next vertex. Let (1), (2), (3), (4) denote the unit squares that has G_{i+1} as a vertex and let M, N and P denote the adjacent vertices of G_{i+1} in (2), (3) and (4), respectively as shown in Fig. 5. Let $(x_i,\ y_i)$ be the coordinates of the center of square (i) and $\alpha(i)$ be the corresponding bit value of square (i) in the network bitmap, then:

Fact 1. *Having G_i and G_{i+1} just determined as the vertices of an A-polygon from the given network bitmap, the next vertex of this A-polygon (in the clockwise order) can be determined as follows:*

- *M if and only if $\alpha(2) = 1$ (Fig. 5(a)).*
- *N if and only if $\alpha(2) = 0$ and $\alpha(3) = 1$ (Fig. 5(b)).*
- *P if and only if $\alpha(2) = 0$ and $\alpha(3) = 0$ (Fig. 5(c)).*

Using fact 1, we come up with algorithm 3 below to determine an A-polygon of a hole from the network bitmap. It is easy to observe that for a given hole the vertex of the unit square (intersecting it) with the lowest y-coordinate must be a vertex of the A-polygon.

Algorithm 3 Determining A-polygon from array bmp

```
Require: bitmap $bmp$; $(x_A, y_A)$: coordinates of boundary node A that has lowest $y - coordinate$
Ensure: $G$: the set of vertices of the A-polygon
Denote $r$ as the edge length of the unit square
$G \leftarrow G \bigcup \left(\lfloor \frac{x_A}{r} \rfloor r + r, \lfloor \frac{y_A}{r} \rfloor r\right)$
$G \leftarrow G \bigcup \left(\lfloor \frac{x_A}{r} \rfloor r, \lfloor \frac{y_A}{r} \rfloor r + r\right)$
while the top element of $G \neq \left(\lfloor \frac{x_A}{r} \rfloor r + r, \lfloor \frac{y_A}{r} \rfloor r\right)$ do
    $U \leftarrow$ the top element of $G$
    $(x_u, y_u)$ is the coordinate of U
    $V$ is the previous element of U
    $(x_v, y_v)$ is the coordinate of V
    //the following code is for the case when $y_u = y_v$, the other case ($x_u = x_v$) is similar
    $i \leftarrow \lfloor \frac{x_v}{r} \rfloor; j \leftarrow \lfloor \frac{y_v}{r} \rfloor - 1$
    if $bmp[i-1][j] = 1$ then
        $G \leftarrow G \bigcup (x_v, y_v - r)$
    else
        if $bmp[i-1][j+1] = 1$ then
            $G \leftarrow G \bigcup (x_v - r, y_v)$
            $G \leftarrow G \setminus (x_v, y_v)$
        else
            $G \leftarrow G \bigcup (x_v, y_v + r)$
        end if
    end if
end while
```

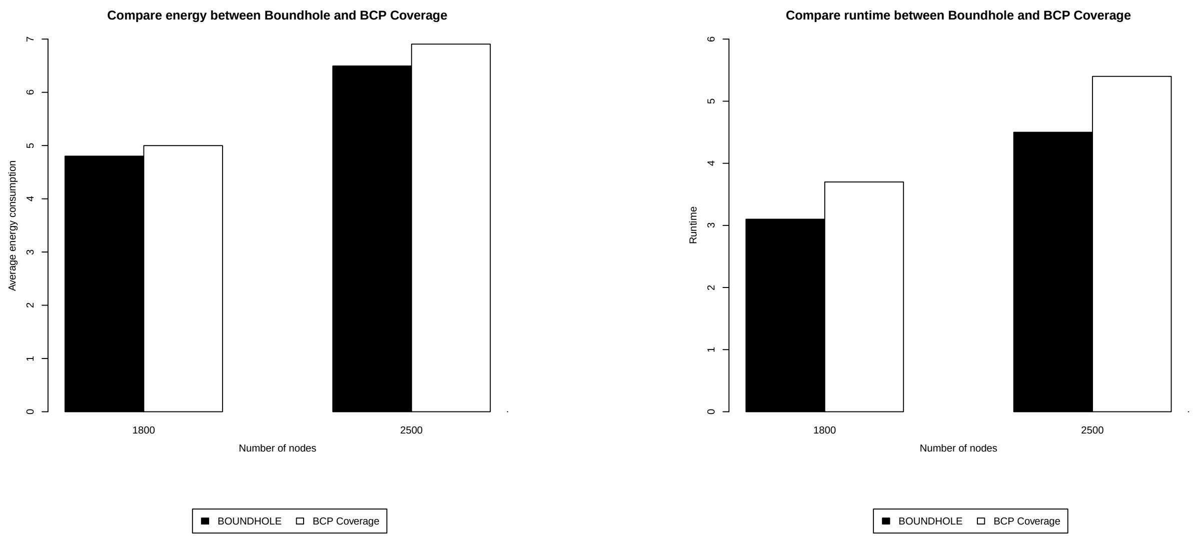
3 Performance evaluation

3.1 Comparison between the approaches in detecting hole boundary

As we mentioned above in section 2.1, we now compare by simulation experiments between the two HBD mentioned approaches, i.e. the Boundhole approach from [10] and the BCP approach from [26] to find out which suits better to our main goal: we need our HBD algorithm to be as fast as possible and to consume energy as less as possible (to help with monitoring expanding holes). [3] In this evaluation, we run the experiments with 2 network models: 1800 nodes and 2500 nodes. Table 1 summarizes the parameters which are suggested by [13]:

The results of experiments are evaluated by these metrics: average consumed energy and running time (the lower is better). We can see in Fig 6, the Boundhole approach gives better result with slightly less consumed energy and running time. It appears that the Boundhole uses simpler operations while the BCP uses a bit too many floating-point operations (especially, when computing circle-circle intersections).

[3] Note that here we do not compare the original algorithms from the two mentioned papers, instead we compare two versions of our HBD algorithm each of which is based on each of these two approaches.



(a) Comparison of consumed energy

(b) Comparison of running time

Figure 6: Comparison between the BoundHole and the BCP approaches

Variables	*Values*
Communication range	40m
Voltage	3V
Receiving current	15mA
Transmitting current	29.5mA
Idle current	3.2mA
Number of nodes	29.5mA
Period of HELLO packet	30s
Simulation time	1050s

Table 1: Simulation parameters for comparing the BoundHole and BCP approaches

3.2 Experiment deployment and settings

We have proposed our very first algorithmic solution to deal with a possibly new and challenging problem of hole expansion in WSN. Bellow we show our initial results in evaluating our algorithm through experiments by simulation. Here we propose to use the following metrics to evaluate our algorithm performance:

- The *Approximation Ratio* (AR) is the ratio between the area that our algorithm approximately describes of a hole and the true area of this hole.
- The *Death Report Error* (DRE) reflects how well we monitor the dead nodes, which is calculated as $\frac{|RN-TN|}{TN}$ where RN is the number of nodes reported as dead by our algorithm and TN is the true number of the dead nodes.
- The *Consumed Energy* (CE) of network which is computed as the average energy consumed by any node in the network.

More specifically, we study the effect of choosing the following parameters, crucial in our algorithm, in achieving good performance by the above mentioned metrics.

- *Dead Node Threshold* (DNT): All the alive nodes periodically sends the HELLO packets to their neighbor nodes, hence in our algorithm after waiting for a time period called DNT if a node does not receive the next HELLO from a certain neighbor, this node can decide that this neighbor is dead.
- *Notification Threshold* (NTT): As described in section 2, the pivot of a black/gray square decides to send a NOTIFY packet to each of its 4 neighbor squares (to notify about the hole expansion) when the ratio of the number of dead nodes (per the total) in its square exceeds a NTT.
- *Report Threshold* (RPT): The pivot of a black square decides to send a REPORT packet to the nearest sink when the ratio of the number of dead nodes in this square exceeds a RPT.

We run the experiments on the ns-2 simulator (802.11 as MAC protocol). Table 2 summarizes the parameters which are suggested by [13]. In our simulation, about 2500 sensor nodes are deployed randomly in an area of $1400m \times 1400m$.

We conduct three different simulation settings to evaluate the effect of these parameters on three metrics described above. Also in our simulation settings, we deploy two different scripts of a large hole expanding: one is called Fast Expansion (EF), where we focus on, and the other

Variables	*Values*
Communication range	40m
Voltage	3V
Receiving current	15mA
Transmitting current	29.5mA
Idle current	3.2mA
Network size	$1400m \times 1400m$
Number of nodes	2500
Period of HELLO packet	30s
Simulation time	1050s for FE 2500s for SE

Table 2: Simulation parameters for our protocol experiments

Slow Expansion (SE). In each script we program so that the hole gradually expands from a chosen center point toward a fixed given shape, which is a random set of points with distances to the center varying randomly from a minimum value (300 m) to a maximum value (450 m). The hole reaches its maximum, final shape in 1000 seconds in our FE script, and 2500 seconds in our SE script. We believe that these scripts would reflect reasonably to the expansion of a typical forest fire.

3.3 Simulation results

1. *Effect of Dead Node Threshold:* We conducted the simulation with four values of the Dead Node Threshold (DNT): 1x, 1.3x [4], 2x, 3x, where nx means the DNT equals n times of the HELLO period (at which the HELLO packets being periodically sent from a node to a neighbor).

Fig. 7 and 8 show the simulation results of our first simulation settings (with the Fast Expansion script) on the effect of DNT on the Death Report Error (DRE) and the Approximation Ratio (AR), respectively. It can be seen that the DRE tends to decrease with the increasing of the time values during the simulation. That is, for all the different DNT values, the trend is that the performance keeps improving (DRE gets closer to 0% and AR gets closer to 1) along the the simulation process (better performance for captures at later simulation moments - larger timestamps).

With the case of 1x, the DRE is very large at first: it can be explained as because the HELLO packets and responses may get delayed or stuck (and dropped) at some nodes; therefore, when the threshold is too short, the nodes whose HELLO packets are delayed or dropped are wrongly judged as dead nodes (so DRE very large, initially). However later the HELLO packets (and its responses) can be retransmitted and thus, the nodes which have been judged as dead can be reconsidered as alive and so the DRE decreases.

The simulation results also shown that $FN - TN$, the difference between the number of reported dead nodes and the true number of dead nodes, tends to decrease when the DNT increases. Especially, with the DNT $=$ 2x or larger $FN - TN$ is even negative, i.e. the number of the dead nodes determined by our algorithm is less than the true number of real dead nodes. This is because, when the DNT increases, the pivots tend to become dead too early to send alerts about its cell situation to the neighbor cells, thus many dead nodes may not be recognized. It can be seen that, the DRE in case of 1.3x is closest to 0 and thus, parameter DNT $=$ 1.3x can be considered (near) the best for maintaining low DRE.

Fig. 8 represents the relation between the approximation ratio (AR) and the DNT. Similarly as with the DRE, the AR decreases with the increasing of the DNT. The AR with the DNT $=$ 1x is quite greater than 1 and the AR with the DNT as 2x/3x is often/always (much) smaller than 1. It can be seen that the hole area determined by our algorithm is always much larger than the real hole with the DNT as 1x and always much smaller than the real hole with the DNT $=$ 3x (or quite smaller with 2x). For the later phase of hole expansion, both 1.3x and 2x seem competing for the best AR (but in both sides of 1). Overall, it can be seen that a small enough DNT value may lead to a high DRE because of many alive nodes being wrongly judged as dead while a large enough DNT value may make unreasonably small AR because of many dead nodes being unrecognized. For our experiment setting with the FE script, the DNT$=$ 1.3x seems (near) the optimum option.

So far we have just analyzed the simulation with the FE script, where the considered hole expands to the maximum distance $450m$ (from the center) in only $1000sec$, we now take a quick look at the results with the SE script (reaching maximum distance in $2500s$). As can be seen in table 3 and table 4, the general trend is that the performance keeps improving (DRE gets closer to 0% and AR gets closer to 1) along the the simulation process for most of cases. Most notably, the choice of DNT$=$ 2x results in poor AR while still quite good at keeping low DRE. For keeping AR close to 1, the ideal DNT value should be between 4x and 5x, however with this range the DRE is rather high (compared to that of DNT$=$ 2x). This is because for a larger DNT the network acts slower with the hole expansion and hence, a number of new dead nodes cannot get reported fast enough. Thus, in fact for the SE script we face a complexity in choosing good DNT (further work needed in future work).

2. *Effect of Notification Threshold:* The effect of the NTT on the Consumed Energy (CE) and Approximation Ratio (AR) is shown in Fig. 9. In this simulation setting with the FE script we fix DNT$=$ 1.3x (as seen best from simulation results above). Another parameter needs taken into account is the size of the unit grid square which we choose within $80m - 120m$ (ranges in small multiples of the sensor transmission range, $40m$).

It is clear that, the consumed energy becomes lesser for

[4] A HELLO packet may be sent a bit later than expected due to some random delay in processing; thus, we take this into account by using DNT$=$ 1.3x.

	Death Report Error (%)			
Timestamp - S.moment (s)	1x	1.3x	2x	3x
560	44.94	17.98	35.58	75.66
620	27.49	15.41	37.76	76.13
680	17.17	12.37	30.3	71.46
740	13.95	10.52	33.26	72.75
800	7.33	11.54	34.07	72.89
860	1.3	10.73	32.03	77.89
920	7.05	7.78	33.92	73.57
980	1.49	6.79	28.13	65.9
1040	3.72	4.52	21.94	60.51

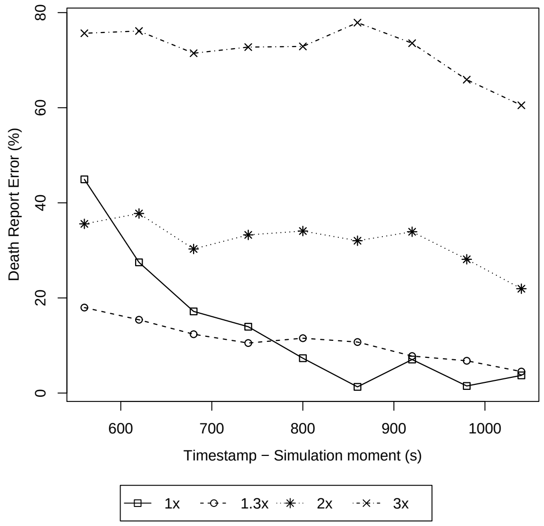


Figure 7: Effect of DNT on the Death Report Error

	Approximation Ratio			
Timestamp - S.moment (s)	1x	1.3x	2x	3x
560	1.6674	1.1672	0.9588	0.0834
620	1.6422	1.1632	0.9238	0.1711
680	1.4422	1.1826	0.8941	0.2019
740	1.3117	1.089	0.8167	0.2227
800	1.2255	1.141	0.8663	0.2113
860	1.1099	1.1099	0.9218	0.2822
920	1.1422	1.1251	0.8865	0.2898
980	1.1289	1.1289	0.9408	0.3136
1040	1.1815	1.1815	1.0127	0.3529

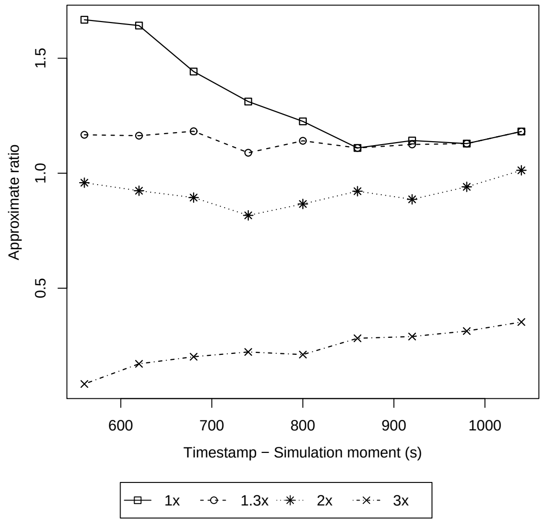


Figure 8: Effect of DNT on Approximation Ratio

larger NTT as well as for smaller grid square (cell) sizes. That is for a larger NTT, the pivots of the black cells need to wait longer to notify its neighbor squares, i.e. the squares become gray later rather than sooner, lessening the monitoring work and hence, reducing the CE incurred due to our monitor mechanism. Also, for smaller cell sizes, obviously the total area being monitored is also smaller and thus, reducing the consumed energy as well.

The effect of the NTT on the AR is reported in Fig. 9(b). It can be seen that the AR decreases with the increasing of the NTT. It can be explained as below. For smaller NTT values, the notification of the expanding hole is done earlier and hence, the gray area tends to get large earlier, i.e. the AR tends to increase. For AR greater than 1, the approximate area of the hole is larger than the real hole area. When the NTT is larger enough (from about 15%) the AR can be smaller than 1 (for grid size 100m or less). This is because the times to send NOTIFY packets may come too late (with a speed slower than the expansion speed of the hole) and thus, many black pivots may get dead before the time to notify their neighbor cells. Consequently, some should-be-black squares get unrecognized and the approximate hole area becomes smaller than the real hole area.

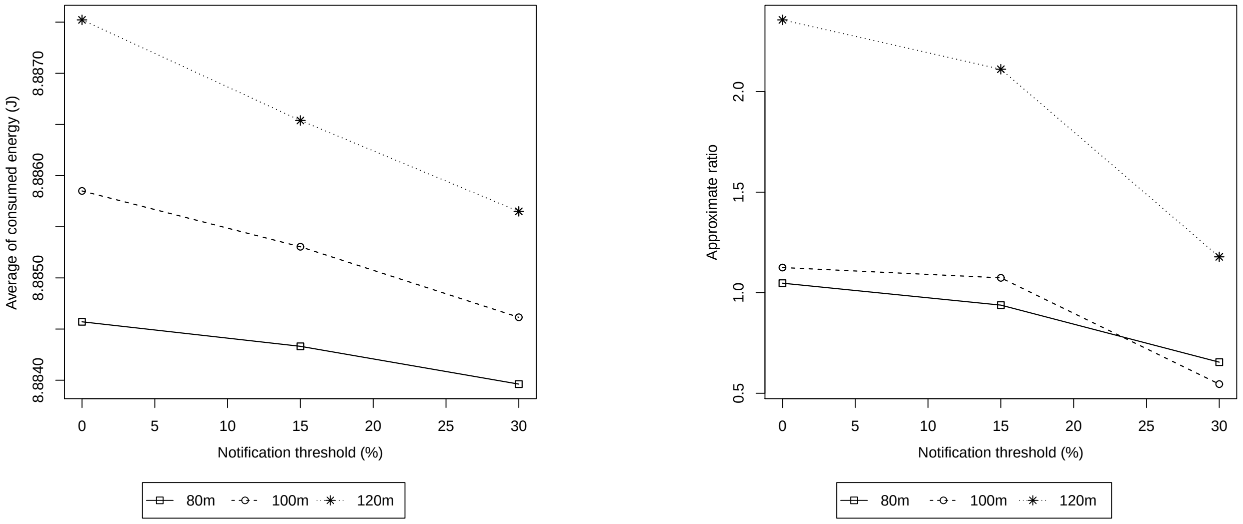
It can be remarked that the NTT should be selected in accordance with the speed of the expansion of the hole. When the hole expands fast, we should choose a small NTT. Par-

	Death Report Error (%)			
Timestamp - S.moment (s)	2x	4x	5x	6x
760	18.75	31.25	39.58	45.83
1000	14.63	28.05	31.71	36.59
1240	15.27	26.72	31.3	38.93
1480	11.29	21.51	25.81	37.1
1720	8.8	20	24.8	35.6
1960	8.18	17.3	22.01	35.53

Table 3: Effect of DNT on DRE with the Slow Expansion script

	Approximation Ratio			
Timestamp - S.moment (s)	2x	4x	5x	6x
760	1.6393	1.4572	1.0929	0.7286
1000	1.2247	1.1134	1.002	0.8907
1240	1.3219	1.175	0.9547	0.7344
1480	1.2502	1.1958	1.0327	0.7066
1720	1.2382	1.1144	0.9493	0.7429
1960	1.3046	1.1742	0.9785	0.6849

Table 4: Effect of DNT on AR with the Slow Expansion script



(a) Effect of NTT on Consumed Energy (with 3 different cell size-settings)

(b) Effect of NTT on Approximation Ratio (with 3 different cell size-settings)

Figure 9: Effect of Notification Threshold (NTT) with FE script

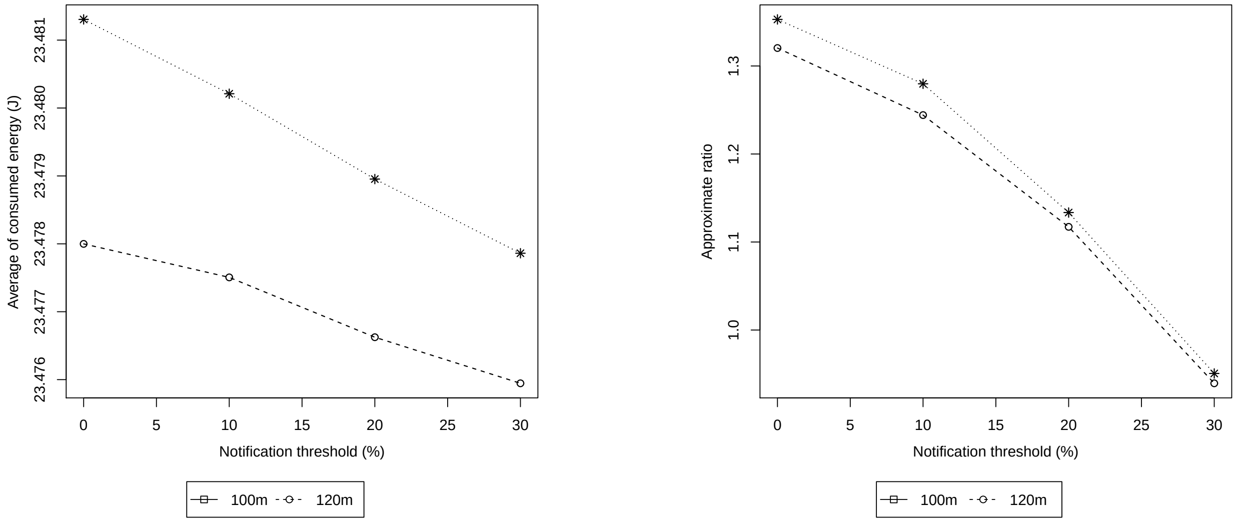
ticularly, in this simulation setting with the FE script, the best NTT is about 15%.

For our simulation with the SE script (see Fig.10), the AR becomes close to 1 for NTT about $25-30\%$, larger than the setting with the FE script, which is an advantage for keeping the gray squares in smaller number. However, the CE looks quite higher to that in the setting with the FE script. This is because the gray/black squares live longer in this SE setting and hence more monitor traffic would get incurred. Of course, one would consider to use a smaller grid size but this would result in need of low NTT which is a disadvantage. Thus, again we face some complexity in choosing proper parameter values that needs further consideration in our future work.

Fig. 11 illustrates the growth of the real hole area (represented by the orange line) and the approximate hole area determined by our algorithm (represented by green line) with images captured at 4 different time values. This also illustrates the same general trend as before: the performance keeps improving (AR gets closer to 1) along the the simulation process.

3. *Effect of Report Threshold:* The simulation result in this aspect is shown in Fig. 12 where we also fix DNT = 1.3x and we evaluate only the AR because the energy consumption is not affected much by the RPT. It can be seen that the AR tends to decrease when the RPT increases; especially if the RPT $\leq$ 30%, the AR becomes $\geq$ 1, i.e. the approximate hole area is larger than the real hole area. However if the RPT is $\geq$ 35%, the AR becomes less than 1.

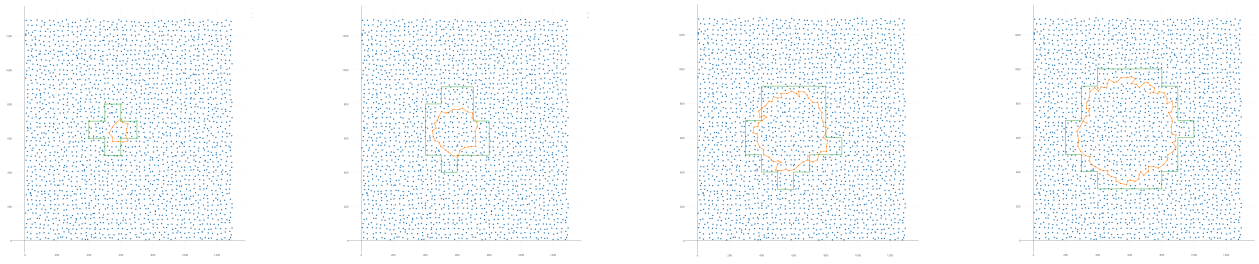
This is because, for RPT small enough the pivots report to the sink about the approximate hole soon enough (after discovering a small enough fraction of the dead node in the their squares), thus the approximate hole area, which is the area defined by the black squares (called the black area), tends to contain the real hole area and hence, the AR is greater than 1. The smaller is the RPT, the larger is the difference between the black area and the real hole are and



(a) Effect of NTT on Consumed Energy (with 2 different cell size-settings)

(b) Effect of NTT on Approximation Ratio (with 2 different cell size-settings)

Figure 10: Effect of Notification Threshold(NTT) with the SE script



(a) Captured at 120s

(b) Captured at 360s

(c) Captured at 600s

(d) Captured at 840s

Figure 11: The growth of the approx. and the real hole areas (DNT= 1.3x; NTT= 15%; RPT= 15%)

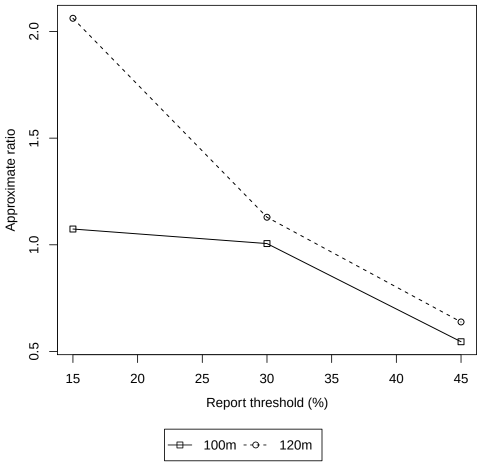


Figure 12: Effect of RPT on the Approx. Ratio

thus, the larger is the AR. However, for RPT large enough (e.g. $\geq 35\%$), some pivots may get dead before the time to send a REPORT packet to the sink and thus, their black squares are not reported to the sink, i.e. the AR becomes less than 1.

Similarly as with the NTT evaluation, it can be concluded that the RPT should be selected in accordance with the speed of expansion of the hole. For example, in this simulation setting, the best RPT is about 30%.

4 Related work

An algorithmic approach for locating holes in WSN has been firstly introduced in [10] where Fang *et al.* propose a procedure to obtain the exact boundary of the hole as a polygon with vertices being adjacent nodes on the hole side. The basic idea is to have a packet traveling around the hole, learning the position of the nodes on its side. Other follow-up works [27, 9, 28] study a generalized problem in computing the exact shape of the network as collection of boundaries, i.e. the outer perimeter of the whole network area and boundaries of inside holes. In [11], we introduce an efficient and flexible approach and techniques to approximate the shape of a hole.

However, this approach is based on the possibility of having a packet to travel a full route around the hole, which could be unlikely possible in our problem scenario – the network hole can be expanding fast enough that could destroy any attempt to send a packet around in one simple loop. Our main solution is actually to distribute this task of capturing the hole perimeter: several HBA packets (sent by the stuck nodes) and later, the cell pivots are responsible to pick up a collection of segments of the hole perimeter.

There are many HDD algorithms have been proposed which can be classified into three categories: geometric methods, statistical methods and topo-logical methods. Geometrical approach uses the coordinates of the nodes and standard geometric tools (such as the Delaunay or Voronoi diagrams) to detect coverage holes and their boundaries. Several simple distributed algorithms have been proposed in [29, 30, 31] to detect the boundary nodes (i.e. *non-covered sensor* in our definitions) and build the routes around holes. In [33], Zhang *et al.* proposed an algorithm to detect the hole boundary nodes on the basis of Voronoi Diagram. The authors also described a method to calculate the accurate location of hole boundary by analyzing the sensing edges of the boundary nodes. In the topological approach, there are neither coordinates nor localization of nodes are required. Instead, this approach use topological properties such as the information of connectivity to identify the boundary nodes; e.g. Ghrist *et al.* [32] proposed a purely connectivity-based coverage hole detection method. Recently, Chu *et al.* [35] exploited information of three-hop neighboring nodes and propose a distributed protocol to identify sensor nodes nearby the hole. In this protocol, the hole is determined on the basis of the so-called *2-hope-neighbor graph* maintained at each sensor node. Although it is said that, the protocol can detect all hole boundaries with a small overhead, it is not suitable for our expanding hole scenario because it requires all nodes running the protocol on their *2-hope-neighbor graph* to detect the expanded hole.

Routing hole is a critical issue in geographic routing where data packets are forwarded based on the positional information of the sensor nodes (assuming that they are equipped with GPS devices). Early approaches are based on greedy and perimeter routing where with the former a packet is forwarded to the 1-hop neighbor that is closest to the destination. However, greedy forwarding can lead into the local minimum phenomenon whenever the packet encounters a routing hole. To bypass such a hole, the traditional schemes appropriately switch between greedy and perimeter forwarding modes (initiated by the GPSR proposal in [2]), in the later of which the data packets are forwarded along the hole boundary [2][14][15][16][17]. These proposals require a specific embedding of a planar graph (e.g. Gabriel Graph). Recently, Yu *et al* [34] proposed a scheme to relieve the local minimum problem by allowing a node in a concave area of a hole to mark itself as a potential stuck node and thus do not participate in data delivery. Through this policy, the packets are prevented from entering the concave area of the hole and thus can avoid the long detour path problem.

There are quite a few papers in geographic routing which propose to learn about the hole shape then disseminate this hole info to the surrounding area for supporting routing tasks performed later. This approach as we call learn-and-disseminate strategy has been used in e.g. [25][5][4], however, the main aim there is to achieve a bounded route stretch in their routing algorithms. As a result, although achieving a desired route stretch, these schemes can incur heavy extra communication in broadcast and significant load imbalance due to congestion on these major routes. For example, in [4], the holes are compactly described as a set of extreme points of the convex hull covering the boundary nodes and thus, the hole detour will be along these points, that may cause the traffic concentration around the boundary of the convex hull.

We briefly review the two mentioned (in section 2.1) approaches in detecting a hole boundary, i.e. the work in [10] for supporting geographic routing in WSNs and the work in [26] for solving the problem of wireless hole coverage. The BOUNDHOLE algorithm [10] can be shortly described as follows. Each stuck node p initiates a HBD message (denoted for Hole Boundary Detection) which includes its location and sends it to p's closest neighbor node with respect to the stuck angle. The closest neighbor can be defined as follows: consider the stuck angle at p (facing the hole area by this angle) if we use the angle's bisector line to conduct counter-clockwise sweeping then the closest neighbor is the first one that is met by the sweeping line. Upon receiving the HBD message, this newly identified hop writes its location into the message and passes it to the next-hop in the same manner mentioned. The HBD message will finally come back to the origin as basically, the HBD message creates a closed cycle of traveling.

The coverage hole detection algorithm in [26] is based on the concept of Boundary Critical Points (BCPs). A boundary critical point is defined as the intersection point of two sensing circles that cannot be covered by any other sensing circles. Each sensor node first computes its BCPs. Note that the sensors that are not along the boundary of any coverage holes cannot have any boundary critical point. Then each consecutive boundary critical points is connected by constructing an boundary line along the border of the nodes having those boundary critical points. The construction of boundary line is continued until the starting boundary critical point is revisited or border of the monitoring region is touched.

5 Conclusion and future work

In this paper we have proposed an algorithmic scheme for detecting, determining and monitoring the shape and boundary of an expanding hole in sensor networks. Our algorithms are designed in a distributed manner to help their surviving of the expanding of a hole and reporting regularly about the hole status.

We have also conducted the simulation experiments to evaluate the effects of the important thresholds (DNT, NTT, RPT) on the performance metrics of our algorithms. The simulation results show that the performance metrics are strongly dependant on how suitably we choose the mentioned threshold parameter for a given network setting. Obviously, the thresholds should be selected in accordance with the characteristics of the network, especially the expansion speed of the considered hole.

It is obvious that although we have done some complicated experiments which require some heavy simulation work there still remain many unanswered interesting questions and important technical issues. There are 3 important threshold parameters yet there still are other important parameters such as the grid square size or the speed of the hole expansion; thus, it seem challenging to mix the selection of these to a good effect. Most challenging, perhaps is the concern that the hole expansion speed is normally unforeseeable and hence, we face a big trouble: a parameter set that is nicely suitable to a fast expansion setting can perform very poorly in a slow expansion setting. Thus, it is natural to think of further work with improved algorithms where the important parameters such as the grid size can be adjusted dynamically during the execution process.

Acknowledgement

This work was partially supported by a grant from Hanoi University of Science and Technology under the project titled "Detecting Holes and Monitoring Their Growth in Wireless Sensor Networks" (PI: Nguyen Khanh Van).

References

[1] Y. Ko and N. H. Vaidya (1998) Location-Aided Routing (LAR) in Mobile ad hoc Networks, *Proc. of MOBICOM'98.*

[2] B. Karp and H. T. Kung (2000) GPSR: Greedy Perimeter Stateless Routing for Wireless Networks, *Proc. of MOBICOM'00*, pp. 243–254.

[3] Fabian Kuhn, Roger Wattenhofer, Yan Zhang, and Aaron Zollinger (2003) Geometric ad-hoc routing: of theory and practice, *PODC*, pp. 63–72.

[4] Myounggyu Won, Radu Stoleru, and Haijie Wu (2011) Geographic routing with constant stretch in large scale sensor networks with holes, *Proc. of WiMob*, pp. 80–88.

[5] Z. Zheng, K. W. Fan, P. Sinha, and Y. Wang (2008) Distributed roadmap aided routing in sensor networks, *5th IEEE International Conference*, pp. 347–352.

[6] Y. Tian et al (2008) Energy-Efficient Data Dissemination Protocol for Detouring Routing Holes in Wireless Sensor Networks, *Proc. of IEEE Intl. Conf. on Communications, ICC'08*, pp. 2322–2326.

[7] F. Yu et al (2008) Efficient Hole Detour Scheme for Geographic Routing in Wireless Sensor Networks, *Proc. of the 67th IEEE Vehicular Technology Conference, VTC'08*, pp. 153–157.

[8] M. Choi and H. Choo (2011) Bypassing Hole Scheme Using Observer Packets for Geographic Routing in WSNs, *Proc. of Intl. Conf. on Information Networking, ICOIN'11*, pp. 435–440.

[9] Yue Wang, Jie Gao, and Joseph S.B. Mitchell (2006) Boundary recognition in sensor networks by topological methods, *5MobiCom'06.*

[10] Q. Fang, J. Gao, and L. J. Guibas (2004) Locating and Bypassing Routing Holes in Sensor Networks, *Proc. of INFOCOM'04.*

[11] Le Nguyen, Quan Bui, Hieu Nguyen, and Khanh-Van Nguyen (2011) Efficient approximation of routing holes in wireless sensor networks, *ACM Proc. of the Second Symposium on Information and Communication Technology.*

[12] Trong Nguyen, Le Nguyen, Hau Phan and Khanh-Van Nguyen (2015) A Distributed Protocol for Detecting and Updating Hole Boundary in Wireless Sensor Networks, *ACM Proc. of the Second Symposium on Information and Communication Technology(SoICT 2015).*

[13] Victor Shnayder, Mark Hempstead, Bor rong Chen, Geoffrey Werner-Allen, and Matt Welsh (2004) Simulating the power consumption of large scale sensor network applications, *SenSys*, ACM, pp. 188–200.

[14] P. Bose, P. Morin, I. Stojmenovir, and J. Urrutia (2008) Routing with guaran-teed delivery in ad hoc wireless networks, *5th IEEE International Conference*, pp. 347–352.

[15] F. Kung, R. Wattenhofer, and A. Zollinger (2002) Asymptotically Optimal Geometric Mobile Ad-hoc Routing, *Dial-M.*

[16] F. Kung, R. Wattenhofer, and A. Zollinger (2003) Worst-Case Optimal and Average-Case Efficient Geometric Ad-hoc Routing, *Proc. of ACM MobiHoc.*

[17] F. Kung et al (2003) Geometric Ad-hoc Routing: Of Theory and Practice, *Proc. of ACM PODC*.

[18] S. Subramatian, S. Shakkottai, and P. Gupta (2007) On Optimal Geographical Routing in Wireless Networks with Holes and Non-Uniform Traffic, *Proc. of IEEE INFOCOM*.

[19] Piyush Gupta and P. R. Kumar (2000) The capacity of wireless networks, *IEEE Transactions on Information Theory*, pp. 388–404.

[20] A. Rao, C. H. P., S. Shenker, and I. Stoica (2003) Geographic Routing without Location Information, *Proc. of ACM MOBICOM 2003*, pp. 96–108.

[21] Q. Fang, J. Gao, L.J. Guibas, V. de Silva, and L. Zhang (2005) Glider: Gradient Landmark-based Distributed Routing for Sensor Networks, *Proc. of IEEE INFOCOM*, pp. 339–350.

[22] R. Fonseca, S. Ratnasamy, J.Zhao, C.T. Ee, D.E. Culler, S. Shenker, and I.Stoica (2007) Beacon Vector Routing: Scalable Point-to-point Routing *Wireless Sensornets*.

[23] A. Caruso et al (2007) GPS Free Coordinate Assignment and Routing in Wireless Sensor Networks *INFOCOM*, pp. 150–160.

[24] R. Kleinberg (2007) Geographic Routing Using Hyperbolic Space, *IEEE INFOCOM*, pp. 1902–1909.

[25] G. Tan, M. Bertier, and A.-M. Kermarrec (2009) Visibility-graph-based shortest-path geographic routing in sensor networks, *Proc. of IEEE INFOCOM*, pp. 1719–1727.

[26] Zhiping Kang, Honglin Yu and Qingyu Xiong (2013) Detection and Recovery of Coverage Holes in Wireless Sensor Networks, *JOURNAL OF NETWORKS, VOL. 8, NO. 4, APRIL 2013*, pp. 822–828.

[27] A. Kroller and S. P. Fekete and D. Pfisterer and S. Fischer (2006) Deterministic Boundary Recognition and Topology Extraction for Large Sensor Networks, *Proc. of SODA '06, the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, 2006.

[28] S. Lederer and Y. Wang and J. Gao (2009) Connectivity-based Localization of Large-scale Sensor Networks with Complex Shape, *ACM Transactions on Sensor Networks (TOSN)*.

[29] Yen-Wen Chen Hwa-Chun Ma, Prasan Kumar Sahoo (2011) Computational geometry based distributed coverage hole detection protocol for the wireless sensor networks, *J. Network and Computer Applications*, pp. 1743–1756.

[30] E. D. Zhao and Y. T. Lv J. Yao, H. Wang (2011) A coverage hole detection method and improvement scheme in wsns, *the International Conference on Electric Information and Control Engineering (ICEICE 11)*, pp. 985–988.

[31] S. Commuri M. Watfa (2006) Energy-efficient approaches to coverage holes detection in wireless sensor networks, *the International Conference on Electric Information and Control Engineering (ICEICE 11)*, pp. 131–136.

[32] Robert Ghrist and Abubakr Muhammad (2005) Coverage and hole-detection in sensor networks via homology, *Proceedings of the 4th international symposium on Information processing in sensor networks*, IEEE Press, pp. 34.

[33] Yunzhou Zhang, Xiaohua Zhang, Zeyu Wang, Honglei Liu (2013) Virtual Edge Based Coverage Hole Detection Algorithm in Wireless Sensor Networks, *IEEE Wireless Communications and Networking Conference (WCNC): NETWORKS*, pp. 1488–1492.

[34] Fucai Yu, Shengli Pan, and Guangmin Hu (2015) Hole Plastic Scheme for Geographic Routing in Wireless Sensor Networks, *IEEE ICC 2015 - Ad-hoc and Sensor Networking Symposium*, pp. 6444–6449.

[35] Wei-Cheng Chu, Kuo-Feng Ssu (2012) Decentralized Boundary Detection without Location Information in Wireless Sensor Networks, *IEEE Wireless Communications and Networking Conference: Mobile and Wireless Networks*, pp. 1720–1724.

Protected Elastic-tree Topology for Survivable and Energy-efficient Data Center

Dieu-Linh Truong, Elena Ouro and Thanh-Chung Nguyen
School of Information and Communication Technology
Hanoi University of Science and Technology, Vietnam
E-mail: linhtd@soict.hust.edu.vn, elenaouro@gmail.com, ntchung92@gmail.com

Keywords: data center, energy efficiency, survivability, elastic tree

Received: January 1, 2016

Recently, using third party data centers has become a popular choice for storing enterprise data and deploying services. The fast growth of data centers in size and in number makes them become huge energy consumption points. Up to 70% of energy consumption is due to server running and cooling. In order to reduce energy consumption, recent researches have proposed turning off certain switches in data centers with little traffic flow. However, when those switches are turned off, the data center becomes vulnerable to failures due to low connectivity between servers. In order to overcome this weakness, this paper proposes to use path protection to ensure that all connections in the data center retain survivability upon any single failure. The paper also proposes an algorithm to calculate a tailored topology for the data center so that unnecessary switches can still be turned off. The simulation results show that the proposed solution makes data centers survivable while still saving energy significantly, mostly in big size data centers.

Povzetek: Nov algoritem poišče primerne varne rešitve v podatkovnih centrih pri varčevanju z energijo na osnovi elastične topologije.

1 Introduction

Cloud computing is currently the common choice for end users and enterprises to store their data and process their services by using third party data centers. End users and enterprises can now focus on their business issues without concerning themselves with the building and maintaining of their own storage servers or network devices since this infrastructure is hosted in data centers. This advantage leads to a quick growth of data centers in size and in number. However, these data centers consume a huge amount of energy for running servers, network devices in order to process user requests and transmit data between servers within data centers. According to US Federal Energy Management Program report in [1], data centers in 2013 accounted for 2.7% of the 3831 billion kWh used in the US. Federal data centers used about 5 billion kWh in 2013, or nearly 10% of federal electricity use. These numbers show the importance of utilising energy efficiently in data centers. It is stated in [2] that cooling infrastructure of data center facilities can require one to two times the energy used to power the IT equipment itself. Therefore, many researches look for efficient energy utilisation solutions for data centers either by locating data centres close to green energy resources, using energy efficient devices [3], or by limiting the number of running devices in data center.

Servers in a data center interconnect through a system of switches. Commonly, these switches are organised in hierarchical topology with 3 layers: a core layer, an aggregation layer and an edge layer. The switches in the edge layer link to servers (hosts). In upstream direction, traffic from source servers arrives at edge switches then is regrouped and routed to the aggregation layer switches. In turn, the aggregation layer switches regroup traffic to forward to a core switch. Traffic follows the downstream direction similarly to get to the destination servers.

Fat-tree topology [4] was proposed for data centers for the first time in 2008 [5]. A k-ary Fat-tree contains k modular data centers called PODs (Performance Optimized Data center) that links together through core layer switches. Each POD contains two layers of aggregation and edge switches where each switch of one layer links to all switches of the other layer. Figure 1 shows an example of 4-ary Fat-tree data center. Fat-tree structure has been developed to reduce the over-subscription ratio and to remove the single point of failures of the hierarchical architecture in the data center network through multiple-linking of a switch to other layers.

Fat-tree data centers tend to consume a lot of energy, however. Data centers in general, and Fat-tree data centers in particular, are usually designed with sufficiently high capacity to tolerate access traffic in peak hours, meanwhile the access traffic varies significantly within a day, a week, or month. As a result, data centers often work under capacity during normal hours. It is wasteful if all devices in a data center are kept working during this time.

With the objective to save energy a solution, Elastic tree, has been proposed in which the main idea is to turn off some devices in Fat-tree [6] when they are not needed. In

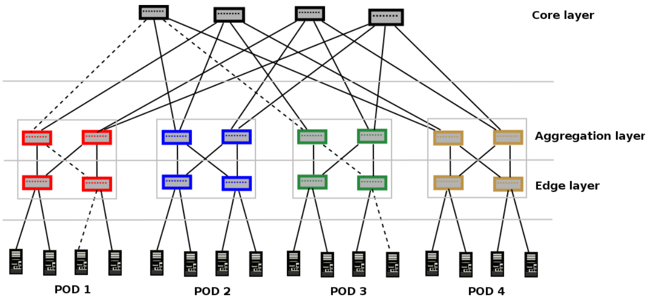


Figure 1: A Fat-tree data center with $k = 4$.

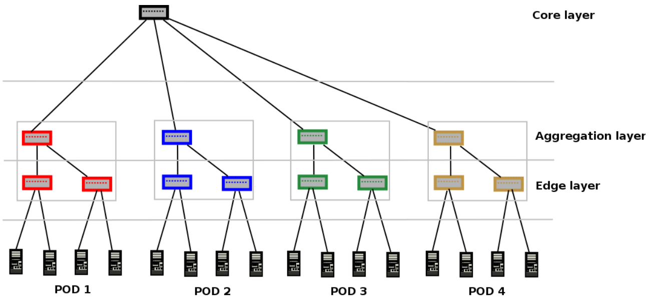


Figure 2: An Elastic-tree data center with $k = 4$ when 3 core switches and some aggregation switches are turned off.

Elastic tree, the traffic is compressed to some switches and links while the free switches and ports are turned off. The topology undergoes dynamic changes in concert with the change of traffic. That is why the topology is called Elastic tree. Elastic tree certainly saves more energy than Fat-tree. The idea of Elastic tree has also been developed further in RA-TAH [7] where besides turning off some devices, some others can also be put into sleep mode or into a reduced working rate for more efficient use of electricity. However, both Elastic tree and RA-TAH make data center vulnerable to failures due to low connectivities. For example, in Figure 2, after turning off 3 core switches and some aggregation switches, the data center becomes a tree with a single root, then there is a single path between any pair of servers. If a failure occurs on any link between the core and aggregation layers, a whole POD will be disconnected. If a failure occurs between the aggregation layer and the edge layer, an edge switch with all its associated servers will be disconnected from the data center. Although it is possible to wake up or to turn on the devices for taking over the traffic from the failed switches/links, the recovery delay remains important due to the slow starting/waking up process and long traffic re-direction process.

In this paper, we propose to add the protection capability to the Elastic tree so that even if some devices are turned off, the data center remains survivable upon any single failure. The proposed architecture is called Protected Elastic tree. We focus on protecting the communication between servers within the data center only. The main contributions of the paper are: i) proposition for the use of a path protection model in Elastic tree, ii) calculation of the minimum energy consumption topology for the Protected Elastic tree, and iii) analysis of the impact of different factors on the energy saving capability of the Protected Elastic tree topology.

A preliminary result of this research was presented in [8]. In that paper, due to limitations in the implementation, we were only able to perform the experiments on a single data center size with low traffic loads. In this paper, with the new implementation, we perform more extensive experiments and study further the impacts of network load and data center sizes on the energy saving level of the proposed Protected Elastic tree topology.

The remainder of this paper is organised as follows. The next section presents briefly the idea of conventional Elastic tree. Section 3 presents the protection model that we propose to use to reinforce Elastic tree data centers. Section 4 explains how to find the topology of the Protected Elastic tree. Section 5 shows how much energy can be saved with Protected Elastic tree and analyse the impact of traffic load and data center size on energy saving. Finally, Section 6 concludes the paper.

2 Elastic tree

Fat-tree term usually refers to the hierarchical topology where links closer to the root have greater capacities, although the Fat-tree concept here should be understood differently. It is a type of multistage circuit switching network. A k-ary Fat-tree contains k PODs. Each POD contains two layers of $k/2$ aggregation switches and $k/2$ edge switches. Each edge switch links to $k/2$ servers (or hosts) and $k/2$ aggregation switches of the upper layer. Each core switch links to k PODs by k aggregation switches. All switches are identical with k ports. Figure 1 shows the connection path between two servers through 3 layers of switches in dash line in a 4-ary Fat-tree data center. Fat-tree has a robust structure but it still consumes a significant amount of energy.

Elastic tree as proposed in [6] is a system for dynamically adapting the energy consumption of a Fat-tree data center network. Elastic tree is controlled by three logical modules Optimiser, Routing, and Power control as shown in Figure 3. The role of the Optimiser module is to find the minimum power network subset, which satisfies current (or statistical) traffic conditions. Its input consists of the topology, a traffic matrix, and the power model of each switch. The Optimiser outputs a set of active components to both the Power control and Routing modules. The Power control then turns on or off ports, linecards, or entire switches according to this output, while the Routing module chooses routes for all flows, then pushes routes to the switches in the network.

Elastic tree refers to the switch power consumption model proposed in [9]. Power consumption of a switch consists of a fixed component (chassis power and power

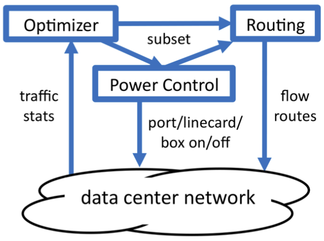


Figure 3: Control components of Elastic tree system (figure from reference [6]).

consumed by each linecard) and a variable component that depends on the number of active ports and the capacity and utilisation of each port.

$$P_{switch} = P_{chassis} + n_{linecard} \times P_{linecard} + \sum_i np_{r_i} \times P_{r_i} \times F_u \quad (1)$$

Where P_{switch}, $P_{chassis}$, $P_{linecard}$ and P_{r_i} are respectively, the power consumed by a switch, a chassis, a linecard without active ports and an active port running at rate r_i. np_{r_i} is the number of active ports running at rate r_i. F_u is an utilisation scaling factor for each port. For simplicity, this factor is considered identical for all switch ports in Elastic tree.

It is clear that an active switch without traffic processing still consumes energy. Consequently, a minimum power network subset is sought for carrying the required network load for a data center. This network contains only switches and links that must be active in each layer in order to satisfy the network load. With the help of Software Defined Network technologies such as OpenFlow [10], unnecessary switches, line cards or ports are turned off to conserve energy. The network traffic is then routed so that it flows only over the active switches. Network traffic may change and the minimum power subset of active switches must be recomputed.

Some algorithms have been proposed to identify the minimum power network subset for a data center given a required network load. The algorithms include a multicommodity flow formulation which is a mixed integer linear program, a greedy bin-packing algorithm and a Topology-aware Heuristic (TAH) [6]. Amongst these algorithms, TAH is the fastest with the least computational effort. TAH does not identify exactly which switches are a part of the minimum power network subset but instead identifies the number of active switches in each layer. From these numbers the total power consumed by the Elastic tree can be estimated.

The idea of TAH is as follows; based on the statistic of traffic between layers, TAH identifies the number of links required between an edge switch and the aggregation layer to support the up and down traffic from and to that edge switch. Assuming that the traffic is perfectly divisible, this number of links is equal to the traffic bandwidth divided by the link rate. The number of active switches in the aggregation layer is then equal to the number of links required to support traffic of the most active source from the upper layer or lower layer. Similar observation holds between the the aggregation layer and the core. Detailed computation can be found in [6]. In this paper, TAH is used to estimate the energy consumption by an Elastic tree.

In [6], the authors have also discussed adding a redundancy level to the Elastic tree by adding k parallel Minimum Spanning Trees (MSTs) that overlap at the edge switches. However, the study does not investigate further on how many MSTs would be sufficient.

3 Protection model for Elastic tree Data center

As seen in Figure 2, an Elastic tree is sensitive to failures. Its connections need to be protected. In this study we focus solely on single failure scenario. The single failure scenario assumes that there is, at most, one failure in the data center and this failure is repaired before another one may occur. The single failure scenario is a typical assumption in the research and also in practice since the frequency of failure is low [11] making the possibility of having multiple failures negligible.

There are several well-known topological protection schemes: path protection, link protection, and ring protection [12]. In the path protection model, the connection to be protected is called working path. Another path, called backup path, that shares the same end nodes with the working path is used to replace the working path when the latter fails due to link or node failures. When no failure is present the backup path is idle. Since the backup path should not be affected by any failure on the working path, the two paths should be disjoint. In link protection, each link in the working path is protected separately by a backup segment going from one end of the link to the other end. When a link fails, only one backup segment is used to replace the link. Intuitively, the link protection tends to require more backup resource than path protection since many backup segments are involved for protecting a single path. Ring protection is used in networks with ring topology. In that network, a connection follows a part of the ring and is protected by a backup connection following the remaining part of the ring in inverse direction. Due to the tree form topology of Fat-tree data centers, we focus on the path protection scheme.

A server-to-server connection in data center may travel through a suite of devices in the following order: source server - edge switch - aggregation switch - core switch - aggregation switch - edge switch - destination server. In Protected Elastic tree, we propose to use the path protection model to protect the part between the edge switches. The part between the source server - edge switch and the part between the edge switch - destination server will be left unprotected.

According to path protection model, the part between two edge switches will be protected by a disjoint path between the same two edge switches. From this principle, we apply three different protection configurations for Near, Middle and Far traffic. The notions of Near, Middle and Far traffic are defined as in [6] and [7].

- Near traffic refers to a flow between source and destination servers linked to the same edge switch. For this kind of flow, no protection is offered.
- Middle traffic refers to a flow between source and destination servers linked to different edge switches of the same POD. This kind of flow passes through an intermediate aggregate switch then returns to the edge layer within the POD. The backup flow simply uses another aggregate switch in order to guarantee that the working flow is disjointed. See Figure 4 for an example.
- Far traffic refers to a flow between source and destination servers linked to different PODs. A flow in this traffic passes through an aggregation switch of the source POD, to a core switch, then to an aggregation switch of the destination POD, and finally to the destination edge switch. The backup flow needs to use different aggregation and core switches than the working flow in order to guarantee the disjointedness. Readers are referred to Figure 5 for an example of a working flow and its backup flow of the far traffic.

Looking at the topological aspect only (without link capacity consideration), the protection configuration for Middle traffic is always possible since all aggregation switches and edge switches of the same POD link to each other. Protection configuration for Far traffic is also always possible. Since each core switch links with all PODs then with $k \geq 4$ there are always at least four core switches linking to both the source and destination PODs. Amongst these four core switches there exists at least two core switches linking to two different pairs of aggregation switches; one pair in the source POD, and the other in the destination POD. Therefore, there always exists two disjoint paths between two edge switches through these two different pairs of aggregation switches. If one path is serving as the working path then the other can be utilised as the backup path.

Now consider the bandwidth capacity aspect. In non-protection mode, a Fat-tree data center with unified link capacity is fully loaded when all servers saturate their links to edge layer. In protection mode links between edge, aggregation and core layers have to carry not only the working flows but also the backup flows, therefore the total consumed bandwidth is doubled in comparison with non-protection case. Consequently, under high traffic load, it is still possible that the data center will not have enough capacity to allocate backup flows for all working flows.

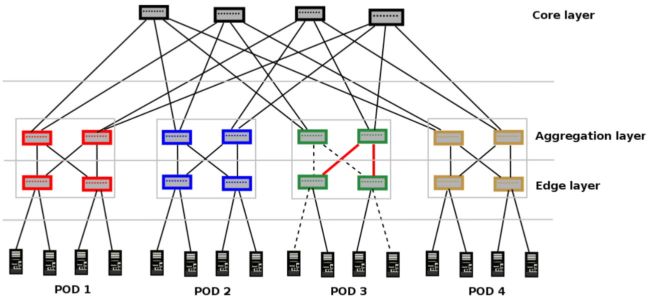


Figure 4: Example of a backup configuration for Middle traffic. The working flow is in dash line and its backup flow is in red line.

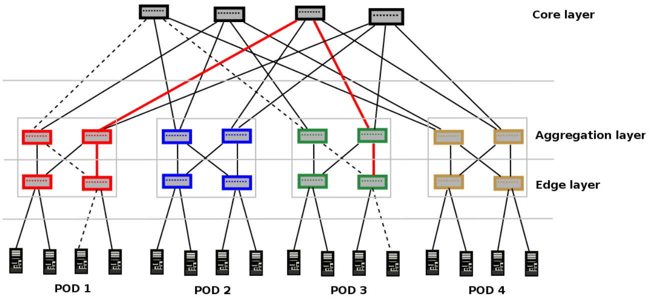


Figure 5: Example of a backup configuration for Far traffic. The working flow is in dash line and its backup flow is in red line.

4 Minimum active topology with path protection

Similar to the conventional Elastic tree, the Optimiser in Protected Elastic tree must find the minimum active topology, however in this instance the path protection mechanism is integrated. The minimum active topology with path protection is a minimal network subset that can accommodate not only all working flows for a given traffic matrix but also a backup flow for each working flow. The traffic matrix considered here contains the demanded flows between pairs of edge switches. Since we offer the protection between edge switches, we are interested in flows between edge switches only. A flow of traffic between a pair of edge switches is in fact an aggregation of multiple flows between servers. Here we assume that a rough traffic matrix between edge switches is known in advance. This assumption is quite practical since this rough traffic matrix can be estimated easily by observing the traffic statistics over time.

In order to find such a minimal network subset, we will try to accommodate working and backup paths for all flows in the traffic matrix. During this accommodation process, switches and ports are activated gradually. We assume that all switches and links in the data center are identical. Switches have k ports and link rate is r. The switches in data center are linked in k-ary Fat-tree topology. We also assume that flows are perfectly divisible. Working flows between edge switches are allocated one by one, followed by the backup flows. The flows are aggregated maximally to active links, switches and ports until those active devices are fully used before activating new devices. The following steps describe how to build the minimum active topology with path protection.

- We start with an initial active topology as the set of switches forming a Minimum Spanning Tree (MST) with a core switch as root and all servers are leaves.
- For allocating the working flow from a source edge switch e_s to a destination edge switch e_d, we browse all active paths between e_s and e_d to find the one with largest residual bandwidth. In the case that all paths are fully used, we browse from all remaining possible paths between e_s and e_d to find the one that requires least additional power consumption for activating. The power consumption of an newly activated path includes the power for activating new switches, ports and line cards as estimated by (1).
- Once all working flows are routed their backup paths will be accommodated. In order to find a backup flow for a working flow between e_s and e_d, we first exclude the aggregate switches of its working flow from the list of usable switches for the backup path. This exclusion ensures that the backup path is disjoint from the working path. Then, the backup flow is sought in similar manner as when we allocate a working flow.

If the algorithm fails to find backup paths for some of the working flows, the Optimiser ends with a false status. Once the algorithm terminates successfully each connection request in the traffic matrix will have a backup path which is disjoint with the working path. Therefore, when a single failure occurs in the network the working path or the backup path will not be affected, thus one of them is available for carrying the traffic. Consequently, the data center is 100% available to carry the traffic in the traffic matrix under any single failure.

Although in the above procedure we have to browse all the possible paths between two edge switches, the special structure of Fat-tree limits the number of paths available to browse. From an edge switch there are $k/2$ choices of aggregation switches to go up. For each aggregation switch, there are again $k/2$ choices of core switches. From a core switch there is only a single choice of route to an edge switch. Therefore, there are at most $k^2/4$ possible paths between two edges switches in Far traffic. Similarly, there are at most $k/2$ possible paths between two edge switches in Middle traffic.

Once the Optimiser has calculated the minimum active topology and the working and backup paths for flows in the traffic matrix, it gives the Routing module this information, e.g., the list of active switches and the list of working and backup paths for flows between edge switches. Later on, when the Routing module receives a connection request between two servers, it can easily identify the two edge switches associated with the two servers thanks to child-parent relationship of these devices. The Routing module will route the connection request over the working path registered for the flow between the two switches while the corresponding backup path is used for protection.

5 Evaluation of energy consumption of Protected Elastic tree data centers

The advantage of path protection scheme integrated in Protected Elastic tree data center is obvious as the data center is 100% survivable upon any single failure. However, with the presence of a backup path in parallel with a working path for each flow between edge switches, the data center consumes more energy than the conventional Elastic tree data center. In order to evaluate how much of energy the protection scheme imposes into the data center, the energy consumption of the Protected Elastic tree is compared against the fully active Fat-tree data center and against the conventional Elastic tree data center without protection.

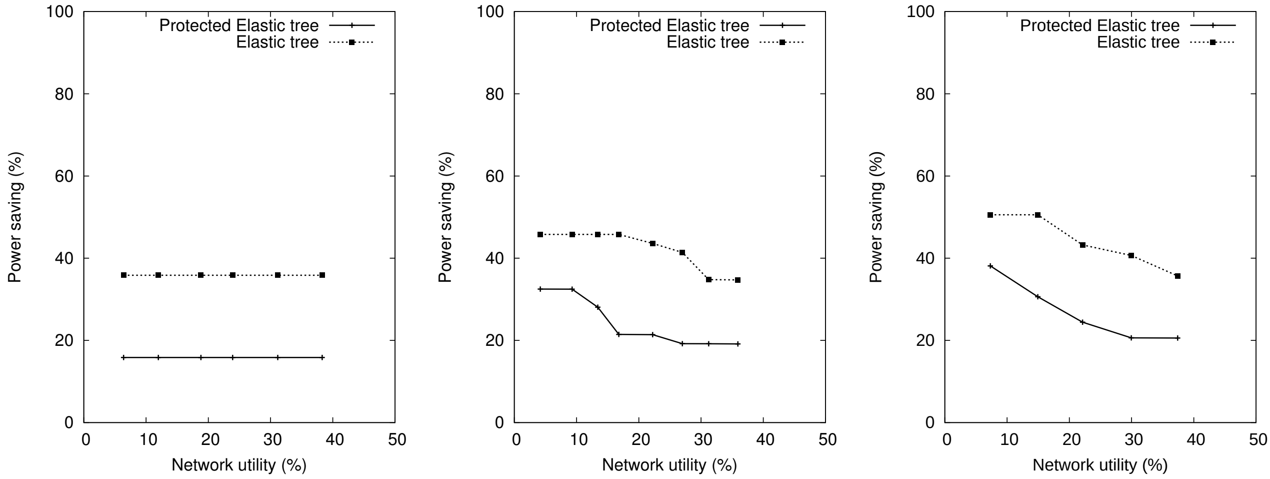
The best topology for the Protected Elastic tree is identified by “Minimal active topology with path protection” algorithm proposed in Section 4. In the previous work in [8], the algorithm was implemented in a data center emulation platform: Ecodane [13] which did not allow tests with a high traffic load due to the large computation effort that they involve. In this paper, the algorithm has been implemented in Scilab [14], an open source numerical computational package. With the new implementation, we can perform more extensive experiments. The minimal topology for the conventional Elastic tree is calculated using the TAH algorithm which has also been implemented in Scilab.

Let λ_{ij} be the total requested bandwidth from server i to server j. Let $\#server$ be the number of servers in the data center. The total bandwidth capacity provided by links between servers and edge switches is $\#servers \times r$. Since a flow between server i and server j uses two links between server layer and edge layer, the maximum total acceptable load between servers in the data center is $0.5\#servers \times r$. From this observation, we define the network utilisation index as the ratio between the total requested bandwidth between servers in data center and the maximum total acceptable load of the data center. It is calculated by:

$$u = \frac{\sum_{ij} \lambda_{ij}}{0.5\#servers \times r} \times 100\% \qquad (2)$$

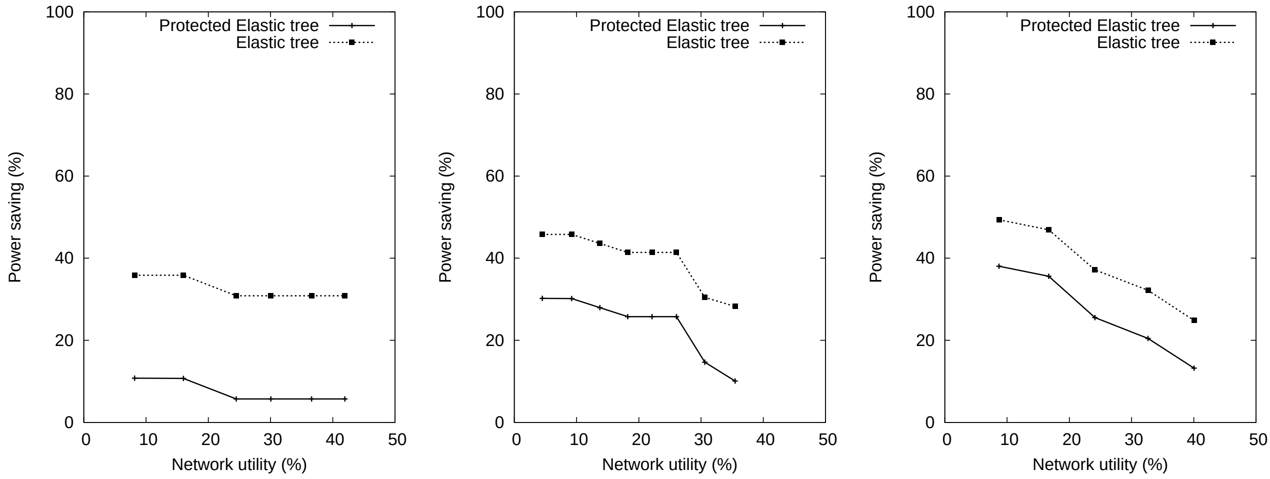
In case on non-protection, when $u = 100\%$ the data center is maximally loaded. In case of protection, for each pair of servers, a flow for working path needs another flow for backup path. Therefore, the maximum network utilisation for protected data center is $u = 50\%$.

In all simulations, switches are wired in Fat-tree topology. Data center sizes vary with $k = 4, 6, 8$. All links



(a) k=4. (b) k=6. (c) k=8.

Figure 6: Power saving level with Middle traffic



(a) k=4. (b) k=6. (c) k=8.

Figure 7: Power saving level with Far traffic

are bi-directional with bandwidth capacity of $r = 10$ Gbps. Traffic requests are generated for 3 traffic models: Far traffic, Middle traffic and Mixed traffic. Mixed traffic does not include Near traffic since we are not interested in protecting Near traffic in this research. In order to reserve enough spare capacity for protection, traffic is generated with increasing network utilisation until it approaches $u = 50\%$ but without saturating the data center. Each server-to-server connection requests a bandwidth uniformly distributed in range [0-1] Gbps.

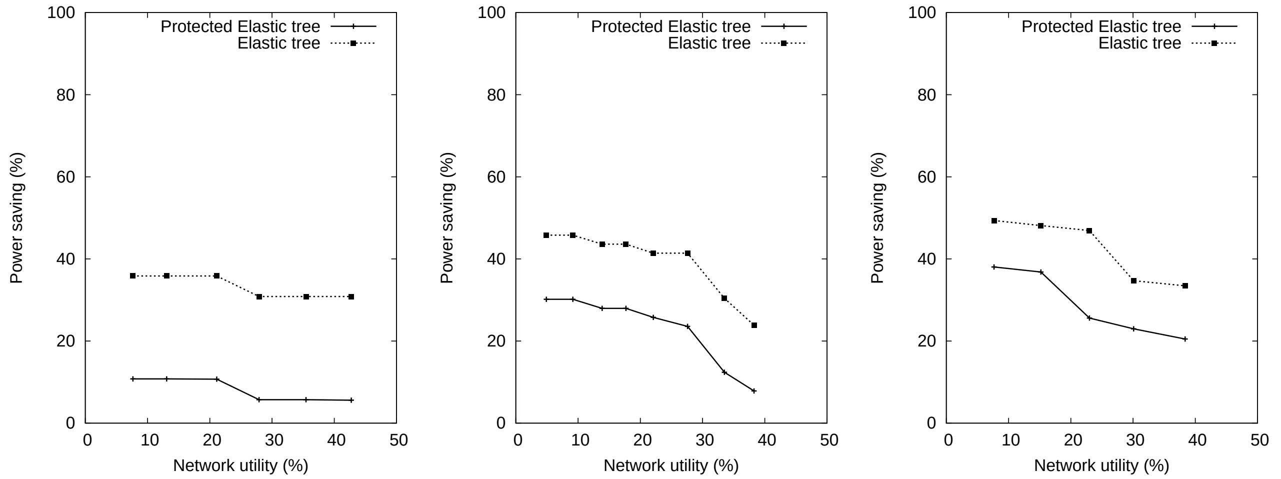
The total energy usage by Fat-tree and conventional Elastic tree are evaluated based to their number of active devices. According to (1), the power consumption of a switch consists in a fixed part including power consumption of chassis and line cards, and a dynamic part including power consumption of ports. Let us denote the fixed part of the power consumption by P. Let us also consider that the power consumption of a port is constant regardless of its working rate and is denoted it by p.

It is easy to prove that a k-ary Fat-tree data center has a total of $5k^2/4$ switches, $5k^3/4$ ports and $3k^3/4$ links. Since they are all active, the power consuming by a Fat-tree, regardless of traffic matrix, is

$$P_{Fat} = \frac{5}{4}k^2P + \frac{5}{4}k^3p \tag{3}$$

In Elastic tree, all edge switches must be active in order to be ready to receive data from the servers, therefore, the number of active edge switches is always $k^2/2$. Assume that a conventional Elastic tree topology, identified by TAH for a given traffic matrix, uses x aggregation switches ($x \leq k^2/2$), y core switches ($y \leq k^2/4$) and has n_p active ports, then the total energy consumed by the Elastic tree is

$$P_{Elastic} = \frac{k^2}{2}P + xP + yP + n_pp \tag{4}$$



(a) k=4. (b) k=6 (c) k=8

Figure 8: Power saving level with Mixed traffic

Th Energy saving level of the Elastic tree over the Fat-tree is defined as:

$$\left(1 - \frac{P_{Elastic}}{P_{Fat}}\right) \times 100\% \quad (5)$$

We denote the energy consumed by Protected Elastic tree by $P_{Protected.Elastic}$. The energy consumed by Protected Elastic tree is also evaluated by using (4) where x, y and n_p are the number of active aggregation switches, the number of active core switches and the number of active ports in Protected Elastic tree. Those numbers are obtained from the calculations in the algorithm "Minimal active topology with path protection". The energy saving level of the Protected Elastic tree over Fat-tree is defined as:

$$\left(1 - \frac{P_{Protected.Elastic}}{P_{Fat}}\right) \times 100\% \quad (6)$$

In all tests, the power consumed by a switch chassis including line cards is set $P = 146$ watts, the power consumed by a port is set $p = 0.9$ watts. These numbers have been chosen after analysing the study of switch power consumption in [9].

Figure 6, 7 and 8 show the power saving level of the Protected Elastic tree and Elastic tree with $k = 4, 6, 8$ for Middle, Far and Mixed traffic. The detailed results are shown in Table 1, 2 and 3 where the columns show the following information:

- Network utility (in percentage)
- Power consumed by Fat-tree topology P_{Fat}
- Number of active switches, number of active ports, total power consumption and energy saving level (in percentage) of the Protected Elastic tree
- Number of active switches, number of active ports, total power consumption and energy saving level of the conventional Elastic tree

It is clear that the Protected Elastic tree consumes more energy than the conventional Elastic tree. The observed gap between energy saving level of Protected Elastic tree and Elastic tree varies roughly between 10% and 25%.

In the following sessions the impact of network utility, traffic model and data center size on the energy saving performance of the proposed Protected Elastic tree topology are analysed.

5.1 Impact of network utility on energy saving

We can observe in Figures 6, 7, 8 that, except in Figure 6a, the power saving levels of both conventional Elastic tree and Protected Elastic tree decrease when the network utility increases. The obvious reason is that when the network utility increases, both the conventional Elastic tree and Protected Elastic tree have to use more switches in order to accommodate the increase in traffic load.

In the case of the data center with size $k = 4$ for Middle traffic in Figure 6a, the power saving levels of both Elastic tree and Protected Elastic tree are constant regardless of the network utility. This special case can be explained as follows; with the Middle traffic, all flows (including working and backup) travel only inside a POD and do not involve core switches. In a non-protected Elastic tree data center, all edge switches are active in order to be ready to receive traffic from the servers. At the aggregation layer, one aggregation switch per POD must be used to carry all traffic from the POD's edge switches. With the network utility at 50% or under, one aggregation switch with two aggregation-edge links is sufficient to carry the traffic from/to the four servers of the POD. Therefore the total number of active switches is always: 8 (edge) + 4 (aggregation) =12 switches. Now let us consider the Protected Elastic tree data center, again 8 edge switches must be active. Both aggregation switches inside a POD must be active in

		Protected Elastic tree				Elastic tree			
Utility	P_{Fat}	Nb. swt.	Nb. ports	Power	Saving	Nb. swt.	Nb. ports	Power	Saving
Far traffic									
8.16	2992	18	46	2669.4	10.78	13	24	1919.6	35.84
16	2992	18	48	2671.2	10.72	13	24	1919.6	35.84
24.49	2992	19	52	2820.8	5.72	14	28	2069.2	30.84
30.05	2992	19	52	2820.8	5.72	14	28	2069.2	30.84
36.59	2992	19	52	2820.8	5.72	14	28	2069.2	30.84
41.95	2992	19	52	2820.8	5.72	14	28	2069.2	30.84
Middle traffic									
6.4	2992	17	40	2518	15.84	12	16	1766.4	40.96
11.98	2992	17	40	2518	15.84	12	16	1766.4	40.96
18.81	2992	17	40	2518	15.84	12	16	1766.4	40.96
23.91	2992	17	40	2518	15.84	12	16	1766.4	40.96
31.16	2992	17	40	2518	15.84	12	16	1766.4	40.96
38.29	2992	17	40	2518	15.84	12	16	1766.4	40.96
Mixed traffic									
7.65	2992	18	46	2669.4	10.78	13	24	1919.6	35.84
13.08	2992	18	46	2669.4	10.78	13	24	1919.6	35.84
21.11	2992	18	48	2671.2	10.72	13	24	1919.6	35.84
27.91	2992	19	52	2820.8	5.72	14	28	2069.2	30.84
35.45	2992	19	52	2820.8	5.72	14	28	2069.2	30.84
42.73	2992	19	56	2824.4	5.6	14	28	2069.2	30.84

Table 1: Test results with k=4

		Protected Elastic tree				Elastic tree			
Utility	P_{Fat}	Nb. swt.	Nb. ports.	Power	Saving	Nb. swt.	Nb. ports	Power	Saving
Far traffic									
9.22	6813	32	96	4758.4	30.16	25	48	3693.2	45.79
13.75	6813	33	100	4908	27.96	26	52	3842.8	43.6
18.21	6813	34	104	5057.6	25.77	27	56	3992.4	41.4
22.13	6813	34	104	5057.6	25.77	27	56	3992.4	41.4
26.04	6813	34	104	5057.6	25.77	27	56	3992.4	41.4
30.6	6813	39	132	5812.8	14.68	32	74	4738.6	30.45
35.47	6813	41	156	6126.4	10.08	33	78	4888.2	28.25
Middle traffic									
9.28	6813	31	84	4601.6	32.46	24	36	3536.4	48.09
13.41	6813	33	92	4900.8	28.07	24	36	3536.4	48.09
16.79	6813	36	106	5351.4	21.45	24	36	3536.4	48.09
22.22	6813	36	110	5355	21.4	25	40	3686	45.9
26.99	6813	37	114	5504.6	19.2	26	44	3835.6	43.7
31.23	6813	37	116	5506.4	19.18	29	58	4286.2	37.09
35.92	6813	37	118	5508.2	19.15	29	62	4289.8	37.04
Mixed traffic									
9.16	6813	32	96	4758.4	30.16	25	48	3693.2	45.79
13.87	6813	33	100	4908	27.96	26	52	3842.8	43.6
17.7	6813	33	100	4908	27.96	26	52	3842.8	43.6
22.09	6813	34	104	5057.6	25.77	27	56	3992.4	41.4
27.59	6813	35	108	5207.2	23.57	27	56	3992.4	41.4
33.51	6813	40	142	5967.8	12.41	32	76	4740.4	30.42
38.26	6813	42	162	6277.8	7.86	35	86	5187.4	23.86

Table 2: Test results with k=6

		Protected Elastic tree				Elastic tree			
Utility	P_{Fat}	Nb. swt.	Nb. ports	Power	Saving	Nb. swt.	Nb. ports	Power	Saving
Far traffic									
8.7	12256	51	164	7593.6	38.04	42	84	6207.6	49.35
16.7	12256	53	172	7892.8	35.6	44	92	6506.8	46.91
24.12	12256	61	246	9127.4	25.53	52	122	7701.8	37.16
32.64	12256	65	288	9749.2	20.45	56	152	8312.8	32.17
40.04	12256	71	300	10636	13.22	62	176	9210.4	24.85
Middle traffic									
7.32	12256	51	154	7584.6	38.12	40	64	5897.6	51.88
14.94	12256	57	202	8503.8	30.62	40	64	5897.6	51.88
22.13	12256	62	232	9260.8	24.44	46	92	6798.8	44.53
29.95	12256	65	266	9729.4	20.62	48	116	7112.4	41.97
37.41	12256	65	272	9734.8	20.57	52	144	7721.6	37
Mixed traffic									
7.67	12256	51	164	7593.6	38.04	42	84	6207.6	49.35
15.19	12256	52	168	7743.2	36.82	43	88	6357.2	48.13
22.99	12256	61	238	9120.2	25.59	44	92	6506.8	46.91
30.09	12256	63	268	9439.2	22.98	54	134	8004.6	34.69
38.36	12256	65	282	9743.8	20.5	55	142	8157.8	33.44

Table 3: Test results with k=8

order to provide disjoint working and backup paths for each connection in the POD. Therefore, 8 aggregation switches will be involved. Since Protected Elastic tree topology is built from a MST with a core switch as root then one core switch will also be involved. Consequently, the total number of active switches is always 8 (edge) + 8 (aggregation) + 1 (core)=17 switches regardless of network load. This phenomena of constant power saving does not happen with Middle traffic for bigger data center sizes.

5.2 Impact of traffic model on energy saving

Since Middle traffic does not go through core switches as it happens in Mixed and Far traffic, those core switches can be turned off. This characteristic leads to several advantages in Middle traffic:

- The Protected Elastic tree with Middle traffic saves, in general, slightly more energy than Far and Mixed traffic for the same data center and with the same network utility (see Tables 1, 2, 3).

- Protected Elastic tree data centers under Middle traffic always save a certain amount of energy regardless of the network utility. This amount is equivalent to the energy consumed by core switches. That is the reason why the energy saving level lines of Middle traffic become stable as the network utility increases, meanwhile, those for Far and Mixed traffic continue to decrease. We can observe that even for networks with high utility (close to 40%) the saving ratios are still considerable, i.e., 15.84%, 19.15% and 20.57% when $k = 4, 6$ and 8 respectively.

We can also notice that the power saving level lines for Far and Mixed traffic have similar shapes. This similarity demonstrates that the Far traffic has a stronger impact on power saving than Middle traffic. The reason is that Far traffic involves more links and switches due to their long connection paths and Middle traffic can profit from those links and switches for its connections without activating additional elements.

5.3 Impact of data center size on energy saving

The performance of Protected Elastic tree over different data center sizes can also be observed in Figures 6, 7, 8. We have two remarks:

- Power saving lines of both Elastic and Protected Elastic tree shift up slightly from $k = 4$ to $k = 6$, and the same observation can be made from $k = 6$ to $k = 8$, for all traffic patterns. This means that both the Elastic tree and the Protected Elastic tree become more energy efficient as the size of the data center increases.

- The power saving lines of the Protected Elastic tree for Far and Mixed traffic are closer to those of the Elastic tree when size of the data center increases. The conclusion to be drawn from this is that the extra energy consumption for protection becomes less important in large data centers. The main reason of this phenomena is that larger data centers have a higher connectivity and, therefore, more possibilities to route backup flows over switches and ports that are already active to carry working flows.

These two observations show that the Protected Elastic tree topology is an option worthy of consideration for large size data centers.

6 Conclusions

The Fat-tree topology has been proposed as a topology for data centers characterised by a low over-subscription and a high availability. Elastic tree has been proposed in order to reduce the energy consumption of Fat-tree data centers by deactivating unnecessary switches and links. Although Elastic tree is a highly energy efficient topology, it severely damages the availability of the Fat-tree since the network connectivity is reduced remarkably. In this paper, we proposed the possibility of adding protection capabilities to the conventional Elastic tree by allocating backup paths for each aggregated flow between edge switches. Many backup paths can profit from existing active switches and ports while some others require the activation of additional elements in the network. This results in a Protected Elastic tree topology where 100% of connections are survivable for any single failure. The simulation results show that the path protection scheme generates a decrease in the energy saving ratio of 10%-25% for the proposed Protected Elastic tree in comparison with the conventional Elastic tree. However, the Protected Elastic tree data center saves a notable amount of energy if it is compared with the Fat-tree, mostly for the Middle traffic model. Moreover, the proposed solution becomes more energy efficient as the data center size increases.

Acknowledgement

The authors would like to thank Assoc. Professor Nguyen Huu Thanh for his encouragement to perform this research.

References

[1] FEPM, "FEMP Data Center Program Overview," Tech. Rep. DOE/EE-1191, Federal Energy Management Program, March 2015.

[2] Anthesis, "Data Center Efficiency Assessment," Tech. Rep. IP:14-08-A, US Natural Resources Defense Council, August 2014.

[3] K. K. Nguyen, M. Cheriet, M. Lemay, V. Reijs, A. Mackarel, and A. Pastrama, "Environmental-aware virtual data center network," *Computer Networks*, vol. 56, no. 10, pp. 2538 – 2550, 2012. Green communication networks.

[4] C. Leiserson, "Fat-trees: Universal networks for hardware-efficient supercomputing," *Computers, IEEE Transactions on*, vol. C-34, pp. 892–901, Oct 1985.

[5] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, SIGCOMM '08, (New York, NY, USA), pp. 63–74, ACM, 2008.

[6] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "Elastictree: Saving energy in data center networks," in *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, (Berkeley, CA, USA), pp. 17–17, USENIX Association, 2010.

[7] T. N. Huu, N. P. Ngoc, H. T. Thu, T. T. Ngoc, D. N. Minh, V. G. Nguyen, H. N. Tai, T. N. Quynh, D. Hock, and C. Schwartz, "Modeling and experimenting combined smart sleep and power scaling algorithms in energy-aware data center networks," *Simulation Modelling Practice and Theory*, vol. 39, pp. 20 – 40, 2013. S.I.Energy efficiency in grids and clouds.

[8] D. L. Truong and T. C. Nguyen, "Protected elastic-tree topology for data center," in *Proceedings of the Sixth International Symposium on Information and Communication Technology, Hue City, Vietnam, December 3-4, 2015*, pp. 129–134, 2015.

[9] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A power benchmarking framework for network devices," in *Proceedings of the 8th International IFIP-TC 6 Networking Conference*, NETWORKING '09, (Berlin, Heidelberg), pp. 795–808, Springer-Verlag, 2009.

[10] "Openflow." `https://www.opennetworking.org/sdn-resources/openflow/`.

[11] M. To and P. Neusy, "Unavailability analysis of long-haul networks," *IEEE Journal on Selected Areas in Communications*, vol. 12, pp. 100–109, Jan. 1994.

[12] J. Zhang and B. Mukheriee, "A review of fault management in WDM mesh networks: basic concepts and research challenges," *IEEE Network: Magazine of Global Internetworking*, vol. 18, pp. 41–48, Mar. 2004.

[13] H. T. Nguyen, D. C. Bui, D. T. To, N. N. Pham, Q. T. Ngo, T. H. Truong, and M. N. Tran, "Ecodane: A customizable hybrid testbed for green data center networks," in *Advanced Technologies for Communications (ATC), 2013 International Conference on*, pp. 312–317, Oct 2013.

[14] "Scilab." `http://www.scilab.org`.

MISNA: Modeling and Identification of the Situations of Needs for Assistance in ILE

Nadia Beggari and Tahar Bouhadada
Laboratory of Computer Research (LRI)
Badji Mokhtar – Annaba University P.O. Box 12, 23000 Annaba, Algeria
E-mail: nadiabeggari@hotmail.fr, bouhadada.tahar@univ-annaba.org

Keywords: assistance system, traces, indicators, proactive assistance, difficulties situations, interactive learning environment

Received: July 31, 2015

For the majority of assistance systems in online learning, the design and the implementation of an intervention strategy are among the most prominent obstacles, and introducing an adequate assistance in the good time is not an easy task. In this research paper, we proposed an active assistance approach that allows calculating during the learning process, revealing indicators of the learners' difficulties. This approach is based on IMAC model that defines and models the assistance situation using a trace indicator with its calculation rule, an intervention modality, an Aspect, and a Category of assistance situation. An operational prototype was developed to evaluate and to implement this approach containing a learning environment, an assistance editor, a collector of traces and our detector of the situations of assistance needs. In order to evaluate our proposal, we conducted field experiments in the domain of professional PowerPoint presentations. The impact of the information provided by revealing indicators, both on assistance system intervention and learner activity, was analyzed. The results suggest that the revealing indicators improve the effects of the proactive interventions and have a positive impact on learners' reactions

Povzetek: Predstavljena je nova metoda aktivne pomoči, testirana na pomoči snovalcem prezentacije v PowerPointu, ki se uči sproti.

1 Introduction

In the field of ILE (Interactive Learning Environments), the importance of help notion and help systems is commonly admitted and it makes a subject of the abundant literature revolving around five major research areas [25]. First, the problematic concerns the problem identification encountered by the learners [29]. A second line of research, the help requests that have formulated by the learners [31]. Third, the design and the characterization of help proposed to the learner [28]. The fourth axis covers research in the construction of learner's profiles and the integration of this profile in the formulation process of the help [37]. Finally, the evaluation as a fifth axis, which is additional research domain, allows validating the proposed help and evolving its place in the environment.

Nevertheless, despite all these efforts, the help system remains generally ignored or rarely consulted by users. Either because of irrelevant intervention following an interruption during the task, or fear to lost time [6]. For these reasons, it seemed very interesting to think about the design of an intervention strategy. This strategy take account the specific needs of each learner like: his learning strategy, his preferences, his objectives and his experiences, in order to provide an adequate assistance with a proactive or a reactive intervention.

Our work focuses on the characterization of this intervention strategy and the determination of its components, with the stakes related to the identification of conditions and intervention criteria, on the one hand, and on the other hand, the stakes related to the proposal of a personalized assistance approach for calculating, during the learning process, the revealing indicators of the difficulties encountered by learners.

Our proposal is based on the collect of an interaction traces according to a predefined formalism which describes the concepts and relationships manipulated by the learners in ILE. These traces can be considered as a source of knowledge that the system can use in order to provide individualized assistance for learners. It involves analyzing the traces left by the learners during their activities for calculating the revealing indicators of the difficulties situations that require the proactive assistance interventions. Besides, these revelations can be used to enrich the content of the assistance system for adapting to the new situations not anticipated during the design phase.

The article is organized as follows: Section 2 presents the problematic and the research questions of our work. Section 3 presents the related works. Section 4 presents our approach and our system architecture with the theoretical foundations. It details the IMAC model we have proposed to represent the assistance situations, the classification of revealing indicators, the proposed indicators with their formulas and the treatment process of indicator calculate. We present in Section 5 our environment called Modeling and identifying the situations of Needs for assistance (MISNA). In Section 6, we discuss the relationships between the assistance situation and learners' traces through an experiment, conducted with 40 students, to validate our assumption about deducing difficult situations from learners' interactions. In Section 7, we present our results and discussion. The case study, in Section 8 and finally Section 9 concludes and future perspectives.

2 Problematic

This work aims to develop an active and personalized assistance system, capable of identifying the learners in difficult situations and providing the appropriate assistance.

How to improve the acceptability of the assistance system? How to raise its effective use by learners? and, How to favor the learners' persistence? This is the general problematic of our work. For this, we focus on the analysis of the interaction learners' traces on one hand and on the positive influence of proactive interventions on learners' motivation, on the other hand. Therefore, many research questions arise:

RQ1: *Which difficult situation nature is appropriate to web-based learning systems?*

RQ2: *How to model the assistance situations and the difficulty notion?*

RQ3: *On which criteria we take the initiative for proposing a proactive assistance?*

RQ4: *Is it possible to reveal automatically information about difficult situations from interaction traces and how can be achieved?*

3 Related works

Much effort has been made in an assistance system domain, in order to offer an adapted and personalized assistance to ILE's learners. For instance, we can mention the works offering the interfaces that "give and take advice" in interaction with the users [20]. The web browser Letizia gives advices and proposes assistance to the user, the user can accept, ignore or reject the proposal. The works aiming to help in the classification of electronic messages [13], in the use of design tools [15] or help in the navigation on complex websites [33]. Although this works use the interactions user-environment stored in memory, in order to propose to the users the action suggestions, the classification techniques for organizing messages, and take a reasoning-based rule in order to complete a design case. However, these assistants are based on pre-defined strategies that prevent them to develop for adapting to the new situations that are not anticipated by their designers, so they are based on a specific model for a specific application.

However, the generic adaptive assistants are designed for adapting to users in individual way to one or more needs, whether explicit or not. According to [21], the generic assistants are based on the help individualistic approach in the generalization process for automation. These tools are applicable to various fields of knowledge, to various host systems (assisted system) as described in [7]. This generic assistant is made up of a task model, an user and a group model, as well as a set of assistance interventions allowing to respond to the user's needs by providing appropriate assistance. Generic adaptive tools exploit the knowledge they have on users for adapting their behavior and for adapting to other users have the same case of difficulties [30].

There are also, the assistance systems that use the learning traces. These systems allow analyzing natively or externally the produced traces [3]. For example, the Pixed project [16] reuses learner's interaction traces in order to help them to find their way. This system is an application of the Musette approach (Modelling USEs and Tasks for Tracing Experience) to ILE. This approach consists of exploiting the interaction traces to aim the activity analysis needs and equally the help to the user. An External, a trace analyzer will allow finding in the database traces a similar episodes to an explain task signature. In [26], the authors provide an approach for collecting and exploiting communication traces of a forum to provide the help to the learners, tutors, teachers and researchers during and after their activities. His proposal focuses on two points: (1) the tracing communication activities and (2) the tool proposal allowing the tutors and the learners to analyze and to visualize, in real-time, the obtained traces. The Tool proposed by [5] is also a work concerning the assistance systems based on traces, this system proposed eighty indicators (the individual indicators, the group indicators and the general indicators) whose aim is to provide a direct assistance to the users (including the learners working in collaboration) to activate their metacognitive processes, allowing them to regulate their activities and offer to teachers the possibility to identify the situations and the difficulties that require the regulatory interventions.

However, most of these systems don't provide individualized and adapted assistance in real time assistance to the learner. They integrate a traditional pedagogical approach (behavioral) based on the prescriptive and the specific models for a specific application instead of adopting the recent didactic approaches (Constructivism and Social-Constructivism), which are based on the open models to provide a realistic environments rather than pre-determined learning sequences [2].

Thus, these different systems are generally based on the idea that learners (or users) are self-regulating, which means that they manage their learning independently. For this, they provide assistance only at the request of the student. Nevertheless, several studies have shown that most learners do not know that they are in a difficult situation and whether they know it; they prefer to use the training in order to acquire the necessary knowledge, or they ask their colleagues. Consequently, the help systems are usually ignored or rarely consulted [6].

In this paper, we focus on the development of active assistance system capable of providing a spontaneous assistance (that is to say, relevant and well-timed) to ILE's learners, either to their request or at the system's decision. For this, we focus on the tools based on the sharing and reuse of experience, which aims to find the right system intervention dosage according to the learners' needs.

4 Approach

To be useful and effective, an assistance system must be capable of providing to the learner the necessary assistance at the exact moment, when he really needs it. In order to determine this moment, it is necessary to identify the difficulties that the learner will encounter during his learning and as it is not possible to predict all difficulties at the time of system design [14] it is conceivable to calculate a revealing indicators of these difficulties during the learning sessions from learners' interaction traces.

However, it is necessary, above all, to define correctly, the key concepts of our approach:

- **Learning difficulty:** According to Perraudeau [30]: Learning difficulty is an ordinary time of learning that is not to punish, but to take as an activity indicator of the learner." This difficulty is individual or social source; the personal one is revealed in the complex relations between the development of thought and the knowledge to acquire. However, the social one is revealed in the relations of the learner with others through two dimensions: one macro-social (family, culture) and the other micro-social (relationships with other students, teachers and the learning context). In this Research paper, we interest in the kind of difficulties that have their origin in the mobilization of the thinking operations and the procedures implemented, like discovery or research complex tasks and to micro-social difficulties.
- **Assistance situation:** the assistance situation is a formalism of assistance specification, formulated by a couple of a problematic situation and an assistance proposal.
- **Problematic situation:** in literature, the problematic situation is a general learning strategy, when the teacher confronts the students with an important problem. In our system, the problematic situation is a learning situation proposed by the teacher with his a prevention problem.
- **Assistance proposal:** is an assistance action made by assistance designer for responding to learner's needs. The assistance actions in ILE are realized by a set of assistance means like: message, example, and change the interface of ILE.

In order to deduce the assistance situation automatically by the learning indicators in ILE, we need first to specify which assistance situation is appropriate to web-based learning systems and by which indicators we can be detected. The following sub-sections provide some answers, describing our approach, explained in Figure 1.

4.1 Difficult situation modeling

Our assistance characterization is based on a formalism which models the assistance situations in couple form problematic situation and assistance proposal (explained in Figure 2). The problematic situation appears with a model that specifies the learning moments when the learner needs assistance by a set of characteristics: a Trace indicator, a modality of intervention, an Aspect

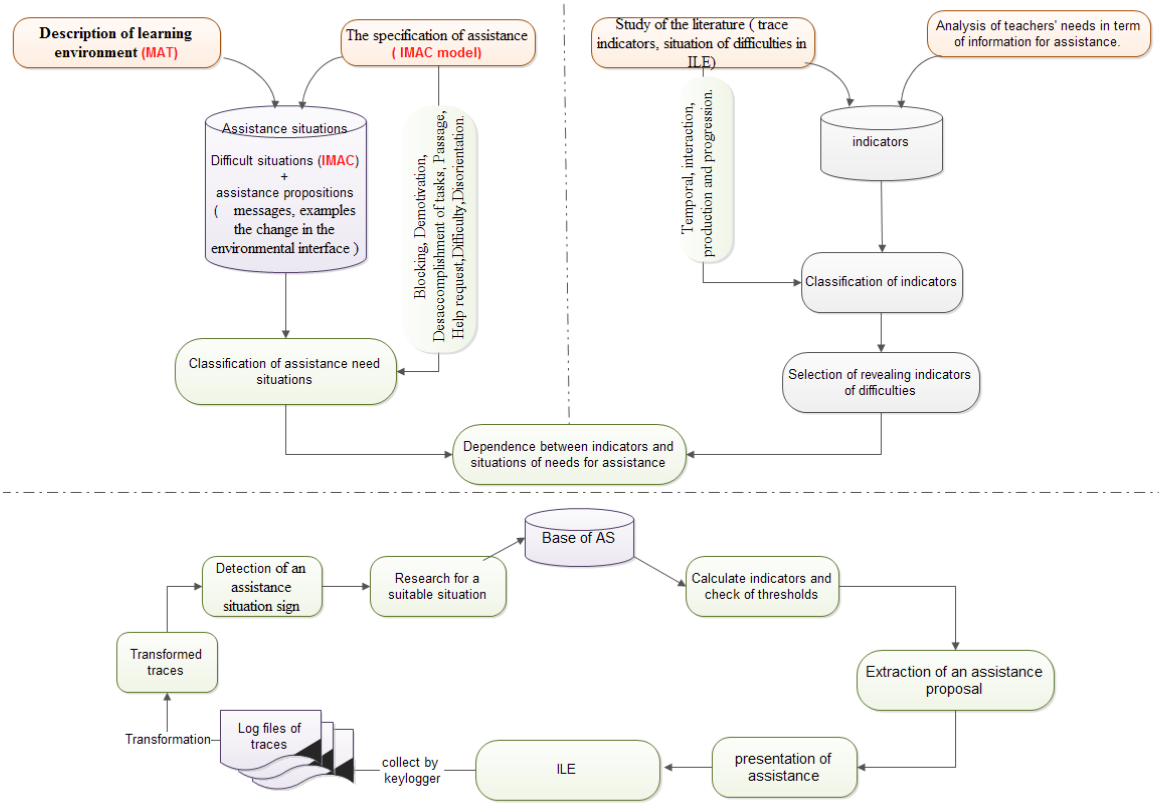

Description of learning environment (MAT)
The specification of assistance (IMAC model)
Assistance situations
Difficult situations (IMAC) + assistance propositions (messages, examples the change in the environmental interface)
Blocking, Demotivation, Desaccomplishment of tasks, Passage, Help request, Difficulty, Disorientation.
Classification of assistance need situations
Study of the literature (trace indicators, situation of difficulties in ILE)
Analysis of teachers' needs in term of information for assistance.
Temporal, interaction, production and progression.
indicators
Classification of indicators
Selection of revealing indicators of difficulties
Dependence between indicators and situations of needs for assistance
Base of AS
Detection of an assistance situation sign
Research for a suitable situation
Calculate indicators and check of thresholds
Transformed traces
Extraction of an assistance proposal
Transformation
Log files of traces
collect by keylogger
ILE
presentation of assistance

Figure 1: Architecture of the Assistance System.

and a Category. The assistance proposal is defined by assistance means associated to component of learning environment.

4.1.1 IMAC model (indicator, modality, aspect and category)

IMAC is based on four components: the trace indicator that reveals the learner difficulties, the modality for specifying the form of intervention, the aspect that determines the assistance dimension and the category that indicates the kind of situation.

a. Learning Indicators

In order to assist the learner's activity and to determine the conditions of the proactive interventions, the revealing indicators of learners' difficulties are offered. To calculate these indicators, we use the transformed traces (traces collected after processing, defined later). Each indicator is defined by a calculation rule and an acceptability domain of values allows identifying the critical situation that may be an assistance situation. The assistance system intervenes, if the indicator' value is in its domain of values. For example: the inactivity time indicator above to a threshold (defined by an expert) can be an assistance situation' indicator of blocking kind. If a learner clicks on many areas of the interface for a relatively long time, it could mean that he is looking for something, he is lost or he is disoriented.

b. Intervention Modalities

In the field of the learning environment, two style of intervention exist: proactive and reactive, *a* proactive style when the tutor or the assistant intervenes spontaneously beside the learner and a reactive style when the tutor or assistant only react to the learner's request. However, the assistant can decide to intervene proactively throughout of training or in specific times. This intervention can be systematic or opportune.

In the present work, we are particularly interested in the proactive modality to anticipate the difficulties encountered by the learners, and consequently to reduce the perturbations that could be associated there.

Thus, we have identified three types of assistance according to three intervention modalities:

- A reactive assistance that represents the learner' request to his assistant or his peers and vice versa, enabling them to advance in their task.
- A proactive assistance that represents the means and strategies provided by the upstream assistant and before any use of actors, in order to mediate this process. The proactive assistance may be systematic or *opportune*.
  - In a **systematic** proactive assistance, the assistant decides to intervene in a proactive organized manner when, for example, at the end of a stage, the assistant provides significant information for the rest of the activity.
  - In an **opportune** proactive assistance, a trigger sign (a particular committed error, an irrelevant choice, etc.) is the opportunity for the assistant intervenes, for example, in order to orient the learner towards more efficient learning strategies. In this system, the release sign is a trace indicator.

c. Aspect

The problematic situation can be characterized by an Aspect or a dimension, which determines the content which brings. Many researches are focused on this characterization, such as the authors in [34], when they are interested in the didactic dimension of the help. In [26], the authors are looked for the cognitive dimension, while the author in [37] is interested in the technological dimension.

Thus, on the basis of the help dimensions' topology proposed by author in [18], we associated to the assistance needs situation seven aspects:

- ***Cognitive Aspect:*** we connect to the cognitive aspect, every critical situations when its content focuses on the comprehension and appropriation activity. For

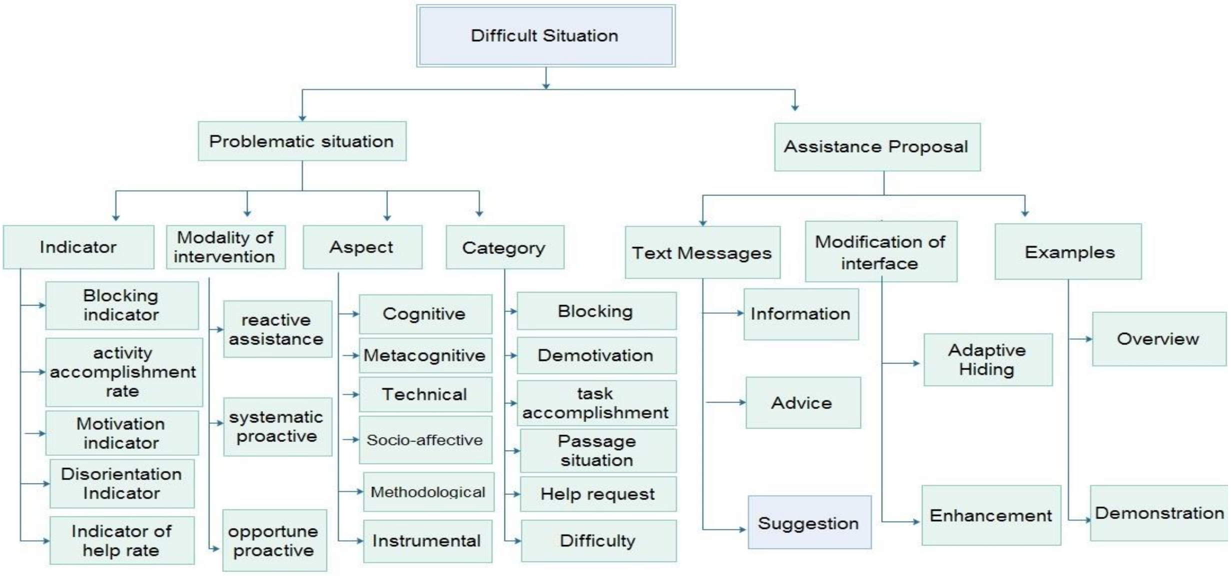


Figure 2: The IMAC model.

example, the analysis of the tackled problem, the confrontation of views and the incitement to apply concretely the discovered concepts, the blocking in theoretical learning and memorizing or reasoning difficulty.

- ***Metacognitive Aspect:*** In this aspect, we group every critical situation comprises the problems on the new knowledge construction activities, which need a metacognitive reflection.
- ***Organizational Aspect:*** for the organizational aspect, we adjust every critical situation comprise on the organization of students' work in a different category, qualified according to the authors of "organizational", "regulative" or "managerial" [32] such as, the situations of time management problems that facilitate the planning of activities, in order to respect the deadlines.
- ***Technical Aspect:*** We attach to the technical aspect every situation related to the technique such as: point out a technical problem or a malfunction on the machine, a connection problem, the problems related to the use of Training software: ask how to run or to work particular software.
- ***Socio-affective Aspect*****:** it groups the assistance situations intended to support the socio-affective assistance of social order, such as, the difficult situations to set up the positive relationships between the users (learner and teacher), the contact problems and the situations of collaboration and communication problems.
- ***Methodological Aspect:*** for this aspect, we are very interested on the main methodological problems situations that encounter the learners. For instance: the problems of the learning resources management, the research of the methods and the approaches to achieve some activities.
- ***Instrumental Aspect****:* a situation is called instrumental, when the learner searches the minimum information that him lacks to succeed the task by his own means [27]. We can determine this kind of situations via a mastery goal indicator, when the learner looks to develop his skills. Also, different authors such as, the authors in [35] show, that more the learner pursues a mastery goal, more he searches the instrumental help and so he is in an assistance situation.

d. Categories of Assistance Situations

The aim of assistance systems in general, is to respond to a particular need, specific to each learner of the learning system that is in a situation of exchange in foreign language, of blocking or interrogation facing to many choices that are available to him [25].

We tried to generalize these situations that are related to a specific domain, in order to establish this topology.

- ***Blocking situation***: We detect a blocking situation when, after a more or a less fruitful period of trial, a learner still fails to progress. Therefore, his behavior is characterized by a refusal, a provisional and an apparent inability to pursue the learning and to react to a situation (GDP Larousse).
- ***Demotivation situation***: according to the author in [41] "the students in learning difficulty have often motivational problems. Their difficulties to learn, their many failures and their image in the eyes of other students bring many of them to demotivate and to loss all interests to learn in a school context". According to this principle, we can reveal a difficult situation through the value of the motivation indicator where, according to the same author, "the students that have learning difficulties are more likely to have a low motivation or a fragile motivation than the others". That is to say, if the motivation indicator value is low, the learner is in a difficult situation.
- ***Situation of task accomplishment***: in the training session, the learner must learn, improve and develop his knowledge and his skills from the professional tasks that are the well-defined units of work. Each unit is divided into an organized sequence of steps for the tasks of the procedure kind or the guidelines that have to be applied for the tasks based on principles, like "to organize a conference". Therefore, the learner can be blocked in a stage or in a guideline that prevents him to accomplish his task correctly and to achieve the course's goal.
- ***Passage situation*****:** the content of the online training can include the learning resources, the interactive online lessons, the electronic simulations and the work tools. These various components put the learner in a confusing situation, where he is not capable to select the optimal methodology of navigation and he will not be able to answer a sequence of questions: Which path, I will choose? ", "What component, I will consult? ", "what do I have to do now? "," What is the next step". In this case, the assistant should intervene in order to provide to the learner guidelines for achieving his learning goal.
- ***Help request:*** in the learning sessions, the learner reacts according to the context and the tasks; ask a question about the detention, the steps that follow, the time that he is allotted and the organizational succession. Through these questions the learner looks for steps to be followed in order to realize the job he has to do [18].
- ***Difficulty:*** being in a situation of learning difficulty means having problems at level of perception, understanding and (or) use of concepts. These problems cause a delays in a development and (or) difficulties with one or all of These aspects: organization in reasoning, attention, memory, reasoning, coordination, communication, reading, writing, spelling, calculation, social skills and emotional maturity.
- ***Disorientation:*** to be disoriented in terms of progression in the learning system, means having difficulties for: identifying his position in the structure of information, reconstruct the path which has brought him to this node, discern the possible choices which available to him, select a destination and generate a path to a node which he knows his

existence [12]. In [11] the authors defined the disorientation by the difficulty to extract the information in order to realize a task or perform the treatments on content such as the understanding, the selection or the learning.

4.2 Indicators identification

In order to provide to ILE's learners a proactive and a personalized assistance, the revealing indicators of their difficulties have been proposed. The revealing indicators are triggering conditions of the assistance, which through their values, the system detects whether the learner in a situation that needs help or not. The calculation of these indicators is based on the analysis of learners' interaction traces with ILE.

For identifying these indicators, we acted on a studied of state of the art on learning indicators, digital traces, as well as the assistance in ILE, on the one hand. On the other hand, we made a survey (on needs of online assistance) with several teachers, for exploiting their experiments to determine some parameters allowing identifying signs and characteristics of the difficult situations encountered by the learners during a learning sessions.

In order to provide generic indicators, we asked the teachers to answer a set of question on (observed actions, training time, number of sent messages, rate of individual production, forum participations, percentage of activity realization, etc.) which necessary for obtaining global view on learning situations. As a result, in the 13 responses collected, we identified a set of indicators that have been classified according to three dimensions: time indicators "The temporal aspect", indicators on the learners' interactions "interaction aspect "and indicators of learners' production "The aspect of production and progression".

4.2.1 The temporal aspect

The training time remains always an important factor, to indicate the learner's progress. In psychometrics, many studies have demonstrated its utility as an easiness indicator with which the learner complete a task [21];[38];[40];[43]as a task's difficulty indicator [17] ;[23] ;[44], and also as a motivation indicator in ILE [1] ;[7].[8].

In order to detect the assistance situations, we considered that is very important to define the following temporal data:

- The real-time of training in active session.
- The inactivity time: This parameter can be an indicator of blocking cases.
- The time spent for passing of task to another: this parameter can be an indicator of possible disorientation.

4.2.2 The interaction aspect

To achieve relevant assistance, the system must analyze the learners' interactions. For this reason, we have taken in consideration four types of interaction: cognitive interactions, social interactions, interactions of navigation inside and outside the ILE (action on machine files or free navigation on the web).

➢ ***Cognitive Interactions:*** this type of parameters gives some information about learner and his handling method of objects and pedagogical resources available in learning environment. From this information, the

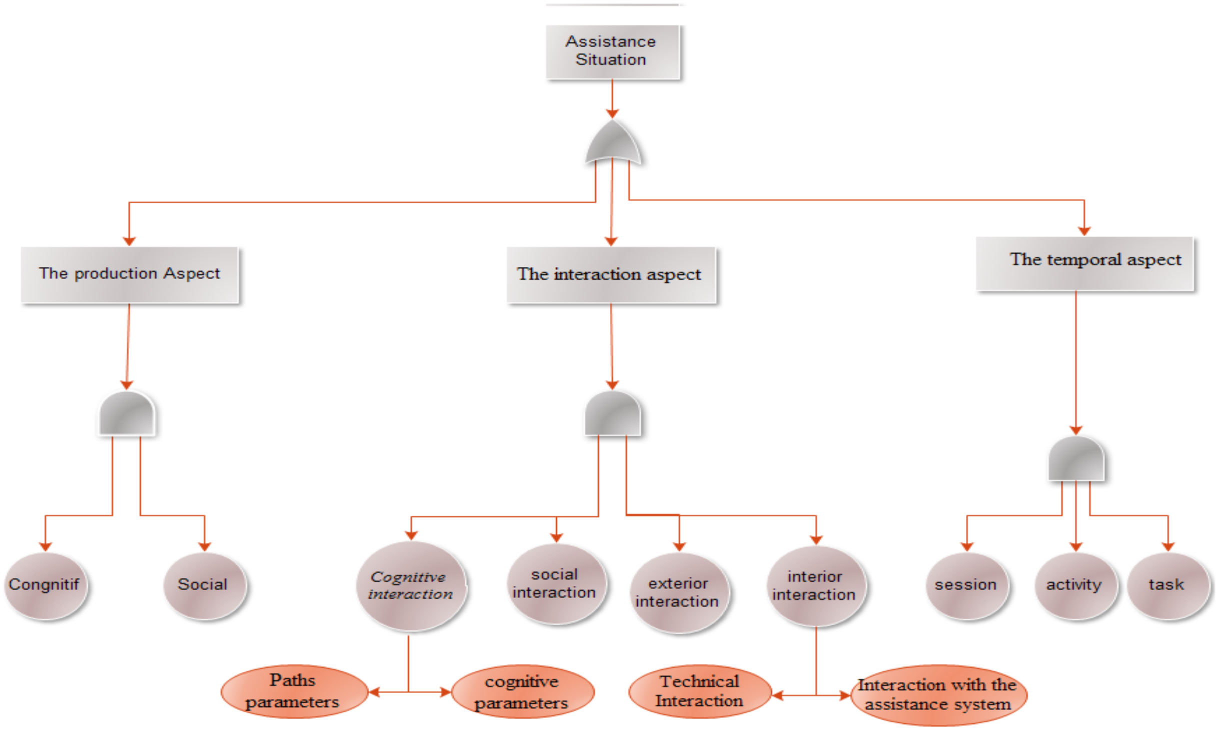


Figure 3: Indicators Classification.

assistance system can comprehend the learners' objectives. Two subtypes are distinguished:

- Information about the use and the handling of resources called the Paths parameters. For example, a learner in ILE can manipulate pedagogical objects through the access to its content, to certain of its technical components that can be buttons, scrollbars (scrollbar), and a menu. And can to navigate to a learning object to another. This kind of parameters may allow the assistance system to define the critical moments in relation to the environment functions (for example: as the learner clicks the button X)
- Information about the knowledge manipulated by learners, we call them cognitive parameters. These parameters allow the personalization of assistance according to information related to the acquisition strategy of knowledge, of expertise and of problem resolution adopted by the learner.

➢ ***Social Interaction:*** the social parameters inform about learners' communicative interactions with the other participants (tutors- teacher and peers) by communication tools (email, forum ... etc.). For example, a learner uses the forum in order to get some information about the course, seeking the help of his tutor or his teacher. This type of parameters allows the assistance system for detecting some situations of help request through the semantic proximity indicator between sent mails, consulted forums and the course content.

➢ ***Navigation interactions inside the environment:*** this type of parameters gives information about the use and the handling of tools and about system functions. We distinguish two subtypes:

- *Technical interaction*: These parameters provide information about any technical problems that may encounter the learners during the use of resources or tools of ILE.
- *Interaction with the assistance system*: Through the collection of this information, the system can define the practical difficulties that learner may meet during learning sessions. For example: according to the search keywords used by the learner during his interaction with the assistance system.

➢ ***Navigation interactions outside the environment:*** Some studies as well as the authors in [6], they have shown that most users, when they encounter difficulties, they prefer, either to use the training for acquiring the preliminary knowledge necessary to the software use or to call for other participants through communication technologies (mails, discussion forums). Consequently, we aim to analyze the learners' interaction and navigation traces outside the ILE during the learning session in order to identify these difficulties situations.

The idea is to save these interactions and to consider them as a source of knowledge that the system can use to detect assistance situations on one hand and to construct and to update the contents of the help, on another hand.

4.2.3 Production and progression aspect

During the training sessions, the learner discovers the software at the same time while he tries to accomplish his learning task, hi is thus facing to double difficulties: learn to use the software functionalities and to apply his knowledge and skills. In order to detect this type of difficulties, the system should calculate some indicators such as, progress indicator in the course, proportion indicator for learner's productions, the progress rate indicator, indicator of success and completion for each activity and task, indicator of percentage and realization level of the activity.

These indicators can be classified into two types: cognitive and social.

- Cognitive: we link to the cognitive type, every indicator that gives to the system the ability to obtain the information about the personal progress of the learner and on the realization percentage of his activities.
- Social: The calculation of this type of indicators offers to the system the information about the collective production of the learners and about the global progressions of the groups.

To understand the process allowing to calculate these indicators and to deduct the situations of assistance needs, our solution is explained in the next section.

4.3 Revealing indicators of difficulties

In order to make our assistance approach applicable for ILE, we were interested in proposing indicators that can be produced from learner's actions trace. Our approach aims at exploring the possibility of deducing the difficulties encountered by the learner based traces interaction.

According to the classification of indicators and the category of assistance situation, we proposed five indicators with their thresholds and their formulas. For the selection of these thresholds, we have made several adjustments to the calculation of each indicator based on preliminary test, in order to estimate the appropriate values. However, our system offers the ability for assistance designers to change these thresholds. The selected thresholds shown in Table 1.

	Inactivity maximal threshold	Semantic similarity threshold	Threshold for the Accomplishment
Selected value	3mn	0,3	50%

Table 1: The value of thresholds.

▪ **Blocking Indicator**: This indicator can be measured by the calculation of the inactivity time during the learning session. However, it is not always easy to distinguish the inactivity time for actual working time (real time learning) [4]. For this, we are oriented towards the analysis of the navigation interactions outside the learning session. This allows knowing, if the learner tries to accomplish his task, or on the contrary, he wants to change it or abandon it completely.

In order to inform whether the learner is more active or not, we have used an indicator of interactivity rate that takes into account the number, the type and the handling duration of action.

$$R_{int} = \frac{\sum_{i=1}^{n} numbre\ of\ actions}{\sum_{j=1}^{m} numbre\ of\ page\ concepts} \ldots\ldots\ldots.(1)$$

N: number of actions performed by the learner on the page and m: number of page concepts.

If the rate of interactivity tends to 0, this means that the number of learner's actions on the content of the task is relatively low, which implies that the learner is inactive, so he is in a blocking situation requiring an intervention of assistance. However, if the number of learner's actions on the content of the task is relatively high, the rate of interactivity tends to 1 or more, which implies that the learner is active. In this case, the system will start the calculation of another indicator: the indicator of semantic similarity between the content of the task and the content of the pages visited by the learner during the learning session. This indicator was inspired from the work of [4] who studied the semantic similarity between the courses and pages visited during the learner navigations [19].

Sim(c_i,p_i): the semantic similarity between the studied course and the visited page. It is calculated by using the following formula [4]:

$$Sim(q,d) = \frac{\sum_i q_i \times d_j \times sim(i,j)}{\sum_i \sum_k qi \times d_k} \ldots\ldots\ldots.(2)$$

With:

- I is concept of the document q (the course);
- J is the concept of document d (page having the maximum similarity with i);
- K represents the concept of the document d;
- q_i And d_j are the weight of the concepts i and j in documents q and d;

Sim (i, j) is the semantic similarity between concepts i and j, calculated by the Lin's formula [19].

$$\mathbf{Sim(i,j)} = \frac{\mathbf{2.IC(lcs(i,j))}}{\mathbf{IC(i)+IC(j)}} \ldots\ldots\ldots.(3)$$

Where IC determines the information content of a concept and LCS finds the lowest common subsuming concept of concepts i and j.

If the semantic similarity is relatively high, it means that, the learner has consulted other content having the same subject of task, either in order to learn more or to explore new perspectives. Here, we estimate that this learner is inherently active and therefore is not blocked. But if semantic similarity is very low, it means that the learner has changed his task or abandoned according to waste of motivation facing the difficulties encountered. In this case, the learner may be in a blocking or abandonment situation.

▪ **Indicator of Activity Accomplishment rate:** The educational structure of a course in ILE is divided into modules, activities and tasks. In order to calculate our indicator of accomplishment, we used the principle of Despres [9], where each task is a feasible entity in all or nothing, that is to say, which can take only two states: "accomplished" or "not accomplished." Thus, for each task, we must associate a weight and a coefficient. The task weight sum of the same activity must be equal to 1. Therefore, the percentage of activity accomplishment is calculated according to the weight of the tasks that the learner has accomplished.

$$R_{\mathbf{Accomplishment}} = \mathbf{100} \times \sum_{i}^{n} Pi \ldots\ldots\ldots.(4)$$

With Pi: the weight of tasks, n: the number of completed tasks.

- ➢ If $R_{\mathbf{Accomplishment}}$ ≥ 50%, we estimate that the learner has completed his tasks, he does not need a help and if he needs, he asks the assistance.
- ➢ If $R_{\mathbf{Accomplishment}}$<50%, we estimate that the learner is in a difficult situation and he needs a help to accomplish his tasks.

▪ **Motivation Indicators:** Three types of indicators are defined in the literature in order to measure learners' motivation: the indicator of interest and pleasure to accomplish a learning activity, the indicator of cognitive engagement, and the perseverance indicator [41].

- ❖ **Indicator of Interest**: it is very difficult to measure the learner's interest. However, in our case, we have use only the rate of concentration to calculate the interest. This indicator can be measured by comparing between the activity developed outside the training content and the activity performed on the platform or the course and depending the recorded periods of inactivity.

$$\mathbf{I_{interest}} = \frac{\mathbf{\sum_{i=1}^{n} sim(Ci,Pi)}}{\mathbf{N}} \ldots\ldots\ldots.(5)$$

With: Ci: the performed concept of task; Pi: the page visited during the performing of task; N: number of pages visited.

This indicator was used by author in [39] to determine the types of motivation and by authors in [4] to determine learning styles. But it will be borrowed here to determine the difficult situations. However, this indicator can have three values:

➢ *Low interest*: if its value tends to 0, the semantic similarity between the proposed task and the content of visited pages is very low. This means, that the learner has a low interest to the proposed task and consequently, he is not motivated. In this case, we assume that he is in a difficult situation.

- *Medium*: If its value is equal to 1, this means that the learner has consulted only the content offered in training. This means, that he was not interested essentially by the activity itself, but he accomplishes it, because it is imposed. In this case, the learner is not in a difficult situation.
- *Interested*: If its value is greater than 1, the learner consults other contents that are semantically close to the content of the proposed task. This means, that he has consulted these contents voluntarily and by interest. In this case, it seems that he is motivated and thus is not in a difficult situation.

- **Indicator of Cognitive Engagement degree:** To calculate this indicator, we based on the principle of Rolland [44] that reflects the cognitive engagement by the fact that a student or learner will navigate in depth on web sites. In other words, he will examine all aspects of these sites and not only the images or the sound effects. Moreover, we consider that all concepts are characterized by a depth measuring their distance from the root.

$$\mathbf{D}_{\mathbf{Cog-eng}} = \frac{\sum_{\mathbf{i}} \mathbf{Di} * \mathbf{depth(ci)}}{\sum_{\mathbf{i}} \mathbf{Di}} \dots\dots\dots.(6)$$

With depth (ci): the depth of the *concept Ci and Di:* Di: the consultation time of concept Ci.

This indicator can have three values:

- ***Weak:*** if the learner consults the learning environment superficially, that is to say the learner examines only home pages or images and sound effects.
- ***Average***: the learner uses an intermediate navigation strategy.
- ***Strong***: If the learner's navigation strategy is more in depth. In other words, the student consults and examines the concepts of great depth.

- **Perseverance Indicator:** the perseverance is one of the indicators that seem more relevant to measure learning motivation in distance education, than the interest and pleasure to accomplish the task. According to the author in [44], we may be interested in sound or visual effects of a website and take pleasure in "surfing" on its contents without actually learn. This perseverance is manifested by the time that the student devotes to accomplish a task or an activity and the number of times that he repeats it [44].For this, an indicator of correspondence rate between the realization real time of the activity and the time spent by the learner for performing this activity with a component for repetition, has been defined.

$$\mathbf{R_C = D_{r\acute{e}al}/T_{effect} * N_{rep}} \dots\dots\dots.(7)$$

Where:

- R_C is the indicator of "correspondence Rate"
- $D_{réal}$ is the realization period.
- $T_{effect}$is the effective realization time defined by the teacher or the course designer
- N_{rep} is the number of repetition

The different scenarios:

- ***Scenario 1*:** If R_C> 1, this means that the learner spends more time for doing his activity. In this case, we assume that the learner has persevered to accomplish his task despite the obstacles and difficulties that he encountered.
- ***Scenario 2*:** If R_C<1, and the accomplishment rate indicator inferior than 50%, this means that the learner spends less time to perform his activity. At this point, we assume that perseverance of the learner to accomplish his activity is very low and he is considered as not motivated. We assume that these indicators are factors of dropping out in learning and so, we are in difficulty situation.
- ***Scenario 3*:** If $R_C \leq 1$ and accomplishment rate indicator is superior to 50%, it means that the learner is able for doing his activity in the effective time. In this case, we assume that the learner's persistence and motivation tend to average, which implies that the learner is in a normal learning situation.

- **Disorientation Indicator:** The ILE can be considered a problem space when each movement or interaction is a revealing of the learner's cognitive activity type (understanding, disorientation, information confrontation, etc.). In this meaning, the authors in [33] proposed four variables for revealing the disoriented learner:
  - Redundancy (red): This variable measures the number of nodes opened more than once by the learner during the learning phase.
  - The number of additional readings (read-addi): the learner can ask for additional reading to understand the concept in progress. If the request is repeated several times, this means that there is a problem.
  - The number of additional solutions (solu-addi): the number of times that the learner answers questions he has already given a solution, then he can continue his work.
  - The number of returns to a previous part (ret): The problem arises if the learner has acquired concepts and he asks, subsequently, to return to these concepts.

We take these variables and we add another to define our disorientation indicator:

The Number of Clicks (nb-clicks): the problem arises if the learner clicks on many concepts or areas of the screen for a relatively long time. This may be a sign that he is looking for something or he is lost in the learning environment.

We consider as a disorientation indicator, the sum of the values taken by these variables:

$$I_{dis} = \mathbf{red + read{-}add + solu{-}addi + ret + nb{-}clicks} \ldots\ldots\ldots(8)$$

The different scenarios:

- ***Scenario 1*****:** When the values of these variables are low, it means that the learner is well oriented in the environment.
- ***Scenario 2*****:** If a variable value is high, it does not necessarily mean that the learner is disoriented, but, it can rather confirm a learning style.
- ***Scenario 3*****:** when several variables are characterized by high values, this indicates disorientation in the environment.

- **Indicator of Help rate:** During the learning session, the learner can seek help, in order to get out of a blocking situation. Depending on the nature of this solicitation, we can determine: the disorientation of the learner, his navigational difficulties in the learning environment, his apprehensions to go into details... etc. And we can also detect its failure signs (personal, technical, skills...).

$$\mathbf{R_{Help}} = \frac{\sum_j \mathbf{Hj}}{\sum_i \mathbf{Ci}} \ldots\ldots\ldots.(9)$$

With Hj: the number of call for help and Ci: the number of task concept**.**

4.4 Indicators calculation

To calculate the five revealing indicators, we propose, a treatment process composed of two steps: collection of traces and transformation.

- **Collection of interactions' Traces:** three approaches are available to perform the collection of traces [4]: user-centered design approaches, server-centric approaches and specific software-based approaches. Each approach has some advantages and disadvantages. In order to have a complete perception of all learners' activities and the information about his inactivity time during learning sessions, we have opted for the user-centric collection approach through a program installed on the learner's machine. Therefore, we have used an existing keylogger. Among the collection software available in the free version, we have chosen MiniKey; it allows saving all the actions of the learner on his machine, whether made inside or outside the ILE in real time.

However, the traces generated by this tool are primitive and difficult to exploit as such and required a transformation and a modelization. Moreover, it must undergo a pre-treatment process to eliminate the noise (not found pages, URLs wrong)

- **Transformation and formalization of traces:** Generally, the traces collected by the keylogger software are digital traces that contain rich information about user's behavior. However, the collected traces quantity is usually huge and the traces are very detailed which makes the interpretation process difficult on the analyst side [24]. A transformation mechanism is necessary for obtaining an adequate volume of traces at the right level of granularity that makes the interpretation process easier. This transformation process is based on some methods: cleaning (To eliminate the noise), filtering (To extract relevant tracks according the objective of the analysis) and traces structuring (to structure and to model the tracks of interaction).

Our aim is to use the traces of learners' experiences in order to calculate the revealing indicators of difficulties and in the absence of a standard trace model. We applied the model proposed in IDLS [4] to our context of assistance.

The trace is defined as an observed temporal sequence, providing the information about the learner's actions collected in real time from his interaction with the learning environment and with external resources, such as files or programs running on his machine or on the Internet. Formally, **T <U, (O1, O2, O3 ... On)>**

- U: observed user,
- Oi: observed trace. Each **Oi** is a couple of (Pi, Ai) , when:

Ai is a learner's actions, for each action Ai we have recorded all learners' interactions with the content of the page, when each action is identified by:

- An order of appearance
- A date, hour and execution time
- A type of action: (mouse actions: left, right or middle button click, the keyboard keys: F1, F2, CTL + C and CTL + X, etc.)
- An object on which the action was performed (scroll, link, text, picture, menu, etc.)
- And the type of interaction performed (open a file, search, copy, paste, print, etc.).

Pi is a page consulted by learner, for each page Pi we have stored:

- URI (Uniform Resource Identifier)
- title T, if it is available
- the program that have executed it PG
- content, we distinguish three types:
  - Pedagogical resource "R" if the content is a set of concepts (courses).
  - Learning activity "AL" if their content is an exercise, a test or homework.
  - Browsing activity "AB", if their content is a file or a program executing on the learner's machine or on the Internet.
- Order of appearance.

5 The environment

The MISNA Environment (Modeling and identifying the situations of Needs for assistance) implements our theoretical propositions through two main tools: an assistance editor and an assistance situations detector.

5.1 The assistance editor

The assistance Editor is a tool intended to the assistance designers. It implements the IMAC model and allows us to specify an assistance system described by a set of assistance situations. The assistance editor provides two interfaces, one for the learning environment description and another for the creation of each assistance situation.

- **Description of Learning Environment**

The aim of this phase is to describe the components that the designer wishes associated to the assistance interventions. This description is based on the Reflet representation, presented in [9]. This representation allows describing the environment according to a tree structure of three levels: MAT (Module-Activity-Task). In this representation, a module can contain another modules (sub-modules) and/ or activities but an activity contains only tasks. Thus, a task may be homework, exercise to do or course to study. These different types of tasks will be represented in the form of web pages or screens composed of several objects (link, text, image, menu ...). The various components of the environment are grouped according to their type (Module, activity, task) in a database; this base can be advanced throughout the life cycle of the system through the analysis of learners' interaction traces.

An interface has been developed for the creation of each basic component without any programming skills. A snapshot of our interface is given in Figure 4; the interface is divided into two parts: The left side (see Ⓐ Figure 4) presents the components that have already been created. The right side (see Ⓑ Figure 4) allows creating, modifying or deleting a component (Module, Activity and Task).

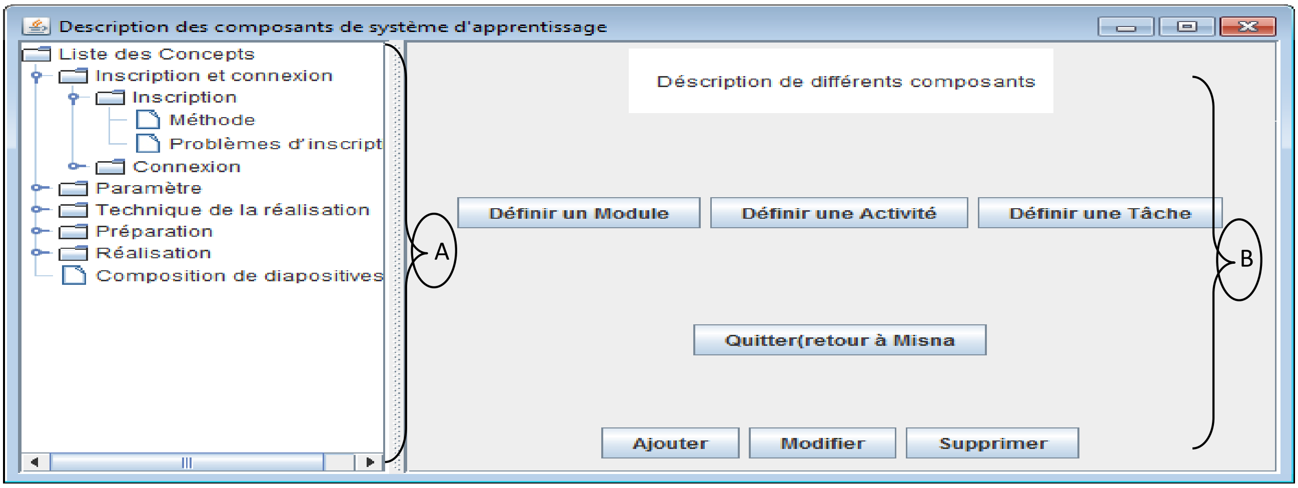


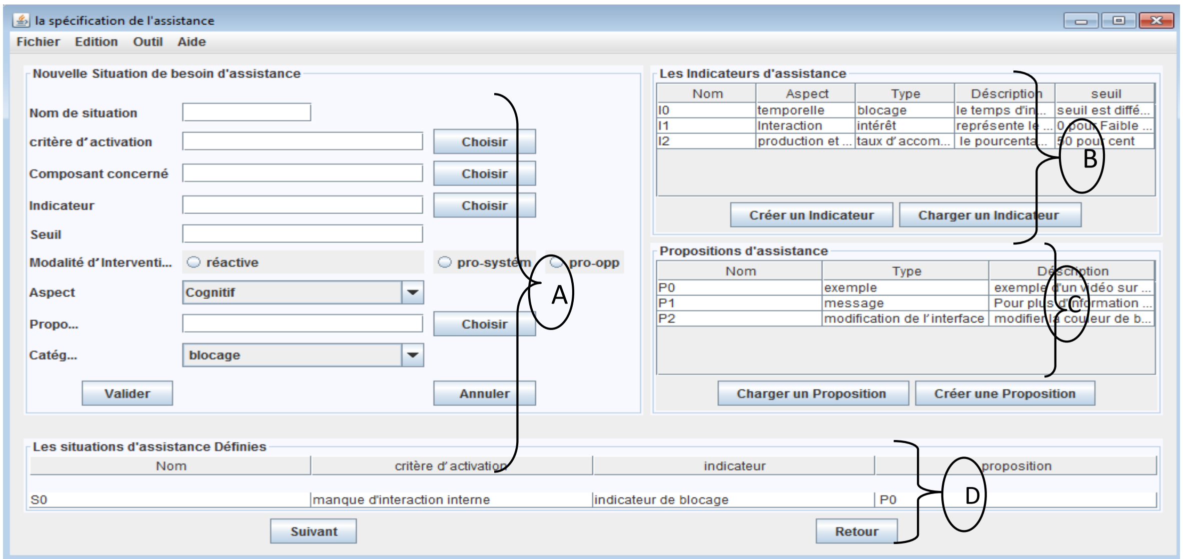
Figure 4: Screen-shot of system description.

- **The Assistance Specification**

After the description of the environment, the designer proceeds to the specification phase. This phase aims to characterize the assistance in learning environment based on a formalism that models the assistance in the form of a couple formulated in a difficult situation and an assistance proposal. Our IMAC model is used for modeling the difficult situations and three forms of assistance are adopted for defining the assistance proposals: messages (in the form of an adjacent page to provide additional information to the learner), examples (like demonstration videos) and the change in the environmental interface (either by adapting or modifying links: addition, deletion, annotation, sorting ... etc.).

In order to implement this step, we developed an interface that allows designers to enrich their learning environment through an assistance system without any programming skills. The designer must create the revealing indicators of difficult situations as well as the assistance proposals.

Figure 5 shows the interface that presents the specification phase. This interface is divided into four parts: the part Ⓐ allows creating a new situation of assistance needs. The partⒷ and Ⓒ present respectively indicators and created assistance proposals. And finally, the part Ⓓ presents the defined assistance situations.



la spécification de l'assistance

Nouvelle Situation de besoin d'assistance

Nom de situation	
critère d' activation	
Composant concerné	
Indicateur	
Seuil	
Modalité d' Interventi...	○ réactive ○ pro-systém ○ pro-opp
Aspect	Cognitif
Propo...	
Catég...	blocage

Les Indicateurs d'assistance

Nom	Aspect	Type	Déscription	seuil
I0	temporelle	blocage	le temps d'in...	seuil est diffé...
I1	Interaction	intérêt	représente le ...	0 pour Faible ...
I2	production et ...	taux d' accom...	le pourcenta...	50 pour cent

Propositions d'assistance

Nom	Type	Déscription
P0	exemple	exemple d'un vidéo sur ...
P1	message	Pour plus d'information ...
P2	modification de l' interface	modifier la couleur de b...

Les situations d'assistance Définies

Nom	critère d' activation	indicateur	proposition
S0	manque d'interaction interne	indicateur de blocage	P0

Figure 5: Screen-shot of assistance specification.

Each situation is characterized by a set of predefined parameters. The assistance designer must specify the types and the thresholds, as well as certain additional information. Consider an example related to the designer when proposes a difficult situation of blocking type, first, he must specify the detection sign (activation criteria), and he must choose, from the list of indicators (predetermined by the assistance system) a favorable indicator to this situation, that is the inactivity time indicator. This indicator is varied according to the type of the task. The inactivity time of a reading task is not the same as of a realizing a duty or of an exercise. For this, the designer must identify a threshold for each indicator. In addition, he must specify the aspect and the category of the situation and the concerned component, in order to facilitate, on one side, the indexing of situations in the creation step and on the other side, the search for a similar situation at the problematic situation in the step of executing assistance.

5.2 The assistance situations detector

In order to provide to ILE's learners, a personalized assistance in a proactive manner, we have developed a detector of assistance needs. This detector exploits the database of the assistance situations created in specification phase and the database of modeled interaction traces, in order to identify the learner in difficulty.

Over the course of learning, when a sign of an assistance situation is detected, the detector starts the information process, in order to determine some parameters about this problematic situation (concerned component, aspect and category of situation). This information will be the source of knowledge for the selection process used to exploit them in order to select the suitable situation to our problematic situation. At the end of this selection, the detector initiates the indicators' calculation process. According to the confrontation between the indicator's value and the threshold's value, the detector decides either to start the execution process to realize the proposals associated to this situation or to ignore them.

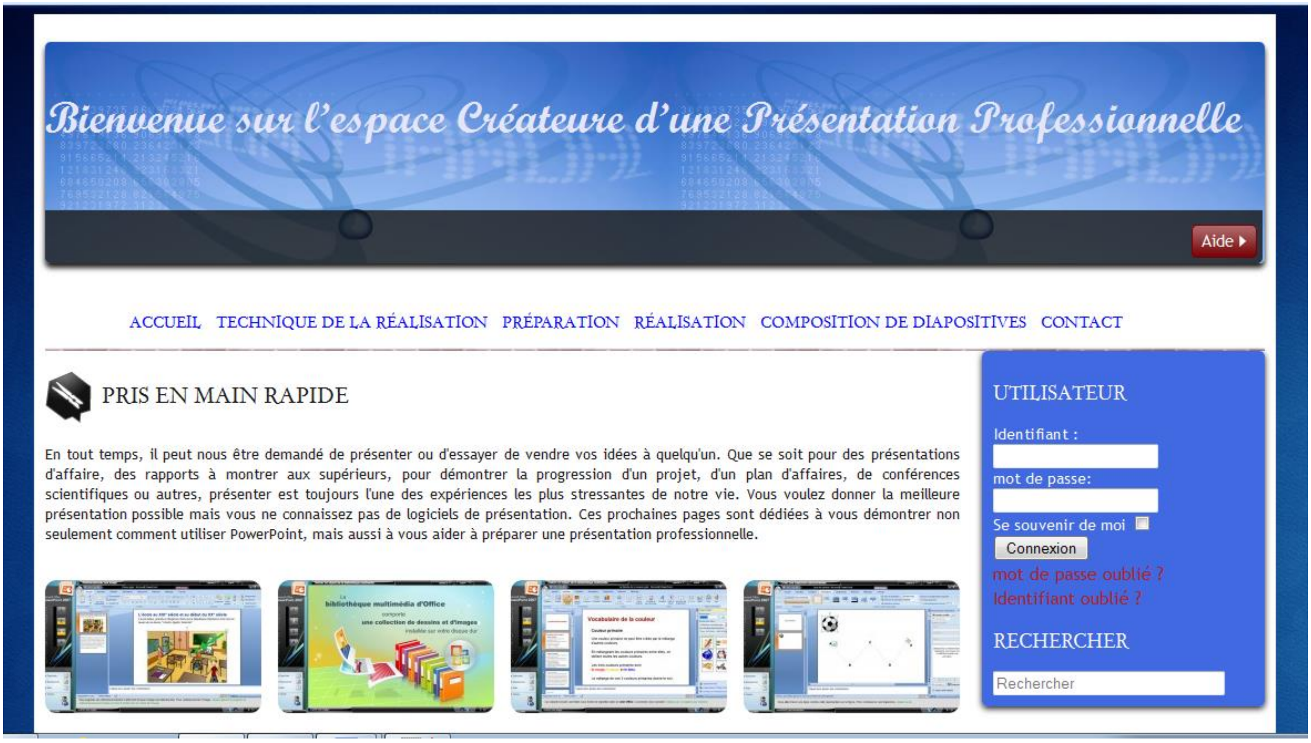
6 Experimentation

To validate our suppositions about deducting assistance situation based interaction traces and the ability for providing a proactive assistance thanks a detection of an assistance needs situations and learner‘s difficulties, we designed a website including forms and assistance means with the theoretical models proposed(Figure 6). This application was designed for undergraduate students at the Faculty of Human Sciences, Department of Communication at Badji-Mokhtar University of Annaba in Algeria. It aims to help these students in the course of professional presentations preparation.

Before, obtaining the license degree, it is expected that students are able to make a professional presentations using the computer software. Therefore, the Practice Works (PW) sessions have been proposed, which are assisted by teachers on computer science. However, each year, these teachers observe the problems about the methodology of presentation preparation, on top of that, the problems about the use of a computer software (like PowerPoint for example). The design of our site can be a solution to these problems; It provides to students the explanations and examples (such as Know-how) in addition to the basic concepts (knowledge) given in masterly course.

In order to test our suppositions, three series of experiments were conducted: two with teachers and the third with learners.

The first experiment involved six (6) assistance designers, three(3) teachers of methodology have a PHD Degrees in Communication with seven (7) years of experiments in the communication department and three (3) teachers of computer science are PHD students, with four years of experiments associated at the communication department. We asked these designers, in four (4) sessions of (3) hours, to work in collaboration,



Bienvenue sur l'espace Créateure d'une Présentation Professionnelle

PRIS EN MAIN RAPIDE

En tout temps, il peut nous être demandé de présenter ou d'essayer de vendre vos idées à quelqu'un. Que se soit pour des présentations d'affaire, des rapports à montrer aux supérieurs, pour démontrer la progression d'un projet, d'un plan d'affaires, de conférences scientifiques ou autres, présenter est toujours l'une des expériences les plus stressantes de notre vie. Vous voulez donner la meilleure présentation possible mais vous ne connaissez pas de logiciels de présentation. Ces prochaines pages sont dédiées à vous démontrer non seulement comment utiliser PowerPoint, mais aussi à vous aider à préparer une présentation professionnelle.

UTILISATEUR

Identifiant :

mot de passe:

Se souvenir de moi □

RECHERCHER

Figure 6: Welcome screen-shot of MISNA website.

for describing the training content. The aim of this collaboration is the description of training environment pedagogically, technically and administratively. After this experiment, we obtained, a structured representation composed of six modules: registration and connection, parameters, realization techniques, preparation of presentation, realization and finally composition and animation of the slides. Each module is divided into one or more activities and each activity consists of one or more tasks. For example: the parameter module is divided into two activities: site parameter and learning parameter. The first one contains two tasks: parameter of site usage and parameter of the contact.

In the second experiment, we asked our assistance designers team (teachers of two modules: Computer science and methodology) for passing to the second phase of experiment, the specification phase of assistance system. Thanks to specification interface of our MISNA system, the designers defined 112 assistance situations (associated to the six (6) predefined modules), 51 indicators, 67 thresholds and 138 proposals of assistance. The assistance needs situations appear in different types (pedagogical, technical, cognitive, and methodological) and the proposals appear in three types: text messages, modification of the interface and examples. The message content is either prerequisite proposals or explanations of the following steps (Figure7).

We note that the majority of the proposals are example or message type. Probably, because the designers think that these two types are more adapted to their students' learning style.

Finally, in spring semester 2015, we asked our students to use our training site during a PW session for 90 minutes, after 5 minutes of explanation about the

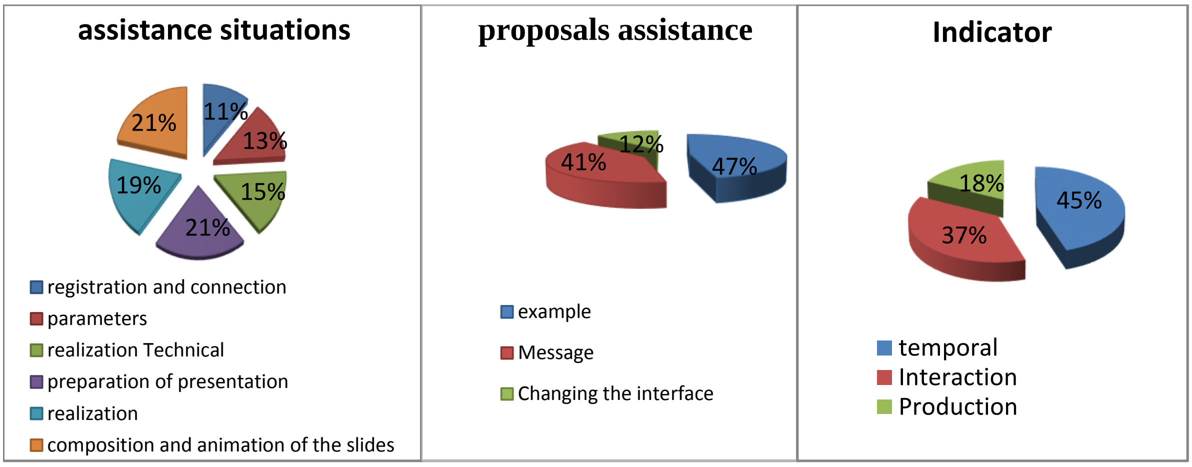


Figure 7: Results of the specification phase.

experiment objectives. Thanks to this explanation, the students are motivated to participate in our study.

This experimentation was applied with 40 undergraduate students, 14 male and 26 female, aged between 20 and 24. Their participation was voluntary and they were divided into two (2) groups of 20 students due to the number of available machines. In order to avoid the confounding variable, we have fixed the values of age and previous knowledge and we have divided the participants in equal way, where the both groups are equal in gender, age and previous knowledge according to their exam results of the first semester.

The students performed a professional presentation of a proposed text in the communication field. The proposed texts are reports made by the same students on themes related to their thematic. At the end of the presentation creation step, participants were invited to fill in a satisfaction survey about their experience using MISNA in order to measure the contribution of the interventions and assistance propositions. The learner declares his views on the relevance of each intervention by the satisfaction or non-satisfaction. This satisfaction was measured by a global manner, in two contexts; one is on time and mode of intervention and the other on assistance content.

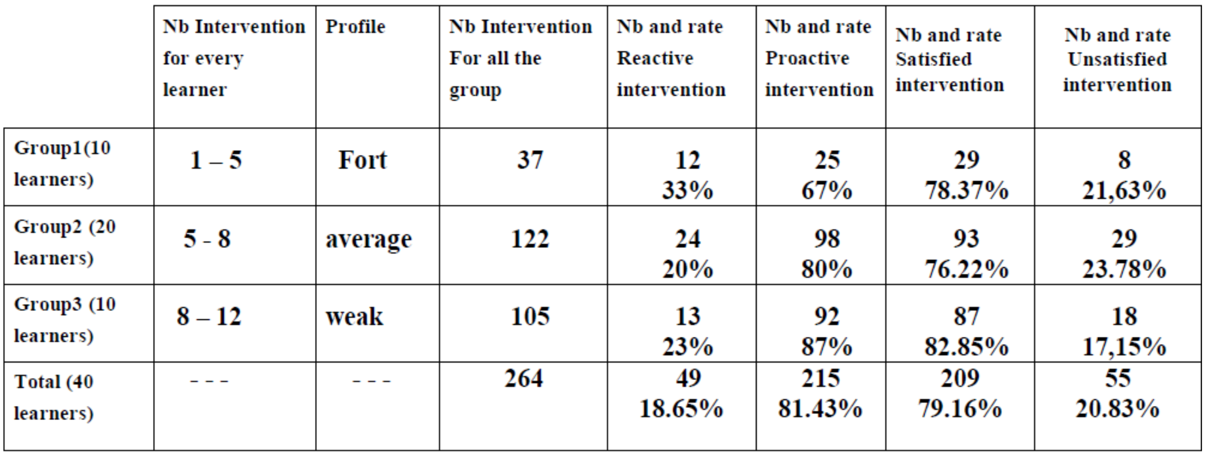
In order to validate the proposed indicators with a data from the experiment and to verify with the students the trigger conditions, our system saves the learners' interactions after each assistance intervention, in order to evaluate the effect and the contributions of proposed assistances, the interactions are stored in log file (in XML format).

7 Results and discussion

The satisfaction survey shows that learners are relatively satisfied by the assistance interventions in particularly by those that have been triggered through an alarm from indicators such as interaction and temporality. These indicators have enabled us to easily identify learners that are lost in the learning environment or which take more time to complete their tasks. The blocking indicator and interactivity rate allowed us to get a global view of the type and nature of learners' interactions. This view allows to identify the real-time learning, and to distinguish between abandonment situations and blocking ones. These indicators show that the semantic similarity between the offered courses and the content of the visited pages, during the inactivity time, is very high which proves that (87%) of inactivity situations are situations of assistance needs. This result seems in agreement with those in the studies of [6], who found that the learners prefer to resort to the training in order to acquire the necessary knowledge to use the software and that the occasional users appeal their colleagues to overcome blocking situation (Table 2).

The three indicators "degree of cognitive engagement, correspondence rate and interest indicator" allow revealing the learners that lose their motivation in front of the difficulties they encounter during learning sessions. However, the learners reject 40% of assistance interventions identified by the interest indicator which indicates that, if the learner has a low interest does not mean that he is in a difficult situation, but he may be in a task change situation or passing to another stage. With regard to the cognitive engagement degree indicator, the experimental results show that the majority of learners (if not all of them) use the same learning strategy in average depth, which puts us in contradiction with the basic rule that indicate: each person has own learning strategy. These results forced us to examine other indicators such as the indicator of achievement and progression for these critical situations. Following this analysis, we found that 25% of learners use a surface learning strategy after the completion of their tasks. Therefore, we decided to link the cognitive engagement indicator to the task completion indicator in order to distinguish between the learner in difficulty and the one who borrows all the components of the environment in search of a global vision. Thus, regarding the perseverance indicator, 89% of assistance interventions related to this indicator are accepted by learners, which shows that the connection between the correspondence rate and accomplishment rate is very fruitful as well as the motivation rate.

At the end of the second step of experimentation (system specification), we identified the preliminary



	Nb Intervention for every learner	Profile	Nb Intervention For all the group	Nb and rate Reactive intervention	Nb and rate Proactive intervention	Nb and rate Satisfied intervention	Nb and rate Unsatisfied intervention
Group1(10 learners)	1 – 5	Fort	37	12 33%	25 67%	29 78.37%	8 21,63%
Group2 (20 learners)	5 - 8	average	122	24 20%	98 80%	93 76.22%	29 23.78%
Group3 (10 learners)	8 – 12	weak	105	13 23%	92 87%	87 82.85%	18 17,15%
Total (40 learners)	- - -	- - -	264	49 18.65%	215 81.43%	209 79.16%	55 20.83%

Table 2: Some Results related to the experimentation.

thresholds for our indicators, and during the third step of experimentation some thresholds were changed according to the learning styles adopted by our students, like the variable identifying the disorientation indicator. Indeed, a little higher number of additional solutions is not always a sign of disorientation. But, it can rather confirm a learning style. The results confirm that the competent students repeat the same task until three times in order to know the different methods of possible solutions and/or to memorize well the instructions to be followed. Thus, for redundancy and additional reading, the analysis of the results shows that a significant number of learners (65%) prefer to read the concepts acquired more than twice before they start to learn the new concepts.

According to the results of our experiment and the previous knowledge following to their exam results of the first semester, the 40 students that compose the study sample have been divided into three experimental groups:

• ***Group 1:*** when learners' profiles are strong: each learner has benefited from one to five interventions, 67% proactive and 33% reactive, with 78.37% of satisfaction rate.

• ***Group 2*:** when learners' profiles are average: Each learner has benefited from five to eight interventions, 80% proactive and 20% reactive with 76.22% of satisfaction rate.

• ***Group*** 3: when learners' profiles are weak: each learner has benefited from eight to twelve interventions, 87% proactive and 23% reactive, with 82.85 % of satisfaction rate.

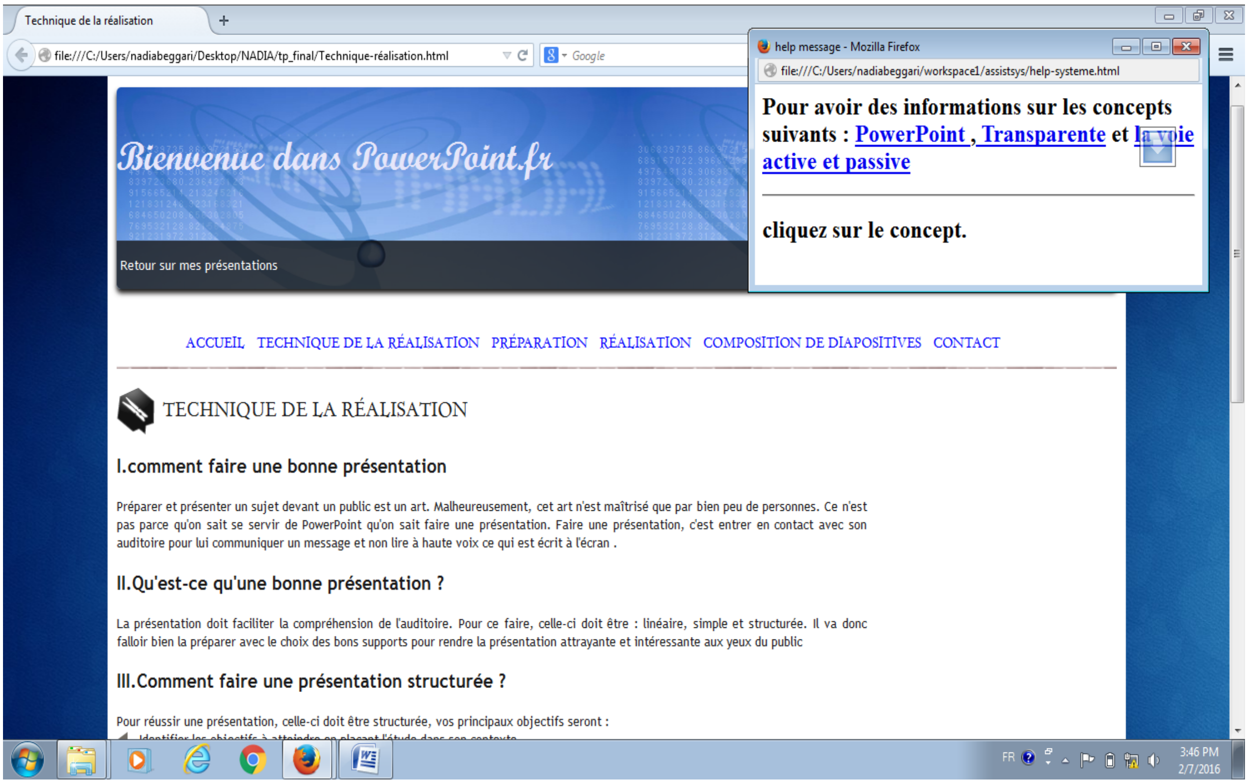
8 Case study

In order to assess the feasibility and correctness of our approach, we will present in the following, an example of the identification assistance situation in framework comprehension the theoretical concepts (technique of realization) by analyze of interactions.

The student (1) worked on machines equipped with a key-logger, with a personal account on the web site. The interaction collection started with their connection to the web site, after activation of the key-logger. At the time of the learning activity, the key-logger provides initial trace, which describes the interactions in XML. Interaction data are temporal sequence of observation (pages and actions). These traces were then processed, by the MISNA system, in order to detect the situations of needs for assistance.

During the learning, a sign of assistance situation is detected, which is the inactivity time. The detector activates the information process to determine the parameters of this situation. The information process identifies the situation by: technical realization as a concerned component, consultation of theoretical concepts that indicates the cognitive aspect and blocking or comprehension difficulties as a category. Following these results, the system activates the selection process for selecting a suitable situation to the detected situation. This selection process selects three assistance situations. In this case, the detector activates the calculate process for calculating the indicator associated to assistance situation, which is in our case, the blocking indicator.

The calculation of this indicator uses the interactivity indicator for distinguishing between the blocking and the


Technique de la réalisation
file:///C:/Users/nadiabeggari/Desktop/NADIA/tp_final/Technique-réalisation.html
Google
Bienvenue dans PowerPoint.fr
Retour sur mes présentations
ACCUEIL
TECHNIQUE DE LA RÉALISATION
PRÉPARATION
RÉALISATION
COMPOSITION DE DIAPOSITIVES
CONTACT
help message - Mozilla Firefox
file:///C:/Users/nadiabeggari/workspace1/assistsys/help-systeme.html
FR
3:46 PM
2/7/2016

Pour avoir des informations sur les concepts suivants : PowerPoint , Transparente et la voie active et passive

cliquez sur le concept.

TECHNIQUE DE LA RÉALISATION

I.comment faire une bonne présentation

Préparer et présenter un sujet devant un public est un art. Malheureusement, cet art n'est maîtrisé que par bien peu de personnes. Ce n'est pas parce qu'on sait se servir de PowerPoint qu'on sait faire une présentation. Faire une présentation, c'est entrer en contact avec son auditoire pour lui communiquer un message et non lire à haute voix ce qui est écrit à l'écran .

II.Qu'est-ce qu'une bonne présentation ?

La présentation doit faciliter la compréhension de l'auditoire. Pour ce faire, celle-ci doit être : linéaire, simple et structurée. Il va donc falloir bien la préparer avec le choix des bons supports pour rendre la présentation attrayante et intéressante aux yeux du public

III.Comment faire une présentation structurée ?

Pour réussir une présentation, celle-ci doit être structurée, vos principaux objectifs seront :

Figure 8: Screen-shot of assistance message.

abandonment situation or the changing of task. We use the indicators related to the number of action performed by the learner on the page and number of page concepts. The result is 1.2, according this value; the system will start the calculation of another indicator, the indicator of semantic similarity between the content of the task and the content of the pages visited by the learner. The student are consulted three pages by his navigator (passive and active voice, example of transparent and PowerPoint software). In this case, the semantic similarity is relatively high it means that, the learner has consulted other content having the same subject of task, in order to learn more or to explore new perspectives. We estimate that, this student is active and is not blocked, but he is in difficult situation and he needs more information for these three concepts. Finally the detector decides to start the implementation process for achieving the proposals associated to this situation, as indicated in the following figure (Figure 8).

9 Conclusion and future works

This paper discussed the possibility to identify automatically the difficult situations based on learner's interaction with his ILE. To bolster up this assumption, we first proposed an active assistance approach able of providing a spontaneous assistance to ILE's learners. This approach is based on a formalism which models the assistance situations in couple form problematic situation and assistance proposal. The problematic situation appears with IMAC model that specifies the learning moments when the learner needs assistance by a set of characteristics: a Trace indicator, a modality of intervention, an Aspect and a Category. The assistance proposal is defined by the assistance means associated to component of learning environment (messages, examples and modification of interface). Secondly, we identified five revealing indicators with their thresholds and formulas. Then, we proposed our assistance system, called MISNA. We implemented this system in an operational prototype with a learning environment, an assistance editor, trace collector and our detector of assistance needs situations. The assistance editor provides two interfaces, one for the learning environment description based on structured representation MAT (Module-Activity-Task) and another for the creation of each assistance situation according to IMAC model. The detector of assistance needs situations exploits the database of assistance situations, created in specification phase and the database of modeled interaction trace, in order to identify learners in difficulty. We conducted a first experiment in order to validate the utility of our assistance approach with 40 students.

In a future work, we intend to test our MISNA system with other learning environments. We also plan to add other indicators for providing more fertility to the detection mechanism proposed by our assistance approach. Accordingly, we want to improve the proposed mechanism by involving learners' profiles management mechanism with regard to a defined formalism. In addition, we hope to add other interfaces such as supervision interfaces, side assistant and assisted.

10 References

[1] R. Baker, (2007). Modeling and understanding students' off-task behavior in intelligent tutoring systems. Proceedings of the SIGCHI conference on Human factors in computing systems, San Jose, California, USA.

[2] N. Beggari, and T. Bouhadada (2012). Designing a model of assistance based web services in Interactive Learning Environment. Conference proceedings of "eLearning and Software for Education"), issue: 02 / 2012, pages: 5055, on www.ceeol.com.

[3] M. Ben Saad, (2008). Découverte de connaissances dans les traces d'interaction : une approche par similarité des séquences temporelles. Thesis of research Master, UCBL, LIRIS.

[4] N. Bousbia (2011). Analyse des traces de navigation des apprenants dans un environnement de formation dans une perspective de détection automatique des styles d'apprentissage. PhD Thesis of Pierre and Marie Curie University (France) and the National School of Computer (Algeria).

[5] T. Bratitsis and A. Dimitracopoulou (2008). Interpretation Issues in Monitoring and Analyzing Group Interactions in Asynchronous Discussions. International Journal of e-Collaboration, IDEA Group Inc, 4(1), 20-40

[6] A. Capobianco and N. Carbonell (2006). Aides en ligne à l'utilisation de logiciels grand public : problèmes spécifiques de conception et solutions potentielles. Intellectica, 44, 87-120. Retrieved from the archive HAL: http://hal.archives-ouvertes.fr

[7] M. Cocea (2011). Disengagement Detection in Online Learning: Validation Studies and Perspectives. IEEE Transactions on Learning Technologies, 4, p. 114-124

[8] M. Cocea and S. Weibelzahl (2009). Log file analysis for disengagement detection in e-Learning environments. User Modeling & User-Adapted Interaction, 19(4), p. 341-385. doi:10.1007/s11257-009-9065-5

[9] D. Després and T. Coffinet (2004).Reflet, un miroir sur la formation. In proceedings of the Information Technologies and Communication in Education Conference.

[10] A. Dufresne, J. Basque, G. Paquette, M. Léonard, K. Lundgren-Cayrol and S. Prom Tep (2003). Vers un modèle générique d'assistance aux acteurs du téléapprentissage. Volume 10, 2003 Research Article, Science and Technologies of Information and Communication for Education and Training.

[11] W. Elm and D. Woods (1985) Getting lost: A case study in interface design. In proceedings of the human factors society 29th Annual Meeting, p. 927-931. 1985.

[12] P. Fastrez (2002). Navigation entailments as design principles for structuring hypermedia. In: Education, Communication and Information, Special Issue on Hypertext and Hypermedia, Vol.2, no.1, p.7-22 (2002). http://hdl.handle.net/2078.1/69317.

[13] M. Freed, J.G. Carbonell, G.J. Gordon, J. Hayes, B.A. Myers, D.P. Siewiorek, S.Smith, A. Steinfeld and A. Tomasic (2008). RADAR: A Personal Assistant that Learns to Reduce Email Overload. In AAAI 2008 proceedings. pp.1287-1293.

[14] D. Gaonac'h (2005). Les différentes fonctions de la mémoire dans l'apprentissage des langues étrangères. In the Plenary conference seventeenth Congress of APLIUT, IUT of South Toulon (Var), France.

[15] N. Hawes (2009). Architectures by Design: The Iterative Development of an Integrated. Intelligent Agent. Proceedings of AI-2009, Cambridge, UK. pp. 349-362.

[16] J-M. Heraud and A. Mille (2003). Pixed: assister l'apprentissage à distance par la réutilisation de l'expérience. Proceedings of Workshop Case-Based Reasoning Platform AFIA 03 Laval.

[17] P. Jarušek and R. Pelánek (2012) Analysis of a simple model of problem solving times. 11th International Conference on Intelligent Tutoring Systems, ITS 2012: Vol. 7315 LNCS (pp. 379-388). Chania, Crete.

[18] S-H. Kasdali (2013) Le soutien : entre situation d'apprentissage synchrone et situation asynchrone, quelle différence chez l'apprenant en ligne ? Congress on Current Research in Education and Training (AREF), Montpellier, August 2013.

[19] M. Khatraoui, N. Bousbia and A. Balla (2008). Détection de similarité sémantique entre pages visitées durant une session d'apprentissage. Workshop on Semantic similarity measures (EGC'08). 8th Day of Francophone Mining and Knowledge Management, pp.121—129.

[20] H. Lieberman (2001). Interfaces that Give and Take Advice. Carroll, J. (Ed). Human-Computer Interaction for the New Millenium, ACM Press/Addison-Wesley, pp. 475-485.

[21] F. Lemieux, M-C. Desmarais and P-N. Robillard (2013) Analyse chronologique des traces journalisées d'un guide d'étude pour apprentissage autonome. STICEF Review, Volume 20, 2013, ISSN: 1764 -7223.

[22] D. Lin (1998). An information-theoretic definition of similarity. The 15th International Conference on Machine Learning, San Francisco, CA.

[23] W. J. Linden van der, D. J. SCRAMS and D. L. SCHNIPKE (1999). Using Response-Time Constraints to Control for Differential Speededness in Computerized Adaptive Testing. Applied Psychological Measurement, 23(3), p.195-210.

[24] G-C. Loghin, T. Carron and J-C. Marty (2007).Apporter de la flexibilité dans l'observation d'une activité pédagogique. In proceedings of Interactive Learning Environments conference, page 83-94, NPRI, Lausanne, Switzerland.

[25] C. Loisy and Ch. Péllissier (2012). Des aides pour une consigne ouverte : assistants cognitifs dans Pairform@ance. International Journal of Technologies in Higher Education, IJTHE. volume: 9. Issue : 3. p.43-54

[26] M .May, S. George and P. Prévôt (2008).Tracer, analyser et visualiser les activités de communications médiatisées des apprenants. In JOCAIR Symposium, Amiens : France.

[27] F. Noury, N. Huet, C. Escribe, J-C. Sakdavong, and O. Catteau, (2007). Buts d'accomplissement de soi et jugement métacognitif des aides en EIAH. In proceedings of Interactive Learning Environments conference, ISBN 978-2-7342-1088-7, p. 293-298, June 2007, Lausanne.

[28] C. Pélissier and S. Mailles-Viard Metz (2010). Deviating technologies to design personal and creative help in elearning .Procedia - Social and Behavioral Sciences, 2, 2, 3552-3557, Elsevier Publication, 2010.

[29] C. Pélissier, and Qotb, H. (2012). Web social et l'apprentissage des langues : spécificités et rôles de l'utilisateur « Les multiples rôles de l'utilisateur dans les apprentissages mutuels en ligne», ALSIC review, january 2012.

[30] M. Perraudeau (2005). Les difficultés ordinaires d'apprentissage. Dossier: Helping Students? No. 436 educational books.

[31] M. Puustinen (2010). La demande d'aide de l'apprenant dans différents types de situations d'apprentissage ; l'autorégulation. HDR, Poitiers, November 2010.

[32] J-J. Quintin (2008). Accompagnement tutoral d'une formation collective via Internet. Analyse des effets de cinq modalités d'intervention tutorale sur l'apprentissage en groupe restreints. Phd thesis, University of Mons-Hainaut, 2008.

[33] B. Richard (2008). Une approche épiphyte pour la conception de systèmes conseillers. Phd thesis, University of Maine, Paris, France.

[34] Ch. Rodrigues (2012). L'aide à l'apprentissage du vocabulaire à distance : effets des outils de la CMO. International Journal of Technologies in Higher Education, IJTHE. Vol.9, Issue 3; p. 25-42.

[35] A-M. Ryan and P-R. Pintrich (1997). Should I Ask for Help? The Role of Motivation and Attitudes in Adolescents Help Seeking in Math Class. Journal of Educational Psychology, Vol. 89, n°2, p.1-13, 1997.

[36] K. Sehaba and M. Metz (2011). Using interaction traces for evolutionary design support-Application on the Virtual Campus VCIel. International Conference on Computer Supported Education. Noordwijkerhout, Netherlands.

[37] K. Sahaba (2012). Système d'aide adaptatif à base de traces, International Journal of Technologies in Higher Education, 9(3) www.ijthe.org RITPU, IJTHE.

[38] R. Taraban, K. Rynearson and K. Stalcup (2001). Time as a variable in learning on the World-Wide

Web. Behavior Research Methods, 33(2), p. 217-225.doi:10.3758/bf03195368

[39] R. Tariba (2013). Analyse des traces numériques dans une perspective de détection automatique des types de motivation et de styles pédagogiques. Thesis of Master 2 EFE-2I2N-FEN.

[40] J. J.Thompson, T. Yang and S. W. Chauvin (2009). Pace: An Alternative Measure of Student Question Response Time. Applied Measurement in Education, 22(3), p. 272-289. doi:10.1080/08957340902984067.

[41] R. Viau (2002). La motivation des élèves en difficulté d'apprentissage : une problématique particulière pour des modes d'intervention adaptés. Conference organized by the Department for Coordination of Research in Pedagogical and Technological Innovation. Available on http://sites.estvideo.net/gfritsch/doc/rezo-cfa-408.htm accessed July 2015.

[42] R. Viau (2005). 12 questions sur l'état de la recherche scientifique sur l'impact des TIC sur la motivation à apprendre. University of Sherbrooke. Available at [Http://tecfa.unige.ch/tecfa/teaching/LME/lombard/motivation/viau-motivation-tic.html] Accessed July 2015.

[43] T. Wang and B. A. Hanson (2005). Development and Calibration of an Item Response Model That Incorporates Response Time. Applied Psychological Measurement, 29(5), p. 323-339. doi:10.1177/0146621605275984

[44] S. L. Wise and X. Kong, (2005). Response Time Effort: A New Measure of Examinee Motivation in Computer-Based Tests. Applied Measurement in Education, 18(2), p. 163-183. Doi: 10.1207/s15324818ame1802_2.

Learning Sentiment Dependent Bayesian Network Classifier for Online Product Reviews

Sylvester Olubolu Orimaye, Zi Yang Pang and Alvino Mandala Putra Setiawan
School of Information Technology, Monash University Malaysia
E-mail: sylvester.orimaye@monash.edu

Keywords: sentiment classification, sentiment-dependent, Bayesian network, product reviews

Received: September 17, 2015

Analyzing sentiments for polarity classification has recently gained attention in the literature with different machine learning techniques performing moderately. The challenge is that sentiment-dependent information from multiple sources are not considered often in existing sentiment classification techniques. In this study, we propose a logical approach that maximizes the true sentiment class probabilities of the popular Bayesian Network for a more effective sentiment classification task using the individual word sentiment scores from SentiWordNet. We emphasize on creating dependency networks with quality variables by using a sentiment-dependent scoring technique that penalizes the existing Bayesian Network scoring functions such as K2, BDeu, Entropy, AIC and MDL. The outcome of this technique is called Sentiment Dependent Bayesian Network. Empirical results on eight product review datasets from different domains suggest that a sentiment-dependent scoring mechanism for Bayesian Network classifier could improve the accuracy of sentiment classification by 2% and achieve up to 86.7% accuracy on specific domains.

Povzetek: Razvit je nov Bayesov klasifikator na osnovi mnenjske odvisnosti, uporabljen za ocenjevanje spletnih produktov.

1 Introduction

Sentiment Classification (SC) has recently gained a lot of attention in the research community [1, 2, 3]. More recently, SC has moved from the commonly used bag-of-words models to bag-of-concepts models [4, 5]. This is due to its increasing demand for the analysis of consumer sentiments on products, topic and news related text from social media such as Twitter[1] and online product reviews such as Amazon[2]. In the same manner, Bayesian Network (BN)[6] also known as Bayesian Belief Network plays a major role in Machine Learning (ML) research for solving classification problems. Over the last decade, learning BNs has become an increasingly active area of ML research where the goal is to learn a network structure using dependence or independence information between set of variables [6, 7, 8, 9]. The resulting network is a directed acyclic graph (DAG), with a set of joint probability distributions, where each variable of the network is a node in the graph and the arcs between the nodes represent the probability distribution that signifies the level of dependency between the nodes.

While it is more common to use other ML algorithms for SC tasks [10, 11], few research papers have proposed BN as a competitive alternative to other popular ML algorithms. Considering the huge size of data available from social media and the level of difficulty attached with analysing sentiments from natural language texts, the ability of BN to learn dependencies between words and their corresponding sentiment classes, could undoubtedly produce a better classifier for the sentiment classification task. This paper focusses on constructing a BN classifier that uses sentiment information as one important factor for determining dependency between network variables.

BN has been successfully applied to solve different ML problems with its performance outweighing some of the popular ML algorithms. For example, in [12], a full Bayesian Network classifier (FBC) showed statistically significant improvement on state-of-the-art ML algorithms with the 33 UCI datasets. In the case of SC, Naïve Bayes (NB), which is a special case of BN [13], and one of the leading ML algorithms for SC tasks [10], has surprisingly and repeatedly shown improved performance on movie and product reviews despite its conditional independence assumption. By comparative study, we show that a Sentiment Dependent Bayesian Network (SDBN) classifier has improved performance on popular review datasets such as Amazon product reviews due to the ability of the Bayesian Network to construct a network structure of multiple dependencies [12].

Constructing a BN classifier requires learning a network structure with set of Conditional Probability Tables (CPTs) [6]. Basically, there are two combined steps involved in the BN construction process. The first is to perform variable search on a search space, and the other is to score each variable based on the degree of fitness [14]. The chal-

[1] https://twitter.com/
[2] https://amazon.com/

lenge however, is to ensure that good networks are learned with appropriate parameters using a scoring or fitness function to determine network variables from the given dataset. Thus, much of the research works on BN focus on developing scoring functions for the BN classifier [15]. We argue that such scoring functions rely on many assumptions that make them less effective for SC tasks. For example, K2 algorithm, which is based on Bayesian Scoring function relies on the assumptions of parameter independence and assigning a uniform prior distribution to the parameters, given the class [9]. We believe these assumptions lead to many false positives in the classification results as sentiment classes are better captured by conditional dependency between words, rather than independent word counts [16, 17].

We also suggest that *varying* prior distribution could be assigned to each variable since each word has a natural *prior* probability of belonging to a particular sentiment class, independent of the data. For example, the word "good" is naturally positive and "bad" is naturally negative. Thus, in this work, we propose a sentiment scoring function that leverage sentiment information between variables in the given data. The output of the sentiment scoring function is then used to augment existing BN scoring functions for better performance. Our aim is to ensure sentiment information form part of the fitness criteria for selecting network variables from sentiment-oriented datasets such as reviews.

The proposed scoring function uses a multi-class approach to compute the conditional mutual information using sentiment-dependent information between local variables in each class of instances. The conditional mutual information for all classes are then penalized using the Minimum Description Length (MDL) principle. The local probabilities used in computing the conditional mutual information is computed using the popular Bayesian probability that uses the prior probability of a variable belonging to a natural sentiment class (i.e. independent of the given data by using individual word sentiment score from SentiWordNet [18]) and the observation of the variable in the selected class of instances and other classes in the dataset (e.g. positive and negative). For example, the class probability of each word in a product review is augmented with the polarity probability of the same word from SentiWordNet. The technique takes into account that the network structure would depend on the following criteria:

- The posterior probability from multiple evidences that variables x_i and x_j have sentiment dependency;
- The conditional mutual information between the variables for all sentiment classes;
- The dependency threshold computed using the Minimum Description Length principle; and
- The representation of the network as a full Bayesian Network as proposed in [12].

The importance of the first criterion is that we are able to avoid the *independence assumption* made by the existing BN scoring functions. We capture local sentiment dependency between the variables as a joint probability of evidences from each variable and each class in the given data. Also, existing BN scoring functions uses the conditional *independence* given the data as a whole for determining dependencies between variables [15, 9]. Under such approach in SC, two co-occurring words may occur with the same or similar frequencies in different classes. We observed that training BN classifier without penalizing such occurrences or dependencies, could affect the classifier decision to decide an appropriate sentiment class. As such, our second criterion captures the conditional mutual information between the variables, while the third criterion ensures that a BN classifier uses quality variables that are above the computed threshold. The latter also allow us to enforce strict *d-separation* policy between the network variables, which formally defines the process of determining independence between variables [19]. Thus, only quality variables are used to form the dependency network for the BN classifier. Finally, we introduce the last criterion as an improvement over a network constructed based on the first three criteria. The full Bayesian Network technique ensures that an independent sentiment dependent network is constructed for each sentiment category (i.e. negative and positive).

Section 2 of this paper discusses related work and additional motivations. In Section 3, we explain the problem background and then present the proposed sentiment-dependent technique in Section 4. Our experiment is described in Section 5. Finally, Section 6 gives the conclusion to our study and some thoughts on future research directions.

2 Related work

2.1 Sentiment classification (SC)

The most prominent of SC work is perhaps [20] which used supervised machine learning techniques for the polarity classification of *positive* and *negative* sentiments in movie reviews. As a result of that work, different research directions within the field of sentiment analysis and opinion mining have been actively pursued [2, 3, 4, 5].

[10] proposed a subjectivity summarization technique, which uses minimum cuts to classify sentiment polarities in movie reviews. The technique identifies and extracts subjective portions of review documents using minimum cuts in graphs. The minimum cut approach takes into consideration, the pairwise proximity information supplied through graph cuts that partition sentences which are likely to be in the same sentiment class. This approach gave better improvement from 82.8% to 86.4% on the subjective portion of review documents. The approach also gave similar better improvement when only 60% portion of a product review document is used compared to an entire review. In our work, we propose a classification technique that uses

the entire portion of each product review.

[21] performed classification of approximately 200K product reviews by using different machine learning algorithms. More importantly, the work investigated the significance of higher order n-gram language model ($n \geq 3$) in classifying sentiments from product reviews with an F1 of 90%. While Cui's work was performed on random websites for online products with limited product domains, we performed experiments on Amazon product reviews with 8 different product domains.

Similarly, [22] performed experiment with higher order n-gram features to train a Neural Network model on Amazon and TripAdvisor datasets with the best average error rates of 7.12 and 7.37, respectively. In contrast, our work investigate the performance of a sentiment-dependent Bayesian Network classifier on the Amazon datasets with different product domains.

More recently, [23] proposed a concept-level sentiment analysis technique, which uses the knowledge-based sentic computing technique that has recently gained attention within the sentiment analysis domain [3, 4, 5]. Because the sentic computing knowledge base sometimes omits vital sentiment discourse, [23] combines low-level linguistic features with the sentic computing technique to train machine learning model for polarity detection. In our work, the only knowledge base employed is SentiWordNet [24], which computes the natural polarity values of words rather than concept. Thus, we captured the dependencies between words by constructing and learning a Bayesian Network for sentiment classification.

A detailed review of other recent sentiment classification techniques on different datasets can be found in [1, 2, 3, 4, 5].

2.2 BN classifiers for sentiment classification

[16] and [17] proposed a two-stage Markov Blanket Classifier (MBC) approach to extract sentiments from unstructured text such as movie reviews by using BN. The approach learns conditional dependencies between variables (words) in a network and finds the portion of the network that falls within the *Markov Blanket*. The *Tabu Search* algorithm [25], is then used to further prune the resulting Markov Blanket network for higher cross-validated accuracy. Although Markov Blanket has shown to be effective in avoiding *over-fitting* in BN classifiers [7], the MBC approach finds sentiment dependencies based on the ordinary *presence* or *absence* of words in their original sentiment class only. We identify sentiment dependencies by considering multiple sources of evidence. These include multiple sentiment classes in the data and the *natural* sentiment class of each variable which is independent of its sentiment class in the given data.

Similarly, [26] proposed a parallel BN learning algorithm using MapReduce for the purpose of capturing sentiments from unstructured text. The technique experimented on large scale blog data and captures dependencies among words using mutual information or entropy, with the hope of finding a vocabulary that could extract sentiments. The technique differs from [17] by using a three-phase (drafting, thickening and thinning) dependency search technique that was proposed in [27]. Other than using *mutual information* in the *drafting* phase of the search technique, the work did not capture additional sentiment dependencies using other source of evidence.

Again, we do not focus on developing a search algorithm but a scoring technique that considers multiple sentiment-dependent information as part of the existing state-of-the-art scoring functions.

3 Problem background

3.1 Bayesian network (BN)

A Bayesian Network N is represented as a graphical distribution of the joint probability between a set of random variables [28]. The network has two components: (1) a DAG $G = (R_n, M_r)$ that represents the structural arrangement of a set of variables (nodes) $R_n = \{x_1, ..., x_n\}$ and a corresponding set of dependence and independence assertions (arcs) M_r between the variables; (2) a set of conditional probability distributions $P = \{p_i, ..., p_n\}$ between the parent and the child nodes in the graph.

In the DAG component, the existence of an directed arc between a pair of variables x_i and x_j asserts a conditional dependency between the two variables [8]. The directed arc can also be seen to represent *causality* between one variable and the other [29], that is, variable x_y is an existential cause of variable x_z, hence $x_y \rightarrow x_z$. The absence of an directed arc between a pair of variables, however, represents a conditional independence, such that, given a subset U of variables from R_n, the degree of information about variable x_i does not change by knowing x_j, thus $I(x_i, x_j|\mathbb{U})$. This also implies that $p(x_i|x_j, U) = p(x_i|U)$. The parent(s) of variable $x_i \in R_n$ is denoted by a set $pa_G(x_i) = x_j \in R_n|x_j \rightarrow X_i \in M_r$, and $pa_G(x_i) = \emptyset$ for the root node.

The conditional probability distributions of the DAG G is represented by its CPT, which contains a set of numerical parameters for each variable $x_i \in R_n$. These numeric parameters are computed as the probability of each variable given the set of parents, $p(x_i|pa_G(X_i))$. Over the set of variables in R_n, the joint probability for the BN is therefore obtained as follows:

$$p(x_1, ..., x_n) = \prod_{X_i \in R_n} p(x_i|pa_G(x_i)) \qquad (1)$$

Thus, for a typical classification task, the BN classifier would learn the numerical parameters of a CPT from the DAG structure G, by estimating some statistical information from the given data. Such information include, *mutual information* (MI) between the variables and *chi-square distribution* [15]. The former is based on the *local score* *metrics* approach and the latter exhibits *conditional independence tests* (CI) approach. For both approaches, different *search* algorithms are used to identify the network structure. The goal is to ascertain, according to one or more search criteria, the best BN that fits the given data by evaluating the weight of the arc between the variables. The criteria for evaluating the fitness of the nodes (variables), and the arcs (parameters) in the BN search algorithms, are expressed as fitting or scoring functions within the BN classifier [15]. Our goal is to ensure that those criteria include *sentiment-dependent* information between the variables. We will focus on penalizing existing *local score metrics* with our sentiment-dependent scoring function for the BN classifiers, hence the SDBN proposed in this paper.

The local score metrics are of particular interest to our sentiment classification task because they exhibit a practical characteristic that ensures the joint probability of the BN is *decomposable* to the sum (or product) of the individual probability of each node [28][15]. To the best of our knowledge, very few research papers have considered sentiment-dependent information, as part of the fitness criteria for capturing dependency between the variables, especially for product reviews on different domains.

3.2 BN scoring functions

We focus on the local score metrics functions, K2, BDeu, Entropy, AIC and MDL [15]. The functions define a fitness score, and a specified search algorithm searches for the best network that maximizes the score. Each of these functions identifies frequencies of occurrence of each variable x_i in the data D and a network structure N. Although the performance of these scoring functions may vary on different datasets [15], in this paper, we assume that the scores generated by the scoring functions are somehow naïve, thus, we attempt to mitigate its effect on SC tasks. First, we will define the parameters that are common to all the functions. We will then describe each of the functions with their associated formula and specify their limitations to the SC tasks.

Similar to [30], we use $r_i(1 \leq i \leq n)$ to denote the size or cardinality of x_i. $pa(x_i)$ represents the parents of x_i and the cardinality of the parent set is represented by $q_i = \prod_{x_j \in pa(x_i)} r_j$. If $pa(x_i)$ is empty (i.e. $pa(x_i) = \emptyset$), then $q_i = 1$. The number of instances in a dataset D, where $pa(x_i)$ gets its jth value is represented by $N_{ij}(1 \leq i \leq n, 1 \leq j \leq q_i)$. Similarly, $N_{ijk}(1 \leq i \leq n, 1 \leq j \leq q_i, 1 \leq k \leq r_i)$ represents the portion of D where $pa(x_i)$ gets its jth value and x_i gets its kth value such that $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. Obviously, N represents the size of D.

K2: This metric is a type of Bayesian scoring function proposed by [6]. The function relies on series of assumptions such as parameter independence and uniform prior probability for the network. We reiterate that instead of independent word counts, the sentiments expressed in a given data are better captured using conditional dependency between words and their related sentiment classes [16]. The K2 metric is defined as follows:

$$S_{k2}(N,D) = P(N) \prod_{i=0}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(r_i - 1 + N_{ij})!} \prod_{k=1}^{r_i} N_{ijk}! \tag{2}$$

BDeu: The metric was proposed by [31] as a generalization of K2. It resulted from Bayesian Drichlet (BD) and BDe which were proposed by [32]. The BD is based on hyperparameters η_{ijk} and the BDe is a result of BD with additional assumptions. BDeu relies on the sample size η as the single parameter. Since BDeu is a generalization of K2, it carries some of our concerns expressed on K2 earlier. Most importantly, the uniform prior probability assigned to each variable $x_i \in pa(x_i)$ could be replaced by the probability of the variable belonging to a *natural* sentiment class as stated earlier. We suggest that this is likely to increase the accuracy of the sentiment classifier especially on sparse data distribution. We define the BDeu metric as follows:

$$S_{BDeu}(N,D) = P(N) \prod_{i=0}^{n} \prod_{j=1}^{q_i} \frac{(\Gamma(\frac{\eta}{q_i})}{\Gamma(N_{ij} + \frac{\eta}{q_i})} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + \frac{\eta}{r_i q_i)}}{\Gamma(\frac{\eta}{r_i q_i})} \tag{3}$$

Note that the function $\Gamma(.)$ is inherited from BD, and $\Gamma(c) = \int_0^\infty e^{-u} u^{c-1} du$ [15].

Entropy: Entropy metric measures the distance between the joint probability distributions of the network [15]. This allows dependency information to be identified by computing the mutual information (or entropy) between pair of variables. Thus, a minimized entropy between a pair of variables denotes dependency relationship, otherwise, a large entropy implies conditional independence between the variables [12][32]. While the entropy metric has been successful in measuring dependency information for BN classifiers, the local probabilities involved in the metric is largely computed based on *conditional independence* assumption given the data (i.e. using frequency counts for independent variables). We suggest that a joint probability of multiple sentiment evidences could improve the metric in BN classifiers for the SC tasks. The metric is defined as follows:

$$H(N,D) = -N \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \frac{N_{ijk}}{N} \log \frac{N_{ijk}}{N_{ij}} \tag{4}$$

AIC: The AIC metric adds a non-negative parameter penalization to the entropy method [15], which could also be improved by multiple sentiment evidences as in the case of the entropy method. The metric is specified as follows:

$$S_{AIC}(N,D) = H(N,D) + K \tag{5}$$

Where K is the number of parameters, such that $K = \sum_{i=1}^{n}(r_i - 1).q_i$.

MDL: The MDL metric is based on the minimum description length principle which selects a minimum representative portion of the network variables through coding [15]. The best BN is identified to minimize the sum of the description length for the data. The metric is defined as follows:

$$S_{MDL}(N, D) = H(N, D) + \frac{K}{2}\log N \qquad (6)$$

The use of MDL has not been investigated for sentiment classification on its own except for selecting dependency threshold between variables in BN. The study in [28], suggests that the mean of the total cross-entropy error is asymptotically proportional to $\frac{\log N}{2N}$, which is why the entropy metric is penalized in Equation 6.

In this paper, the proposed sentiment-dependent score function is based on the Information Theory approach. The approach uses the entropy-based conditional mutual information (CMI) technique to measure the dependencies between the variables. The local probabilities for computing the CMI between two variables are derived as joint probability resulting from multiple evidences of both variables belonging to the same sentiment class. This is achieved by using a multiclass approach that measures the CMI in each sentiment class. The sum of the CMIs over the data is thereafter penalized using the MDL principle as suggested in [28].

4 Sentiment-dependent BN

As emphasized earlier, our motivation is to include sentiment information as part of the dependency criteria between the network variables. Similar to [12], we constructed a multi-class Full Bayesian Network and encode the sentiment information within the CMIs that determine the dependencies within the network.

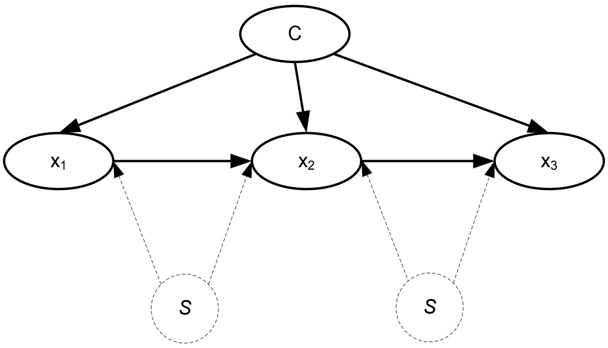


Figure 1: Structural overview of SDBN.

Figure 1 shows the structure of a SDBN. In this figure, C denotes the class node and the parent to all the variable nodes x_1, x_2 and x_3. The edges between variable x_1 and x_2 and x_3 represents dependencies between the variables. The dashed directed lines from each sentiment component S show the contribution of the sentiment information to the dependencies.

The proposed SDBN is created from a sentiment dependent score table (SDST) similar to the conventional CPT which contains network parameters from the data. Given a dataset D containing two or more sentiment classes, we divide D into c subsets, where $D_1 ... D_c$ represent the sentiment classes which are present in D. Thus, for each subset D_c, we create a SDST from the given data, and at the later stage, we use the values in SDST to learn a full Bayesian classifier.

Creating an appropriate CPT or SDST is challenging, especially when there is a sheer number of variables in the given data [27]. In fact, local search algorithms such as *K2*, *Hill Climbing*, *TAN*, *Simulated annealing*, *Tabu search* and *Genetic search* have been developed to address this challenge [7]. Thus, we do not intend to repeat the sophisticated local search process in our scoring technique. We use a straight forward approach that computes CMI as the dependency between a pair of variables, given a subset D_c. The resulting scores for each pair of variables is stored into the SDST. Equation 7 computes the CMI for a pair of variables. Note that this process is equivalent to the *drafting* phase proposed in [27] or the Chow and Liu algorithm in [33]. We can therefore focus on computing the local probabilities $P(x_i)$ and $P(x_j)$ for the CMI. In this work, each local probability encodes the sentiment dependency information as a *joint probability* of multiple sentiment evidences. We suggest that the joint probability is better than using the ordinary variable presence or single frequency count.

$$\begin{aligned} CMI(x_i, x_j|C) = \sum_{x_i,x_j,c} P(x_i, x_j, c) \\ \log\frac{P(x_i, x_j, c)}{P(x_i|c), P(x_j|c)} \end{aligned} \qquad (7)$$

Alternatively, the CMI in Equation 7 can be computed with Equation 7 for penalizing the default CMI for a pair of variables, where $S_\lambda(x_i, x_j)$ is the sentiment *prior* computed from the multiple sentiment evidences for each variable x_i and x_j.

$$\begin{aligned} CMI(x_i, x_j|C) = \sum_{x_i,x_j,c} P(x_i, x_j, c) \\ \log\frac{P(x_i, x_j, c)}{P(x_i|c), P(x_j|c)} S_\lambda(x_i, x_j) \end{aligned} \qquad (8)$$

4.1 Local probabilities for CMI

In order to compute the local probabilities $P(x_i)$ and $P(x_j)$, we adopt Bayesian probability [34], to calculate the joint probability from multiple sentiment evidences. Bayesian probability encodes a *generative model* or *likelihood* $p(D|\theta)$ of the dataset with a *prior* belief $p(\theta)$ to infer a *posterior* distribution $p(\theta|D)$, see Equation 9. The idea is to determine a favourable posterior information of a particular variable belonging to its observed class, such that, the conditional mutual information between two dependent variables x_i and x_j increases when the posterior information for both variables in the same class is large.

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \tag{9}$$

However, in sentiment oriented documents such as product reviews, it is very common to observe variables that belong to different classes in one sentiment class. [20] referred to such scenario as *thwarted expectation.* For example, a “positive" review document may contain certain “negative" words used to express dissatisfaction about an aspect of a product despite some level of satisfaction that the product might offer. With this kind of problem, it is much probable that a dependency network that is learned with ordinary frequency counts of each variable (regardless of the sentiment class) would no doubt leads to inaccurate sentiment classifiers.

Figure 2 shows a sample BN resulting from a product review dataset upon performing attribute selection. In that network, variable “After" has a 1.0 probability of belonging to the *negative* and *positive* classes, respectively. Similarly, variable “not" has a 0.723 probability of belonging to a “positive" class rather than “negative". Every other variables in the network, has split probabilities between both classes. Our aim is to remove such variables from the dependency network or at least minimize its influence in the network such that the quality of the network is improved for sentiment classification.

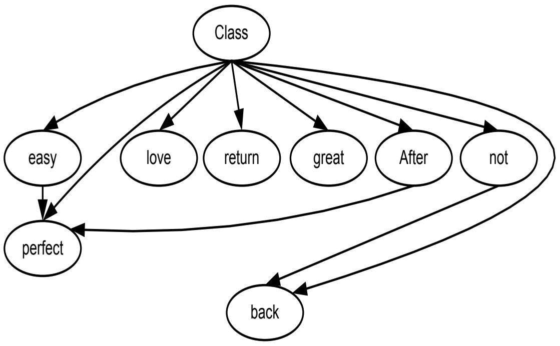


Figure 2: An example Bayesian network from product reviews.

In this work, we compute the posterior information for each variable by considering its *prior* information and joint *likelihood* or *observation* from all the sentiment classes available in the data.

The *prior* information is computed using the *natural sentiment or polarity scores* from SentiWordNet [24]. SentiWordNet gives the polarity scores of corresponding synsets for each English word. However, the polarity scores are often different for each of the synset entries. A synset contains multiple semantic or polarity interpretation of a given word. Each interpretation has three different polarities values. That is, a synset entry (word) would have a *positive*, *negative*, and *neutral* polarity scores which varies depending on the semantic interpretation of the word. An example of such words is “great". Its fourth synset entry in SentiWordNet has 0.25 *positive*, 0.625 *negative*, and 0.125 *neutral* polarity scores, respectively.

In this work, we focus on the “positive" and “negative" sentiments, thus we will only consider positive and negative polarity scores from SentiWordNet. The challenge however, is to compute an absolute and single polarity score for each word from its multiple synset entries. First, we compute the score for each polarity independently and then find the polarity that maximizes the other. The score for the positive or negative polarity of all synset entries for a given word is computed as follows:

$$score_{\phi}(w) = \frac{1}{\epsilon}\sum_{i=1}^{\epsilon} E_c(e_i) \tag{10}$$

where $score_{\phi}(w)$ is the score for each polarity of the given word w, ϵ is the number of synset entries E for the word, c is the polarity or category (i.e. positive or negative) and e_i is each synset entry. Thus, the *prior* or *absolute polarity score* for w is computed as follows:

$$POL_{\phi}(w) = \arg\max_{c \in C} score_{\phi}(w) \tag{11}$$

where $POL_{\phi}(w)$ is the maximum polarity score computed with respect to either *positive* or *negative* category c from all the syset entries.

We compute the *likelihood* information using a multi-class approach. Given a set of sentiment classes C, the probability of a variable belonging to its “first" observed sentiment class, is calculated as a joint probability of independently observing the variable in its first observed sentiment class (i.e.negative and positive) and every other sentiment classes, $C_1 ... C_n$. Thus, the likelihood information is computed as follows:

$$p(x_1, ..., x_C|D) = \prod_{c=1}^{C} p(x_c|D) \tag{12}$$

Where $p(x_c|D)$ is the probability of a variable x belonging to a class c given the data D.

Given the data, our aim is to minimise the effect of the variables which might have appeared in a wrong (false positive) class as a result of *thwarted expectation* that was suggested in [20], thereby biasing the dependency structure. Common examples are *negation* and *objective* words such as *not* and *After* as illustrated with Figure 2. If the word “not" for example, has a probability of 0.723 in a first observed “positive" class and a probability of 0.496 in the other negative class, then its *likelihood* of actually belonging to the “positive" class would be 0.359. Note that each probability is independent in this case as both probabilities do not sum to 1.

In addition, the *prior* or *natural sentiment score* (see Equation 11) obtained from SentiWordNet regulates the *likelihood* further, ensuring that the probability of a variable belonging to its first observed class is also conditioned on the natural sentiment class of the word which is independent of the data. With variable *not* having a probability of 0.625 *negative* from SentiWordNet, the *posterior* Bayesian probability is 0.149. This means the probability of the variable belonging to the *negative* class is higher (i.e. 0.85), and thus, should not be allowed to have strong dependency on a "true positive" variable. We suggest that this technique is more reliable than using the highest probability from both classes at the expense of accuracy (e.g. using only 0.723 and without the *prior*).

Thus, using the Bayesian probability defined in Equation 9, we substitute the *likelihood* information $p(x_1, ..., x_C|D)$ to $p(D|\theta)$ and the *prior* information $POL_\phi(w)$ to $p(\theta)$. Note that $P(D)$ is the sum of the two independent probabilities used in the likelihood (i.e. 0.723 and 0.496).

4.2 Sentiment dependency score

Having computed the local probabilities $P(x_i)$ and $P(x_j)$ using the Bayesian probability approach, we compute the conditional mutual information as the dependency information between pair of variables in each class. Thus, we store the dependency information in the sentiment dependent score table, SDST. Again, the SDST is similar to the conventional CPT. The obvious difference is that sentiment information have been used to generate SDST. However, since we are using conditional mutual information to compute dependencies between variables, certain dependency threshold needs to be met in order to generate a reliable sentiment dependencies between each pair of variables in the SDST. As mentioned earlier, [28] suggested that the mean of the total cross-entropy (mutual information) error is asymptotically proportional to $\frac{\log N}{2N}$. Using that MDL principle, we defined the threshold value as follows:

$$\Theta_{x_i,x_j} = \frac{\log N_c}{2N_c} \tag{13}$$

where Θ_{x_i,x_j} is the sentiment dependency threshold between a pair of variables x_i and x_j, N_c is the size of the data for a particular training class. Note that we generated individual SDST for each sentiment class in our data. In this work, a pair of variables x_i and x_j have strong sentiment dependency and get stored into the appropriate SDST, if and only if, the conditional mutual information $CMI(X_i, X_j|C) > \Theta_{x_i,x_j}$. Otherwise, we store a zero value to the corresponding slot in the SDST. CMI values greater than zero in the SDST is then used to build the resulting sentiment-dependent network structure for the ML task. The process of generating SDST is shown in Algorithm 1.

Algorithm 4 SDST(D)

Input : A set of labelled instances D.
Output : A set of Sentiment Dependent Score Tables for all pairs of variables x_i and x_j.
Steps

```
Create a multi-class structure that partitions instances D into subsets of classes $D_c$.
$SDST_{1,...,C}$ = empty.
for each subset $D_c$ in D do
    Compute the local probabilities $P(x_i)$ and $P(x_j)$ with sentiment dependent information as in Equation 9.
    Use the local probabilities to compute CMI for each pair of variables $x_i$ and $x_j$ using Equation 7.
    Compute the MDL threshold $\Theta$ with Equation 13.
    if CMI > MDL threshold $\Theta_{x_i,x_j}$ then
        Store the CMI into $SDST_c$ columns $x_i, x_j$ and $x_j, x_i$, respectively.
    else
        Store 0 into $SDST_c$ columns $x_i, x_j$ and $x_j, x_i$, respectively.
    end if
end for
Return $SDST_{1,...,C}$ for a full Bayesian Network classifier
```

5 Experiments and results

We conducted set of experiments using the proposed SDBN algorithm on 8 different product review domains. We then compared the accuracy with a state-of-the-art sentiment classification technique.

5.1 Datasets and baselines

Our datasets consist of Amazon online product reviews [3] that were manually crawled by [35]. These include *Health*, *Kitchen*, *Software*, *Video*, *Books*, *DVD*, *Electronics*, and *Grocery* reviews. Each product domain consists of **1000** *positive* reviews and **1000** *negative* reviews, hence each domain has **2000** balanced set of instances. Note that 60% training and 40% testing sets were used on all domains.

As baseline, we implemented the popular sentiment classification technique in [10] as a traditional sentiment classification benchmark on product reviews. The baseline technique produced subjective portions of reviews from our datasets and were used with NB and the ordinary BN classifiers. Baseline-NB denotes the baseline using NB classifier and Baseline-BN represents the baseline with ordinary BN classifier on the subjective portions of reviews, respectively. We performed a grid search with 10-fold cross-validation for the three algorithms and observed that both SDBN and Baseline-BN gave best accuracies using SimpleEstimator with $\alpha = 0.5$ and K2 search algorithm with

[3] http://www.cs.jhu.edu/ mdredze/datasets/sentiment/

the Bayes/K2 scoring function, while NB performed better with the Kernel Estimator.

5.2 Data preparation

We implemented our algorithm within the *weka.classifiers.bayes* package of the WEKA[4] data mining framework. The SentiWordNet library[5] including the lexicon file were also incorporated into the same WEKA directory. Further, we prepared our datasets according to the WEKA's ARFF format by using the term frequency-inverse document frequency (TF-IDF) from the positive and negative reviews for each domain. Our implementation also uses the discretization algorithm within WEKA in order to reduce the classification error. This technique also correct the missing values within the training data.

5.3 Attribute selection

We evaluated the performance of the SDBN with reduced attribute sets since attribute selection tends to improve BN's accuracy [16]. Thus, we ranked and reduced the set of attributes for each of our dataset by using the "Attribute-Selection" filter in Weka. Specifically, we used the *InfoGainAttributeEval* evaluator with the *ranker* search algorithm to select from the top-10 ranked attributes to the top-1000 ranked attributes for each domain. Our experiment showed better result with the top-ranked 100 attributes.

5.4 Results

As emphasised in Table 1, we observed the proposed SDBN to have improved and sometimes comparable performance with the baseline classifiers. SDBN recorded better improvements on the Health and Kitchen domains. We also note that the accuracies on the Amazon video reviews seems to be lower than the accuracies that were reported on the IMDb video reviews by [10]. We suggest that this is a trade-off in sentiment classification on different datasets and/or domains as could be observed in our experiment on different Amazon domains. This could be investigated further in our future work. We believe that increased size of dataset, that is beyond the limited 1000 Amazon reviews, could further improve the accuracy of the SDBN classifier.

We also performed experiment using the SDBN for top-10, top-20, top-30, top-50, and top-100 attributes alone as shown in Table 2. The results showed that the performance of the SDBN increased steadily up to the best accuracy given by the top-100 attributes. We believe this is an indication that the performance of the SDBN could increase with much larger datasets. In addition, we compared between the performance of the top-10 to top-100 attributes for SDBN and the two baselines on the top three domains with better accuracy: Health, Kitchen, and Video. Figures 3, 4, 5,and 6 show consistent improvement with increasing attributes for the SDBN compared to the baselines.

Table 1: Accuracies of SDBN and baseline classifiers on Amazon product reviews.

Dataset	**SDBN**	**Baseline-BN**	**Baseline-NB**
Health	**80.2%**	78.9%	78.6%
Kitchen	**82.3%**	80.5%	81.8%
Software	78.7%	78.6%	78.7%
Video	**75.4%**	75.2%	74.9%
Books	**77.7%**	77.5%	77.3%
DVD	77.6%	77.6%	77.5%
Electronic	79.9%	79.8%	79.7%
Grocery	86.7%	86.7%	86.5%

Table 2: Accuracies of SDBN on Top-10 to Top-100 attributes.

Dataset	**Top-10**	**Top-20**	**Top-30**	**Top-50**	**Top-100**
Health	67.4%	70.7%	73.4%	76.9%	80.2%
Kitchen	71.4%	74.8%	77.2%	79.0%	82.3%
Software	69.7%	73.4%	75.4%	76.1%	78.7%
Video	63.5%	68.6%	72.3%	75.1%	75.4%
Books	68.4%	71.1%	73.6%	75.3%	77.7%
DVD	68.1%	72.3%	73.8%	75.7%	77.6%
Electronic	67.0%	70.1%	70.5%	75.8%	79.9%
Grocery	69.9%	75.9%	80.0%	83.0%	86.7%

As shown in Table 3, we also performed experiment by using SDBN with other scoring functions reported in Section 3.2 using the top-100 attributes, which gave better accuracy. Our observation shows that those scoring functions did not improve the result for SDBN beyond the Bayes/K2 scoring function used in the earlier experiments. This is consistent with the comparative study conducted in [15] on BN scoring functions. Overall, we have observed the SDBN classifier to have reasonable performance that shows a promising research pathway for using Bayesian Network as a competitive alternative classifier for sentiment classification tasks.

[4] http://www.cs.waikato.ac.nz/ml/weka/

[5] http://sentiwordnet.isti.cnr.it/download.php

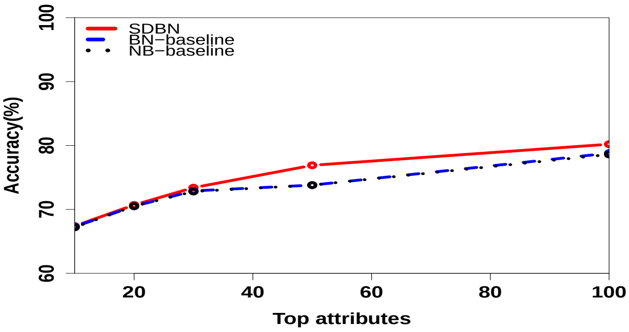


Figure 3: SDBN vs. baselines for top-10 to top-100 attributes on Health domain.

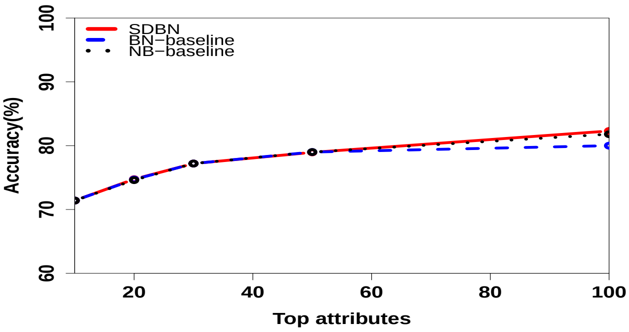


Figure 4: SDBN vs. baselines for top-10 to top-100 attributes on Kitchen domain.

Table 3: Accuracies of SDBN with different scoring functions on Amazon product reviews.

Dataset	K2	BDeu	Entropy	AIC	MDL
Health	80.2%	76.3%	76.3%	76.1%	76.1%
Kitchen	82.3%	75.3%	73.4%	75.2%	75.2%
Software	78.7%	73.1%	73.1%	73.1%	73.1%
Video	75.4%	73.5%	74.4%	72.8%	73.5%
Books	77.7%	70.9%	70.1%	71.1%	71.1%
DVD	77.6%	71.8%	70.0%	71.7%	71.7%
Electronic	79.9%	77.0%	76.2%	76.2%	77.2%
Grocery	86.7%	79.2%	80.0%	80.2%	79.3%

There are some limitations in the use of SentiWordNet though. For example, because there are many domain specific or technical terms (e.g. brand names) that were used in the product reviews, sentiment priors of those terms returned zero (i.e. neutral) as they are neither negative nor positive. This might have affected the sentiment dependencies within the network structure.

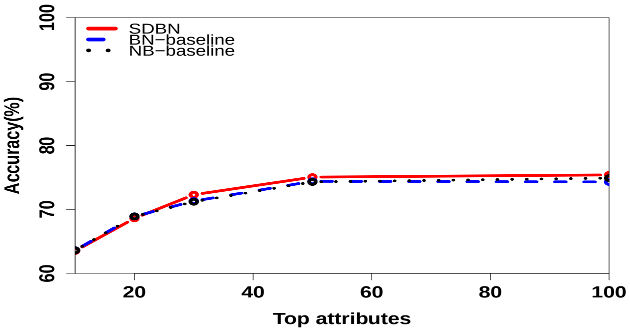


Figure 5: SDBN vs. baselines for top-10 to top-100 attributes on Video domain.

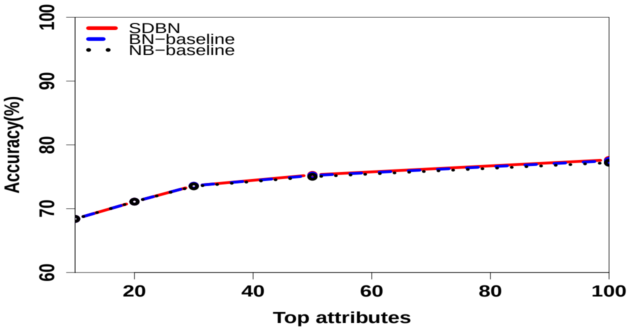


Figure 6: SDBN vs. baselines for top-10 to top-100 attributes on Books domain.

6 Conclusion

In this study, we have proposed a sentiment-dependent Bayesian network (SDBN) classifier. The proposed SDBN uses a multi-class approach to compute sentiment dependencies between pairs of variables by using a joint probability from different sentiment evidences. Thus, we calculated a sentiment dependency score that penalizes existing BN scoring functions and derived sentiment dependency network structure using the conditional mutual information between each pair of variables in a dataset. We performed sentiment classification on eight different Amazon product domains with the resulting network structure. Experimental results show that the proposed SDBN has comparable, and in some cases, improved accuracy than the state-of-the-art sentiment classifiers. In the future, we will experiment with SDBN on large scale Amazon SNAP datasets and the Hadoop platform.

Acknowledgement

The authors would like to thank the anonymous reviewers for the constructive comments.

References

[1] H. Tang, S. Tan, and X. Cheng, “A survey on sentiment detection of reviews,” *Expert Systems with Applications*, vol. 36, no. 7, pp. 10 760–10 773, 2009.

[2] B. Liu, “Sentiment analysis and opinion mining,” *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[3] E. Cambria and A. Hussain, *Sentic computing: Techniques, tools, and applications*. Springer Science & Business Media, 2012, vol. 2.

[4] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, “New avenues in opinion mining and sentiment analysis,” *IEEE Intelligent Systems*, no. 2, pp. 15–21, 2013.

[5] E. Cambria, D. Olsher, and D. Rajagopal, “Senticnet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis,” in *Twenty-eighth AAAI conference on artificial intelligence*, 2014, pp. 1515–1521.

[6] G. F. Cooper and E. Herskovits, “A bayesian method for the induction of probabilistic networks from data,” *Machine learning*, vol. 9, no. 4, pp. 309–347, 1992.

[7] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian network classifiers,” *Machine Learning*, vol. 29, pp. 131–163, 1997, 10.1023/A:1007465528199. [Online]. Available: http://dx.doi.org/10.1023/A:1007465528199

[8] J. Cheng and R. Greiner, “Learning bayesian belief network classifiers: Algorithms and system,” in *Advances in Artificial Intelligence*. Springer, 2001, pp. 141–151.

[9] X.-W. Chen, G. Anantha, and X. Lin, “Improving bayesian network structure learning with mutual information-based node ordering in the k2 algorithm,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 20, no. 5, pp. 628–640, 2008.

[10] B. Pang and L. Lee, “A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts,” in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Barcelona, Spain: Association for Computational Linguistics, 2004, p. 271.

[11] E. Boiy and M.-F. Moens, “A machine learning approach to sentiment analysis in multilingual web texts,” *Information Retrieval*, vol. 12, no. 5, pp. 526–558, 2009.

[12] J. Su and H. Zhang, “Full bayesian network classifiers,” in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 897–904.

[13] J. Cheng and R. Greiner, “Comparing bayesian network classifiers,” in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 101–108.

[14] D. Heckerman, *A tutorial on learning with Bayesian networks*. Springer, 2008.

[15] L. M. De Campos, “A scoring function for learning bayesian networks based on mutual information and conditional independence tests,” *The Journal of Machine Learning Research*, vol. 7, pp. 2149–2187, 2006.

[16] E. Airoldi, X. Bai, and R. Padman, “Markov blankets and meta-heuristics search: Sentiment extraction from unstructured texts,” in *Advances in Web Mining and Web Usage Analysis*. Springer, 2006, pp. 167–187.

[17] X. Bai, “Predicting consumer sentiments from online text,” *Decision Support Systems*, vol. 50, no. 4, pp. 732–742, 2011.

[18] A. Esuli, “Automatic generation of lexical resources for opinion mining: models, algorithms and applications,” *SIGIR Forum*, vol. 42, no. 2, pp. 105–106, 2008.

[19] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[20] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: sentiment classification using machine learning techniques,” in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.

[21] H. Cui, V. Mittal, and M. Datar, “Comparative experiments on sentiment classification for online product reviews,” American Association for Artificial Intelligence (AAAI), 2006.

[22] D. Bespalov, B. Bai, Y. Qi, and A. Shokoufandeh, “Sentiment classification based on supervised latent n-gram analysis,” in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 375–382.

[23] S. Poria, E. Cambria, G. Winterstein, and G.-B. Huang, “Sentic patterns: Dependency-based rules for concept-level sentiment analysis,” *Knowledge-Based Systems*, vol. 69, pp. 45–63, 2014.

[24] A. Esuli and F. Sebastiani, “Sentiwordnet: A publicly available lexical resource for opinion mining,” *Proceedings of LREC*, 2006.

[25] F. Glover, M. Laguna *et al.*, *Tabu search*. Springer, 1997, vol. 22.

[26] W. Chen, L. Zong, W. Huang, G. Ou, Y. Wang, and D. Yang, "An empirical study of massively parallel bayesian networks learning for sentiment extraction from unstructured text," in *Web Technologies and Applications*. Springer, 2011, pp. 424–435.

[27] J. Cheng, D. A. Bell, and W. Liu, "Learning belief networks from data: An information theory based approach," in *Proceedings of the sixth international conference on Information and knowledge management*. ACM, 1997, pp. 325–331.

[28] N. Friedman and Z. Yakhini, "On the sample complexity of learning bayesian networks," in *Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1996, pp. 274–282.

[29] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos, "Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation," *The Journal of Machine Learning Research*, vol. 11, pp. 171–234, 2010.

[30] R. R. Bouckaert, *Bayesian network classifiers in weka*. Department of Computer Science, University of Waikato, 2004.

[31] W. Buntine, "Theory refinement on bayesian networks," in *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1991, pp. 52–60.

[32] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," *Machine learning*, vol. 20, no. 3, pp. 197–243, 1995.

[33] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *Information Theory, IEEE Transactions on*, vol. 14, no. 3, pp. 462–467, 1968.

[34] P. M. Lee, *Bayesian statistics: an introduction*. John Wiley & Sons, 2012.

[35] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," Association of Computational Linguistics (ACL), 2007.

FuAGGE: A Novel System to Automatically Generate Fuzzy Rule Based Learners

Romaissaa Mazouni and Abdellatif Rahmoun
Djilali Liabes University, Computer Science Departement, Algeria
E-mail: rom.mazouni@yahoo.com

Eric Hervet
Moncton University, Computer Science Departement, NB, Canada

Keywords: FuAGGE, grammatical evolution, fuzzy set covering, fuzzy rule-based learners, data mining

Received: January 6, 2016

Data in real world applications are in most cases linguistic information that are ambiguous and uncertain. Hence, such data should be handled by fuzzy set representation schemes to increase expressiveness and comprehensiveness. Moreover, mining these data requires ways to generate automatically useful information/knowledge through a set of fuzzy rules. This paper proposes a novel system called FuAGGE that stands for Fuzzy Automatic Generator Genetic Expression. The FuAGGE approach uses a grammar based evolutionary technique. The grammar is expressed in the Backus Naur Form (BNF) and represents a fuzzy set covering method. The grammar is mapped into programs that are themselves implementations of fuzzy rule-based learners. Binary strings are used as inputs to the mapper along with the BNF grammar. These binary strings represent possible potential solutions resulting from the initializer component and the building blocks from Weka, a workbench that contains a collection of visualization tools and algorithms for data analysis and predictive modeling. This operation facilitates the induction process and makes induced programs shorter. FuAGGE has been tested on a benchmark of well-known datasets and experimental results prove the efficiency of the proposed method. It is shown through comparison that our method outperforms most recent and similar, manual techniques. The system is able to generate rule-based learners specialized to specific domains, for example medical or biological data. The generated learners will be able to produces efficient rule models. The produced rule models will achieves more accurate classification for the specific used domain.

Povzetek: Razvit in testiran je algoritem FuAGGE za učenje mehkih pravil, ki omogoča učenje s slovnico, prilagojeno vsaki problemski domeni.

1 Introduction

Fuzzy systems have gained popularity due to their ability to express ambiguous and uncertain information in various real-world applications [1, 2, 3]. Hence, in order to take advantage of the well-established foundations of predicate logic-based expert systems, researchers focus on extracting fuzzy knowledge from available numerical data [4]. Yet, in [5], the authors propose a fuzzy extension of a rule learner called FILSMR, where they apply fuzzy set concepts [6] to the algorithm PRISM [7]. Later on, Wang and al proposed an algorithm called Fuzzy-AQR [8], where they introduced a seed that represents the highest membership to the positive set. It is used to generate an initial rule which should cover the seed. In [9], Van Zyl and al propose FuzzConRi. It is mainly based on the CN2 method [10]. FuzzConRi is composed of two layers: The upper layer uses a set covering approach, while the lower one is used to induce a single rule. Later on, in [11, 12], the same previous authors propose the Fuzzy-BEXA framework. This framework consists of three layers: The top layer implements a set covering approach, the middle layer uses heuristics to guide the search, and the lower layer is dedicated to refining conjunctions. Huhn and al introduced a new rule learner called FURIA [13]. This is a fuzzy extension of the RIPPER algorithm [14], except the fact that it learns fuzzy unordered rule sets instead of conventional rule lists. It also uses a rule stretching method to solve the problem of uncovered records. In 2014 Swathi et al [15] used fuzzy classification entropy to generate fuzzy decision tree (FDT) and later the parameters of FDT are tuned to further increase the accuracy of FDT. Nevertheless, all of existing methods in the current literature rely on designing fuzzy rule learners manually .In this work we present a grammar evolution based system that automatically generates such rules. We believe that automating the process of writing fuzzy rule based classifiers can highly improve the efficiency of such classifiers. Yet, an automatic generation of fuzzy rule based classifiers alleviates the burden of writing long source codes. The proposed system relies on a grammar that represents the overall structure for fuzzy set covering approaches.

The paper is composed of six sections in addition to the above introduction: Section 2 describes the Grammatical Evolution method. Section 3 describes the fuzzy rule-based classifiers and their features. Section 4 illustrates the automatic generation of fuzzy rule learners. Section 5 offers a description of how the system automatically generates rule-based learners. Section 6 presents the results obtained with the proposed system, and finally section 7 concludes the paper.

2 Grammatical evolution

Grammatical Evolution is a special case of grammar-based genetic programming that uses a mapping procedure between the genotypes and the phenotypes of individuals [16]. Grammatical evolution can generate complete programs in an arbitrary programming language using populations of variable-length binary strings. These computer programs are solutions to a given problem [17, 18]. When using Grammatical Evolution to generate solutions to a given problem, there is no need to pay attention to the genetic operators and how they are implemented: Grammatical Evolution ensures the validity of the generated programs for free. As described in [19], Grammatical Evolution applies concepts from molecular biology to the representational power of formal grammars [20]. The genotype-phenotype mapping in Grammatical Evolution has the advantage, over solution trees used in traditional Genetic Programming, to allow operators to act on genotypes. By analogy to biological DNA, a string of integers that represents a genotype is called a chromosome, and the integers that compose a chromosome are called codons. A Backus Naur Form grammar definition must be introduced prior to using Grammatical Evolution to solve a given problem. This grammar describes the output language produced by the system in [21, 22], and is used along with the chromosomes in the mapping process, which consists of mapping non-terminals to terminals, and completed through converting the binary string data structure into a string of integers, which is finally passed from the genetic algorithm on to the Grammatical Evolution system. The string of integers then goes through a translation process, where rules of the BNF grammar are selected. The production rules, which can be considered equivalent to amino acids in genetics, are combined to produce terminals, which are the components making up the final program. One problem that can occur when mapping binary strings is the production of short genes, meaning we run out of genes, but there are still some non-terminals to map. A solution to this issue is to wrap out the individuals, and to reuse the genes. A gene can be used several times in the mapping process. It is also possible to declare some individuals as invalid by penalizing them with a suitable harsh fitness value. The rules selection is performed by using the mod ulo operator, and every time a codon (an integer) is read, it is divided by the number of the rule's choices. The remainder of this division is the number of the rule to be selected. Grammatical Evolution can be used to automatically generate fuzzy classifiers by using a grammar that represents the overall structure of these fuzzy classifiers. The initial population which is a group of individuals (integer arrays) is used along with the grammar in the mapping process. The GE Mapper produces phenotypes (fuzzy classifiers) which are evaluated using the fitness function. Phenotypes go through an evolution process in the Search Engine until a stopping criterion is met and a best fit fuzzy classifier is found. The different modules of the whole Grammatical Evolution approach are illustrated in Fig. 1.

3 Fuzzy rule induction

One of the main and most studied tasks of data mining is classification, which aims at predicting to which class belongs a certain element according to any given classification model. The classification model can be a set of decision rules extracted, using a given dataset, and which represents local patterns of a model in this dataset. Decision rules can be extracted from other knowledge representations such as Decision Trees [23]. Moreover, they can be drawn out directly from the training set, or may also be induced by using evolutionary algorithms, more specifically, genetic algorithms or genetic programming. Fuzzy set theory lead researchers to look for fuzzy alternatives for data mining problems such as fuzzy induction learners, fuzzy decision trees, and fuzzy clustering. The present work focuses on the sequential covering rule induction paradigm. The fuzzy version of the sequential covering paradigm is called fuzzy set covering. The proposed idea is to train one rule at a time, remove the examples it covers, and repeat the process until all data is fully covered, as described in [24]. There are plenty of proposed algorithms that follow the fuzzy set covering paradigm. These algorithms usually differ in some components such as the search mechanisms [25]. There are three different search strategies: The bottom-up one starts with a random sample from the dataset, then generalizes it; The top-down strategy starts with an empty rule, then specializes it by adding preconditions to it; Finally the bidirectional strategy, which is the least common one, allows to either generalize or specialize the candidates. There are also two different categories of search methods. The most used ones are the greedy method (ex: FILSMR [7]) and the beam method (ex: FuzzConRi [9], FuzzyBEXA [12]). Covering algorithms have different ways to evaluate rules. Some of the existing methods are: The fuzzy entropy (ex: fuzzy ID3 [26]), the Laplace estimation (ex: Fuzzy- BEXA [12], FuzzConRi [9]), the Fuzzy Bexa framework (fuzzy purity, the fuzzy ls-content and fuzzy accuracy function), and the fuzzy info gain (ex: FILSMR [6], Fuzzy-AQR [8]). The final component that differentiates covering algorithms is the pruning method, which consists in handling over fitting and noisy data. There are two types of pruning: pre-pruning that deals

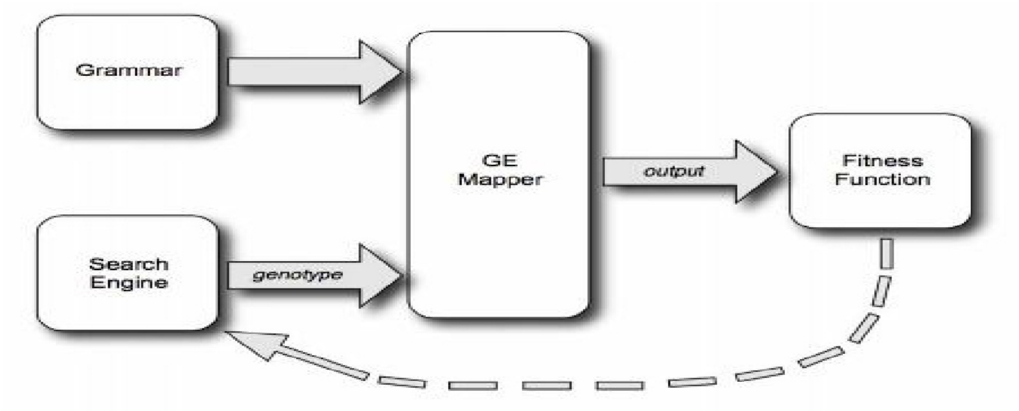


Figure 1: Grammatical Evolution Modules

with over fitting and noisy data, and post-pruning that deals with rejection and conflicts in order to find a complete consistent rule set. Pre-pruning offers the ability to obtain a high-speed model, while post-pruning helps getting simpler and more accurate models. The grammar we use is a context-free grammar in Backus Naur form that contains all elements necessary to build a basic fuzzy rule learner, following the fuzzy set covering paradigm. It contains 19 production rules, each one representing non-terminal symbols, and 40 terminal symbols describing the elements. The grammar produces fuzzy rule set without any specific order needed when applying them. This process provides different initialization, refinement, and evaluation techniques.

4 Automatic generation of fuzzy rule learners

To build a system that is able to generate fuzzy rule-based learners, we used a context-free grammar that represents the whole structure of the sequential fuzzy set covering paradigm (Fig. 2). This grammar was built after reviewing different fuzzy rule set inducers in the existing literature. It contains 19 rules and 40 terminals, where each terminal stands for a building block performing an action. Algorithm 1 illustrates the building block represented by the terminal Include1Selector in the rule number 11 in the grammar. It is worth mentioning that this method is the first one quoted in the literature that automatically generates a fuzzy rule induction algorithm. To do this, we put in place a grammar evolution scheme. The decision rules model has been selected on the basis of intuitiveness and comprehensiveness. In the following section, we propose a system that combines Grammatical Evolution with a context-free grammar, in order to generate code fragments possessing the ability to generate accurate, noise-tolerant, and compact decision fuzzy rule sets.

Algorithm 5 `Include1Selector` (rule R).

```
refinements = $\emptyset$
for i $= 0$ to i $<$ numberAtt do
    for for j $= 0$ to j $<$ numberVal($Att_i$) do
        newAntecedant = R U ($Att_i$,$Val_j$)
        refinements = refinements U newAntecedant
    end for
end for
return refinements
```

4.1 Proposed system: FuAGGE

The suggested method includes five main components. To start, all what we need is a grammar that represents the overall structure of all manually designed fuzzy rule learners that obeys to the fuzzy set covering technique. We use some building blocks from Weka, which is a workbench that contains a set of visualization tools and a set of algorithms for data analysis and predictive modeling. This step will help reading the dataset files, and testing the newly generated rule based learners. It might be seen as "code reusing". We also need some machine learning datasets in order to train and test the newly generated learners. The datasets we used were taken from the UCI machine learning repository [27]. Indeed, we need several and various datasets, so that when fuzzy rule learners (candidate solutions) are trained, these are not tailored to a specific domain. Finally, we have to take care of the mapper of the Grammatical Evolution [17], modified in such a way that when it reads terminals, pieces of Java code representing these terminals actions are generated. At this stage, we used the GEVA framework to implement the whole system [18]. The most important component of our system is the mapper, which must have the ability to read the integer values from the chromosomes / candidate solutions that are called in this paper the FuAGGE classifiers. Then, we had to choose the appropriate corresponding rule to a given non-terminal, import some of the already coded Weka building blocks, and insert them along with the terminals corresponding to Java code into the Java class that builds the new FuAGGE classifier. Fig. 2 represents the whole system and its modules. Individuals in this system are represented as integer arrays that are to be mapped to the fuzzy rule learners. Every array is read one integer at a time [28], and every integer is divided by the number of choices of the current rule. The number of the rule to be chosen and applied next is the remainder of the division (Eq. 1), where N is the currently read integer, Nc is the number of choices of the next rule to apply, and Idrule is the identifier of the rule selected by the mapper. This rule will be mapped into Java code (Fig. 3).

$$N \bmod N_c = Id_{\text{rule}} \tag{1}$$

When using Grammatical Evolution to solve a given problem, we need a measure that favors the selection of the best individuals among a population of possible solutions. The metric, also called the fitness function, used in this work to evaluate the FuAGGE classifiers generated during the evolution process, is the accuracy method. After the initialization of the first population, individuals go through the mapping process and are integrated into Java programs (FuAGGE classifiers), which are the actual classifiers that need to be compiled and executed (trained and tested) using different datasets. Each fuzzy rule learner has a set of different accuracies accuracy per dataset). The average of all these accuracies is used as the classifiers overall accuracy, so we are able to compare the different FuAGGE classifiers over a population. The following equation defines the overall accuracy of a FuAGGE classifier in a given population. acci;j represents the accuracy of the FuAGGE classifier i using the dataset j, and h is the number of datasets:

$$f(i) = \frac{\sum\limits_{j=0}^{h} acc_{ij}}{h} \tag{2}$$

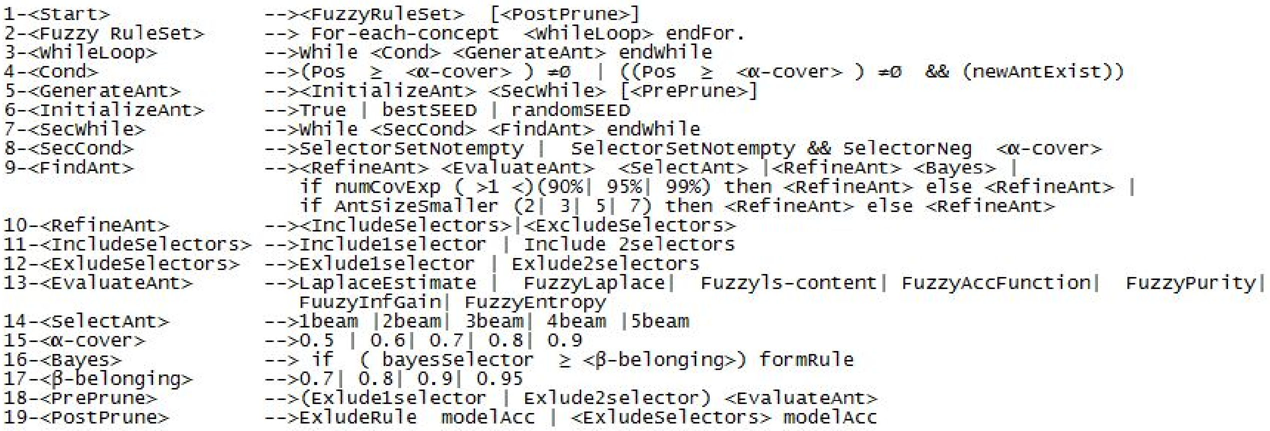
When using Grammatical Evolution, we do not need to check the off-springs generated after a mutation or a crossover operation, because the genetic operators are applied on genotypes: Since these are represented by integer arrays, a crossover or a mutation operation over them generates the same type of arrays, which are then mapped using the grammar. This might not be always the case if we were using the Context Free Grammar Genetic Programming method, because the genetic operators are applied on the phenotypes (syntax trees), and this may generate unsuitable off-springs.

5 Experimental results and discussion

Three components are needed so that the system can start the evolution process: The grammar introduced in section. 4, the meta datasets, and finally the Grammatical Evolution parameters. The parameters have been set as follows: the number of generation has been set to 60, the population size to 150, the mutation probability to 1%, the crossover probability to 80%, and the selection method is the tournament selection with generational replacement. We should note here that all these parameters have been fixed empirically after that we analyzed a certain number of trials and experimentations.

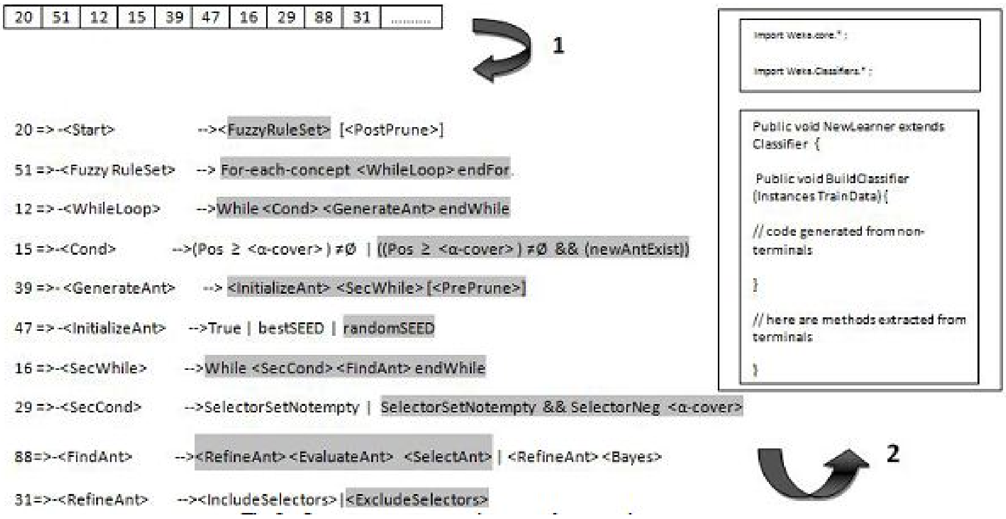
In order to evaluate the newly generated fuzzy learners, we computed the accuracies of two manually designed fuzzy rule learners using all ten datasets. The last column of Table 1 reports accuracies of the new generated fuzzy learners (FuAGGE classifier), while the remaining columns shows the accuracies of the two manually designed classifiers (Furia, Fuzzy CN2). The upper six rows reports the accuracies of the fuzzy rules sets generated by the FuAGGE-classifier, and the two baseline ones show the accuracies using only the meta-training set (each row represents the test accuracy of a single set from the meta training set). These accuracies are reported here to show the success of the training phase, while the lower four rows of the table show the predictive accuracies of the FuAGGE-classifier for sets that were part neither of the training, nor of the validation phase.

After the grammar has been created, the data prepared, and the implementation and testing phases launched, the system proved its ability to produce rule learners that are competitive with manually designed learners. We can clearly see the performance of these formers in Table 1. We should clarify that these accuracies were calculated by averaging the accuracies of the fuzzy rule model generated by the FuAGGE-classifier for each test set over the 10 iterations of the 10-fold-cross-validation method used during experiments. This also applies to the rest of the benchmark rule based learners used for purpose of comparison. It is worth mentioning that in Table 1, the new generated fuzzy learners has almost the same results as the other methods, and if we compare only the baseline methods with each other's we can clearly notice that the Fuzzy CN2 records 5 wins over 1 for Furia even though Furia is more sophisticated: It uses a growing, pruning and optimization phase just like the crisp version RIPPER and a fuzzy rule stretching method. Now if we look at the FuAGGE-classifier accuracies, we can notice how close these accuracies are to the baseline algorithms accuracies, which is very interesting due to the fact that the FuAGGE-classifier is automatically generated, and this removes a great deal of human time coding tasks. Human designers can easily go wrong when parameterizing an algorithm during the design process, contrariwise the chance of having wrong parameters when using automatic evolution of algorithms is very low. The last four rows show that the FuAGGE- classifier records 1 win against the baseline classifiers (Puba), and an equality with Fuzzy CN2 (Haberman). For the Ion dataset, the FuAGGE results are very close to the best accuracy (90.38% versus 91.17%). These results prove that the pro-



```
1-<Start>             --><FuzzyRuleSet>  [<PostPrune>]
2-<Fuzzy RuleSet>     --> For-each-concept  <WhileLoop> endFor.
3-<WhileLoop>         -->While <Cond> <GenerateAnt> endWhile
4-<Cond>              -->(Pos  ≥  <α-cover> ) ≠Ø  | ((Pos  ≥  <α-cover> ) ≠Ø  && (newAntExist))
5-<GenerateAnt>       --><InitializeAnt> <SecWhile> [<PrePrune>]
6-<InitializeAnt>     -->True | bestSEED | randomSEED
7-<SecWhile>          -->While <SecCond> <FindAnt> endWhile
8-<SecCond>           -->SelectorSetNotempty |  SelectorSetNotempty && SelectorNeg  <α-cover>
9-<FindAnt>           --><RefineAnt> <EvaluateAnt>  <SelectAnt> |<RefineAnt> <Bayes> |
                         if numCovExp ( >1 <)(90%| 95%| 99%) then <RefineAnt> else <RefineAnt> |
                         if AntSizeSmaller (2| 3| 5| 7) then <RefineAnt> else <RefineAnt>
10-<RefineAnt>        --><IncludeSelectors>|<ExcludeSelectors>
11-<IncludeSelectors> -->Include1selector | Include 2selectors
12-<ExludeSelectors>  -->Exlude1selector | Exlude2selectors
13-<EvaluateAnt>      -->LaplaceEstimate |  FuzzyLaplace|  Fuzzyls-content| FuzzyAccFunction|  FuzzyPurity|
                         FuuzyInfGain| FuzzyEntropy
14-<SelectAnt>        -->1beam |2beam| 3beam| 4beam |5beam
15-<α-cover>          -->0.5 | 0.6| 0.7| 0.8| 0.9
16-<Bayes>            --> if  ( bayesSelector  ≥ <β-belonging>) formRule
17-<β-belonging>      -->0.7| 0.8| 0.9| 0.95
18-<PrePrune>         -->(Exlude1selector | Exlude2selector) <EvaluateAnt>
19-<PostPrune>        -->ExludeRule  modelAcc | <ExludeSelectors> modelAcc
```

Figure 2: The Fuzzy Set Covering Grammar.



20	51	12	15	39	47	16	29	88	31	....

```
20 => -<Start>           --><FuzzyRuleSet>  [<PostPrune>]
51 =>-<Fuzzy RuleSet>    --> For-each-concept  <WhileLoop> endFor.
12 => -<WhileLoop>       -->While <Cond> <GenerateAnt> endWhile
15 =>-<Cond>             -->(Pos ≥ <α-cover> ) ≠Ø  | ((Pos ≥ <α-cover> ) ≠Ø  && (newAntExist))
39 =>- <GenerateAnt>     --> <InitializeAnt> <SecWhile> [<PrePrune>]
47 => -<InitializeAnt>   -->True | bestSEED | randomSEED
16 =>-<SecWhile>         -->While <SecCond> <FindAnt> endWhile
29 =>-<SecCond>          -->SelectorSetNotempty |  SelectorSetNotempty && SelectorNeg <α-cover>
88=>-<FindAnt>           --><RefineAnt> <EvaluateAnt>  <SelectAnt> | <RefineAnt> <Bayes>
31=>-<RefineAnt>         --><IncludeSelectors> |<ExcludeSelectors>
```

```
Import Weka.core.* ;

Import Weka.Classifiers.* ;
```

```
Public void NewLearner extends
Classifier {

 Public void BuildClassifier
(Instances TrainData) {

// code generated from non-
terminals

}

// here are methods extracted from
terminals

}
```

Figure 3: Integer Arrays to a Java Code Mapping Process.

posed approach is very promising. However, the FuAGGE system is time consuming while evolving FuAGGE classifiers and requires high computational power. Actually, if run on an ordinary computer, the evolution process can take up to one week of continuous calculation. We should also note that this version of the system does not handle missing values. The system eliminates instances with missing value before using the datasets. We should also note that this version uses only numeric attributes. And finally, the data has been fuzzified manually which is really time consuming . This should be tackeled in the next version of the system by giving it the ability to handle missing values either by replacing missing values with the mean or median of the current class, or by the most common attribute values. We could also give the system the ability to automatically fuzzify data, this can be done simply by coding the steps required to fuzzify the data, or by following a method proposed by Swathi et al. [29, 30] which converts numerical attributes into fuzzy membership functions using fuzzy clustering. Swathi et al. [30] presented two heuristic algorithms for the estimation of parameterized family of membership functions, namely, triangular and trapezoidal.

Table 1: Accuracy rates (%) using both meta-sets.

	Furia	Fuzzy-CN2	FuAGGE Classifier
Iris	92.06	95.20	95.13
Pima	75.65	79.10	60.20
Glass	69.63	65.90	70.43
Wine	65.82	97.90	59.81
Vehicle	70.57	73.30	73.58
Wbc	95.28	98.11	96.26
Ecoli	83.63	83.90	79.82
Ion	91.17	89.50	90.38
Puba	67.83	57.44	70.10
Haberman	72.55	72.92	72.91

Table 2 presents the accuracies of rule-based learners generated using our proposed system AGGE [31] in the first column, and those generated using the fuzzy exten-

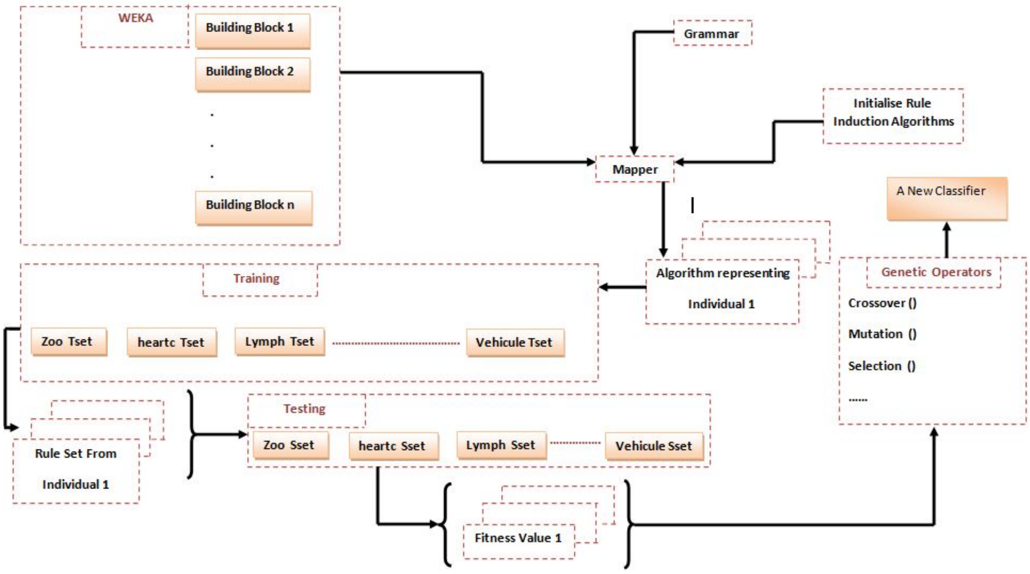


Figure 4: Modules of the proposed system.

AGGE Classifier	FuAGGE Classifier	
Iris	95.09	95.13
Ion	88.14	90.38
Pima	75.34	60.20
Wine	92.03	59.81
Vehicle	90.03	73.58
Glass	62.69	70.43
WBC	94.93	96.26
Ecoli	73.24	79.82
Haberman	67.45	72.81
Puba	76.44	70.10

sion presented in this paper FuAGGE in the second column. It is worth noting that the parameters were set as follows: generations=60, population size=150, crossover=0.8, mutation=0.1, for both AGGE systems. The upper 6 rows represent the accuracies using the meta training set, and the lower rows the accuracies using the meta testing sets. We notice that even though the grammar used for the FuAGGE system is very straightforward, the system's classifiers recorded 7 wins versus 3 wins for the crisp AGGE classifiers. This is due to the efficiency of fuzzy systems versus crisp ones in the classification domain. FuAGGE proves to be more interesting than AGGE in terms of the efficiency of its generated classifiers. However, if the number of generations or the mutation and crossover rates are too high, we might lose the efficiency of the system because of the destructive nature of its operators. Accordingly, setting AGGE and FuAGGE parameters is highly sensitive.

6 Conclusion

In this paper we present a new approach able to automatically produce fuzzy rule-based learners. The system is mainly based on a BNF context free grammar representing the overall structure of the baseline fuzzy rule learners. It also applies the concept of Grammatical Evolution to produce the best fit rule learner. Experiments on a commonly used data benchmark show that it is possible to automatically produce rule learners that can compete with manually designed ones, even with a basic grammar. This is of great importance, because the proposed method reduces the burden of writing manually thousands of lines of source code, and offers a better parameterizing of programs. Our method can be easily extended to a wide range of applications, provided that sufficient related data is available. This may open interesting opportunities and new trends in data mining and computational intelligence. For instance, our system could produce rule based learners dedicated to medical data, biological data, or physics and engineering data. Algorithms would be parameterized in a way to better fit a specific domain and as a result, would achieve more accurate results in this particular field. Another way of extending the system is to change the grammar to make the system capable to generate other data mining algorithms, such as clustering or fuzzy clustering algorithms.

References

[1] X. Wang, D. Nauck, M. Spott, R. Kruse. (2007) Intelligent data analysis with fuzzy decision trees, *Soft Computing*, Springer, pp. 439–457.

[2] H. Shen, J. Yang, S. Wang, X. Liu, (2006) Attribute weighted mercer kernel based fuzzy clustering algo-

rithm for general non-spherical datasets, *Soft Computing*, Springer, 10 (11) pp. 1061–1073.

[3] R. Ab Ghani, A. Salwani, Y. Razali (2015) Comparisons between artificial neural networks and fuzzy logic models in forecasting general examinations results, *2nd International Conference on Computer, Communications and Control Technology*, Malaysia, pp. 253–257.

[4] J. Zyl (2007) *Fuzzy set covering as a new paradigm for the induction of fuzzy classification rules*, University of Mannheim.

[5] C. Wang, J. Liu, T. Hong, S. Tseng (1997) FILSMR: a fuzzy inductive learning strategy for modular rules, *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, pp. 1289–1294.

[6] L. A. Zadeh (1965) Fuzzy sets, *Information and Control*, pp. 333–353.

[7] D. Cendrowska (1987) PRISM: An Algorithm for Inducing Modular Rules, *In International Journal of Machine Learning Studies*, pp.349–370.

[8] C. Wang, C. Tsai, T. Hong, S. Tseng, (2003) Fuzzy Inductive Learning Strategies, *Applied Intelligence*, Publisher, 18 (2) pp. 179–19.

[9] J. Zyl, I. Cloete, (2004) FuzzConRI- A Fuzzy Conjunctive Rule Inducer, *Proceedings of the ECML/PKDD04 Workshop on Advances in Inductive Rule Learning*, pp. 194–203.

[10] P. Clark, R. Boswell,(1991) Rule induction with CN2: Some recent improvements, *Proceedings of the Sixth European Working Session on Learning*,pp. 151–163.

[11] H. Theron, I. Cloete, (1996) BEXA: A covering algorithm for learning propositional concept descriptions, *Machine Learning*, Publisher, pp. 5–40.

[12] J. Zyl, I. Cloete, (2005) Fuzzy rule induction in a set covering framework, *IEEE Trans. Fuzzy Syst*, pp. 93–110.

[13] J. Huhn, E. Hullermeier, (2009) FURIA: an algorithm for unordered fuzzy rule induction, *Data Mining and Knowledge Discovery*, pp. 293–319.

[14] J. Furnkranz, (1994) Pruning Methods for Rule Learning Algorithms, *Proceedings of the 4th International Workshop on Inductive Logic Programming*,USA, pp. 231–336.

[15] J. N. Swathi, B.B. Rajen, P. Ilango , M. Khalid , B. K. Tripathy, (2014) Induction of fuzzy decision trees and its refinement using gradient projected-neuro-fuzzy decision tree, *International Journal of Advanced Intelligence Paradigms*,pp. 346–369.

[16] J. Montana (1994) Strongly Typed Genetic Programming, *Evolutionary Computation Journal*, pp. 199–230.

[17] A. Nohejl, (2011) *Grammar Based Genetic Programming*, Charles University of Prague.

[18] H. Li, L. Wong (2014) Knowledge Discovering in Corporate Securities Fraud by Using Grammar Based Genetic Programming, *Journal of Computer and Communications* pp. 148–156.

[19] I. Dempsey, M. O'Neill, A. Brabazon (2009) *Foundations in Grammatical Evolution for Dynamic Environments*, Springer.

[20] M. O'Neill, E. Hemberg, C. Gilligan, E. Bartley, J. McDermott, A. Brabazon (2008) GEVA: Grammatical Evolution. in Java, *SIGEVOlution ACM*, (3) pp.17-23.

[21] A. De Silva, F. Noorian, R. Davis, P. Leong (2013) A Hybrid Feature Selection and Generation Algorithm for Electricity Load Prediction using Grammatical Evolution, *12th International Conference on Machine Learning and Applications*, Miami USA, pp. 211-217.

[22] P. Gisele, A. Freita, (2010) *Automating the Design of Data Mining Algorithms*, Springer Berlin Heidelberg.

[23] J. R. Quinlan, (1986) Induction of Decision Trees, *Machine Learning* , Kluwer Academic Publishers, pp. 81–106.

[24] A. Freitas,(2002) *Data mining and Knowledge Discovery with evolutionary algorithms*, Springer Verlag.

[25] M. Corn, M. A. Kunc, (2015) Designing model and control system using evolutionary algorithm, *8th Vienna International Conferenceon Mathematical Modelling — MATHMOD 2015*, Vienna, pp. 536–531.

[26] M. Dong, R. Kothari (2014) Classifiability Based Pruning of Decision Trees, *Neural Networks. Proceedings IJCNN '14, International Joint Conference on Neural Networks (IJCNN)*, Washington, pp. 1793–1947.

[27] UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/

[28] A. Nohejl, (2009) *Grammatical Evolution*, Charles University of Prague.

[29] J. N. Swathi, I. Paramasivam, B. B. Rajen, M. Khalid (2015) A Study on the Approximation of Clustered Data to Parameterized Family of Fuzzy Membership Functions for the Induction of Fuzzy Decision Trees, *Cybernetics and Information Technologies*,pp. 75–96.

[30] B. B. Rajen, J. N. Swathi, P. Ilango, and M. Khalid,(2012) Approximating fuzzy membership functions from clustered raw data, *In Proceedings of India Conference (INDICON) Annual IEEE*, India, pp. 487–492.

[31] R. Mazouni, A. Rahmoun, (2015) AGGE: A Novel Method to Automatically Generate Rule Induction Classifiers Using Grammatical Evolution, *Studies in Computational Intelligence*, Springer, Madrid, pp. 270–288.

Software Features Extraction from Object-Oriented Source Code Using an Overlapping Clustering Approach

Imad Eddine Araar
Department of Mathematics and Computer Science
Larbi Ben M’hidi University, Oum el Bouaghi, Algeria
E-mail: imad.araar@gmail.com

Hassina Seridi
Electronic Document Management Laboratory (LabGED)
Badji Mokhtar-Annaba University, P.O. Box 12, 23000 Annaba, Algeria
E-mail: seridi@labged.net

Keywords: feature model, software product line, overlapping clustering, reverse engineering, program analysis

Received: January 16, 2016

For many decades, numerous organizations have launched software reuse initiatives to improve their productivity. Software product lines (SPL) addressed this problem by organizing software development around a set of features that are shared by a set of products. In order to exploit existing software products for building a new SPL, features composing each of the used products must be specified in the first place. In this paper we analyze the effectiveness of overlapping clustering based technique to mine functional features from object-oriented (OO) source code of existing systems. The evaluation of the proposed approach using two different Java open-source applications, i.e. “Mobile media” and “Drawing Shapes”, has revealed encouraging results.

Povzetek: Prispevek vpelje novo metodo generiranja spremenljivk za ponovno uporabo objektno usmerjenih sistemov.

1 Introduction

A software product line, also known as software family, is “a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way” [6]. A feature represents a prominent or distinct aspect that is visible to the user, a quality, or a system characteristic [16]. There are two types of features: (1) the commonalities, that must be included in all products, and (2) the variabilites, which are shared by only some of them. A feature model (FM) provides a detailed description of the commonalities and variabilities, specifying all the valid feature configurations. Driven by the software industrial requirements, i.e. cost and time-to-market, several organizations have therefore chosen to convert to a SPL solution. Such a migration can be achieved using one of three major adoption strategies: proactive, reactive, extractive [19]. Using a proactive approach, the organization analyzes, designs and implements a SPL to support all anticipated products (which are within the scope of the SPL). With the reactive approach, organizations develop their SPLs in an incremental manner. This strategy is appropriate when the requirements of new products in the SPL are somehow unpredictable. The extractive approach is used to capitalize on existing software systems by extracting their commonalities and variabilities. Since the proactive strategy is the most expensive and exposed to risks [19], most researchers are now interested in reengineering commonalities and variabilities from existing systems.

On this matter, the type of artifacts to be used in the SPL reengineering process seems of great importance, since it strongly impacts the quality of the results, as well as the level of user involvement in this process. Most of the existing SPL extractive approaches use various types of artifacts such as FMs of existing systems, requirements documents or other additional data. However, most of the requirement documents are written in natural language and, therefore, suffer from several problems such as scalability, heterogeneity and ambiguity [21, 26]. The software documentation on the other hand may be obsolete after making several changes to the code without updating its documentation. In addition, it happens that some systems are not yet equipped with a FM, knowing that it is indispensable throughout the entire SPL development cycle. In order to obtain a FM of an old system, commonalities and variabilities that characterize such a system must be identified and documented. The manual construction of a FM is an expensive and greedy task [33]. Hence, assisting this process would be of great help.

In this article, we propose a new approach which mines features from Java source code of an existing system for disentangling stakeholder goals. Compared to the existing approaches that mine features, the novelty of the proposed approach is that it provides a generic and reusable features catalogue for a software variant instead of generating a single FM which is specific to a set of product variants. We use an overlapping clustering algorithm in order to minimize the information loss. In the proposed approach, Java programs' elements constitute the initial search space. By conducting a static analysis on the target system, we define a similarity measure that enables to proceed through a clustering process. The result is subgroups of elements each of which represents a feature implementation. The number of clusters is automatically calculated during the mining process which decreases the expert involvement.

The remainder of this paper is structured as follows: Section 2 presents the state of the art and motivates our work. Section 3 describes, step by step, our proposed approach for features reengineering using Java source code of existing systems. Section 4 reports the experimentation and discusses threats to the validity of the proposed approach. Finally, Section 5 concludes and provides perspectives for this work.

2 State of the art

This section carries out a survey of leading papers describing the work carried out so far which are related to reverse engineering feature models. Depending on the type of artifacts used as input, one can distinguish two main subgroups of studies that aim to extract FMs: (1) documentation-based approaches and (2) source code-based approaches.

2.1 Documentation-based techniques

Extraction of FMs from legacy systems can benefit from the existing experience in reverse engineering works. The approach proposed by Graaf et al. [14] consists in migrating an existing architecture to a SPL using automatic model transformations. The transformation rules used are defined using ATL (Atlas Transformation Language). Thus, migration can only be possible if the variability is defined by a meta-model.

Niu et al. [22] present a new approach based on clustering, information theory, and natural language processing (NLP) to extract features by analyzing functional requirements. They use an overlapping clustering algorithm. NLP technique is used to define the similarity between the attributes of FRP (functional requirement profiles); FRPs are abstractions of functional requirements. However, the FRPs and their attributes must be prepared manually which implies a considerable human effort.

Rashid et al. [26] propose a technique based on NLP and clustering for automatic construction of FM from heterogeneous requirements documents. However, the FM generated by their approach was of great size compared with a manually created FM. Hence, the intervention of an expert is always needed to perform pre and/or post-processing. The authors explain that this problem is caused by the used clustering algorithm and irrelevant information contained in the inputs.

Haslinger et al. [15] present an algorithm that reverse engineers a FM for a given SPL from feature sets which describe the characteristics each product variant provides. The features used were obtained by decomposing FMs retrieved from an online repository. Experiments have shown that the basic FMs calculated by this algorithm are identical to the initial models retrieved from the repository.

Ziadi et al. [33] propose an automatic approach to identify features for a set of product variants. They assume that all product variants use the same vocabulary to name the program elements. However, given that their approach uses UML class diagrams as inputs, it doesn't consider the method body. In addition, their approach identifies all common features as a single mandatory feature (a maximal set), that is shared by all the product variants.

Ryssel et al. [27] present a technique based on formal concept analysis (FCA) that analyzes incidence matrices containing matching relations as input and creates FMs as output. The matrix describes parts of a set of function-block-oriented models. Compared to other FCA-based approaches, their approach uses optimization techniques to generate the lattices and FMs in a reasonable time.

2.2 Source code-based techniques

There exist very few studies that have addressed the problem of reverse engineering FMs using the source code as a starting point. Kästner et al. [17] propose a tool, Colored IDE (currently known as the CIDE tool), to identify and mark code fragments that correspond to features. However, the process is still manual and it depends on the experience of the tool user. The CIDE tool seems to be more useful in feature-oriented refactoring tasks.

Loesch et al. [20] propose a new approach for restructuring an existing SPL. Their FCA-based approach consists in analyzing real products configuration files used in a given SPL, and building a lattice that provides a classification of variable features usage in product variants. This classification can be used as a formal basis in interactive tools in order to restructure the variabilities.

Paskevicius et al. [23] propose a framework for an automated generation of FM from Java source code using static analysis and conceptual clustering. The approach uses as input the dependency graph (DG) of a targeted software system. The DG is transformed into a distance matrix that must be analyzed using the CobWeb algorithm [12] in order to create a features hierarchy. This latter is used to generate the final FM as Feature Description Language (FDL) descriptors and as Prolog rules. Their approach may be useful during the partial configuration of a given system, for example, to derive a light version of a system.

Al-Msie'deen et al. [25] propose an approach for generating FMs from Java source code of a set of existing systems. They suppose that the analyzed systems use the same vocabulary to name the program elements, i.e. product variants belonging to the same SPL. First, they explore candidate systems to extract OO building elements (OBE). These items are then analyzed combining FCA analysis, Latent Semantic Indexing (LSI), and structural similarity to identify features. Their approach has given good results. However, the analyzed systems may not use the same vocabulary to name OBEs, which means that the used lexical similarity cannot be always reliable.

2.3 Synthesis

As we can see from the preceding sections, most of the existing approaches that address the problem of reverse engineering FMs from existing systems are semi-automatic, and use the documentation and textual descriptions as inputs. However, such a practice involves several challenges. Although the input data type in a given approach is strongly linked to its purpose and usage type, the abstraction level and formalization of such data must also be considered. Ziadi et al. [33] use as input the parts of UML class diagram, which is likely to omit many details related to the variability in the program implementation. FM generated from requirements in [26] was imprecise and very large. The authors justified their finding by the heterogeneous and especially textual nature of the inputs. These latter are likely to be filled with imprecise language commonly used in conversations. Besides the ambiguity, scalability is a problem in the context of SPL. Indeed, we can find a significant number of documents associated with a given product variant, each of which is very large in terms of size. Other works such that given by Haslinger et al. [15] use a set of FM of existing systems in order to derive the SPL' feature model. However, existing systems do not always have a FM, and even if exists it may not be up to date and, therefore, does not truly reflect the variabilities of these systems.

Regarding the used technique, the majority of approaches use classification, since it is the most suitable for the problem of FM generation. Some approaches have used a clustering technique to generate FMs. Paskevicius et al. [23] consider the hierarchy generated by the Cobweb algorithm as a FM, while there is simply not enough information in the input data in order to decide one preferred hierarchy. Moreover, the use of a simple clustering algorithm to generate disjoint groups of program elements can cause information loss, since a program element can be part of more than one feature (crosscutting concerns). Niu et al. [22], address the overlapping problem using an overlapping partitioning algorithm called OPC [5]. Nevertheless, the OPC algorithm requires four parameters to be specified by the user, which significantly minimizes the automation of the task.

Besides clustering, many researchers have used FCA analysis to extract FMs while taking into account the overlap problem. However, there is a limit in the use of FCA. Indeed, not only FCA does not assure that the generated features (formal concepts) are disjoint and cover the entire set of entities [30], but it is also exposed to the information loss problem. For example, in [25], cosine similarity matrices are transformed into a (binary) formal context using a fixed threshold. The information loss caused by such a sharp threshold usage may affect the quality of the result, as claimed by the authors in [2]. The REVPLINE approach proposed by Al-Msie'Deen et al. [2, 25] generates SPL features using as inputs the source code of product variants. They suppose that analyzed products are developed with copy-paste technique, i.e. they use the same vocabulary. However, if this assumption does not hold, it is therefore essential to have a separate FM for each product variant in order to generate the SPL feature model.

3 A tool support for automatic extraction of features

This section presents the main concepts, hypotheses and techniques used in the proposed approach for mining features from source code.

3.1 Goal and core assumptions

The overall aim of the proposed method is to identify all feature implementations for a given software product, based on static analysis of source code. In fact, We recognize that it is essential to have, for every software, a FM which is up to date and reflects the changes that were made to the source code over time. The generated features can be used for documenting a given system as well as for reverse engineering a FM for a SPL. We adhere to the classification given by [16] which distinguishes three categories of features: functional, operational and presentation features. In this article we focus on the identification of functional features; functional features express how users can interact with a software system.

The functional features are implemented using OO program elements (PEs), such as packages, classes, class attributes, methods or elements of method bodies. We also consider that the PEs can be classified in two categories: (1) *atomic program elements* (APE), and (2) *composite program elements* (CPE). An APE is a basic construction element in the program (a variable or a method). A CPE is a composition of atomic and/or composite PEs (i.e. a class or a package). A dependency is a relation between two PEs. An element A depends on B if A references B. For example a method $A()$ uses a variable B or calls a method $B()$. Given a dependency graph $G = (V, E)$, a cluster is defined as a sub-graph $\grave{G} = (\grave{V}, \grave{E})$ whose nodes are tightly connected, i.e. cohesive. Such clusters are considered as functional feature implementations. We suppose also that feature implementations may overlap; a given PE may be shared by the implementations of several features simultaneously.

In addition, since a class represents the main building unit in OO languages, we assume that a generated feature is represented by at least one class. Indeed, a class is generally referred to as a set of responsibilities that simulates a concept or a feature in the application domain [9]. This hypothesis has been

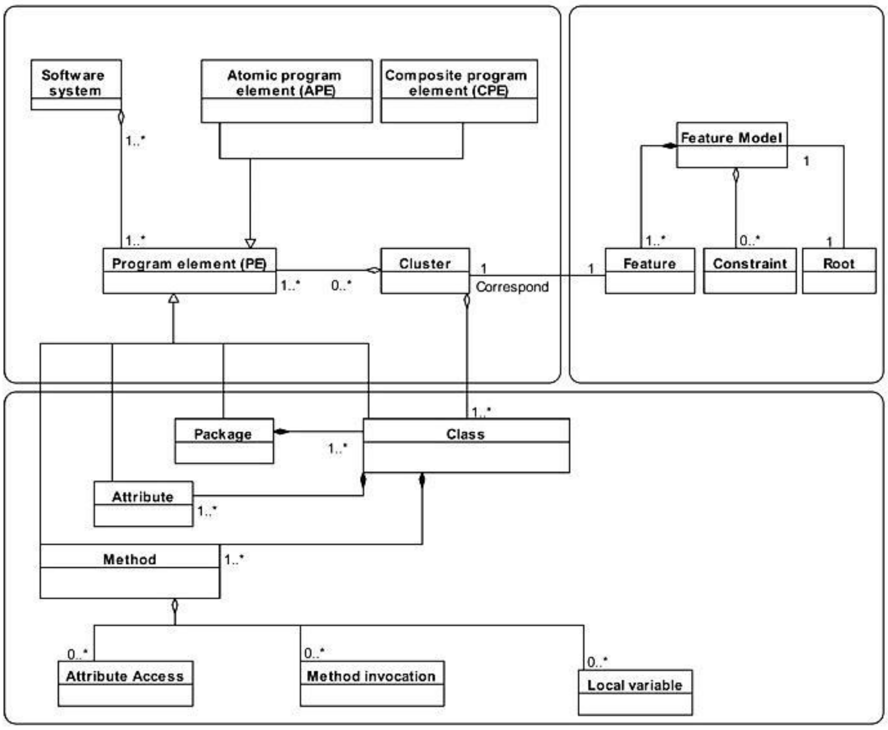


Figure 1: A meta-model to map source code to features.

checked by experiments carried out in [25]. For the sake of simplicity, we make no distinction between classes per se and abstract classes. Taking into account this assumption about classes, interfaces must be pruned from the PEs set. In fact, as a method body must anyway be redefined at the class level when implementing an interface, then considering interfaces in the inputs will not provide additional information to the overall mining process. Consequently, pruning interfaces from the initial PEs set will not affect the validity of our assumptions.

Furthermore, since a software system's features are associated with its behavior, we decided to keep only those PEs which are created by developer to implement the system's specific features. For example, a linked list is a concept from the solution domain which may be implemented in the source code, yet it is not a specific feature of the system. All the concepts we defined for mining features are illustrated in the "program to feature mapping model" of Figure 1.

3.2 Feature mining step by step

This section presents, in a detailed way, the feature reengineering process. Input data were prepared using the method described by Paskevicius et al. [23] while introducing necessary modifications to comply with assumptions and techniques used in our proposed approach. The architecture of our proposed approach for mining features from source code is given in Figure 2.

3.2.1 Extraction of program elements and dependencies

Dependencies of the candidate system was modelled using an oriented dependency graph $G = (F, D)$, such that F is a set of vertices which represent PEs, and D is a set of dependencies. The dependency graph G was generated and saved into an XML file by analyzing ".class" files using *DependencyExtractor*. This latter is a part of a toolbox called *JDependencyFinder*[1]. The use of the Java byte code instead of the source code facilitates the analysis of existing systems whose source code is not available. Moreover, sometimes source code lacks information like, for example, how the compiled code will be organized in execution containers (Jar files). Such information is usually defined in the scripts executed during compilation [7]. The choice of using a dependency graph as input is justified by the nature of the problem as well as the granularity of the processed entities. In fact, we try to build clusters of PEs, i.e. feature implementations, based on functional dependencies between these PEs; a feature implementation is characterized by a strong functional dependency (intra-cluster cohesion) between its composing PEs.

DependencyExtractor was executed through the command line by combining three of its parameters: [-*class-filter*], [-*minimize*] and [-*filter-excludes*]. The [-*minimize*] parameter was used to remove redundant dependencies. In fact, it is often the case that an explicit dependency in the code can be implied from another explicit dependency in that code. Such dependencies do not add anything to the overall connectivity of the graph and, therefore, must be removed. The second parameter, i.e. [-*class-filter*], allows us to select only those dependencies going to/from classes. These latter represent, as explained earlier, the main construction

[1] http://depfind.sourceforge.net/

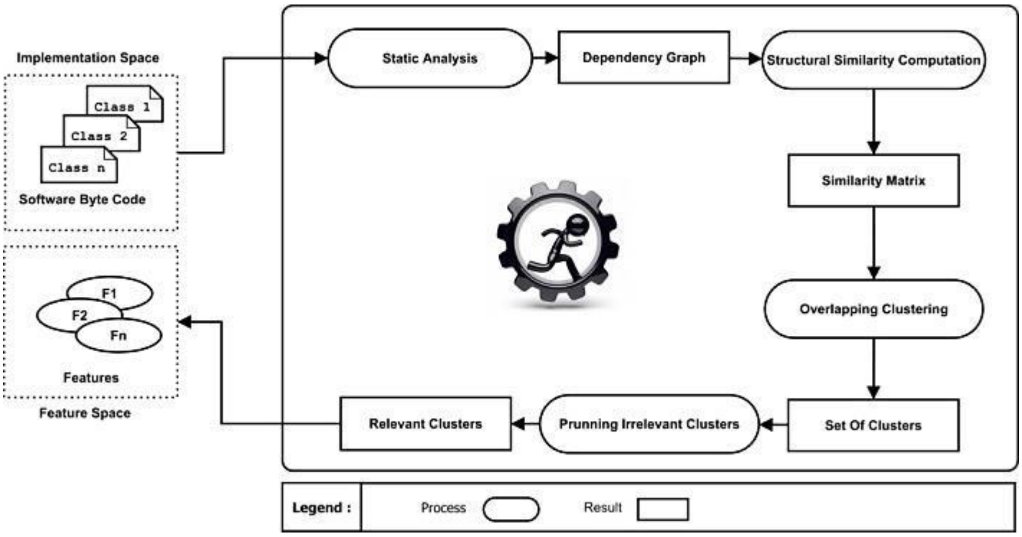


Figure 2: The feature mining process.

units of feature implementations. The choice of using such type of dependencies instead of only considering inter-class dependencies is justified by the fact that such kind of mixed dependencies are likely to enrich the training set. Finally, the third parameter [*-filter-excludes*] is used to remove graph vertices (resp. dependencies) which represents language specific libraries, such that "Java.*" and "javax.*".

3.2.2 Constructing the similarity matrix

The dependency graph generated in the previous step is used in this phase to build a similarity matrix. A native structural distance-based measure was used to evaluate similarity between each pair of vertices. Firstly, the dependency graph G was expressed, as an adjacency matrix C of the size $|F|$, such that $c_{ij} = 1$ means that there exists an explicit dependency from i to j. In order to describe indirect dependencies, the matrix C has been converted into a distance matrix M of size $|F|$ using Floyd-Warshall's all pairs shortest path algorithm (after Floyd [13] and Warshall [31]), such that m_{ij} is equal to the shortest path distance between program elements i and j. The two matrices C and M are asymmetric because the dependency graph G is directed. Given that the used clustering algorithm operates on an undirected graph, the asymmetric distance table M was converted into a symmetric table $\grave{M}$, such that $\grave{m}_{ij} = \grave{m}_{ji} = Min\ (m_{ij}, m_{ji})$. In addition, it is difficult to estimate the similarity between two PEs using absolute distance $\grave{m}_{ij}$. Hence, absolutes distances matrix was converted into a normalized similarity matrix in such a way that two given PEs have a similarity $S(i,j) = 0$ if there is no path between them, and a similarity $S(i,j) = 1$ if they are identical.

3.2.3 Building feature implementations

After preparing the training set using program dependencies, OclustR algorithm [24] was then executed on that data to generate a set of clusters of PEs. Each calculated cluster is considered as the implementation of a single feature. The OclustR passes through two main stages: (1) initialization step, and (2) the improvement step.

The main idea of the initialization phase is to produce a first set X of sub-graphs, i.e. *ws-graphs*, that covers the graph; in this context, each ws-graph consists in a candidate cluster. Afterward, during the improvement phase, a post-processing is performed on the initial clusters in order to reduce their number and overlap. To do this, the set X is analyzed to remove ws-graphs which are considered as *less useful*. These latter are pruned by merging the sets of their vertices with those of a chosen ws-graph.

Formally, let $O = \{PE_1, PE_2, \ldots, PE_n\}$ be a set of PEs. The OclustR algorithm uses as input an undirected and weighted graph $\tilde{G}_\beta = (V, \tilde{E}_\beta, S)$, such that $V = O$, and there is an edge $(v,u) \in \tilde{E}_\beta$ iff $v \neq u$ and $S(v,u) \geq \beta$, with $S(PE_1, PE_2)$ is a symmetric similarity function and $\beta \in [0,1]$ is a user-defined threshold; Each edge $(v,u) \in \tilde{E}_\beta$ is labeled with the value of $S(v,u)$. We assume that each PE must be assigned at least to one cluster, even if the similarity between that element and the cluster's center is very small. Thus, there is an edge $(v,u) \in \tilde{E}_\beta$ iff $v \neq u$ and $S(v,u) > 0$. Consequently, we are sure that every PE in the training set will be assigned to at least one cluster. The OclustR algorithm doesn't need, henceforth, any input parameter, which increase the task automation. The user still can select other values for the parameter β in order to generate features with more fine-grained granularity, so he can have multiple views of the analyzed system with different abstraction levels.

3.2.4 Pruning irrelevant clusters

When constructing the dependency graph, we have selected only those dependencies whose composing nodes contains at least one class, which means that the resulting clusters will be composed of APEs and/or CPEs. Taking into account that classes are considered as the main construction units of feature implementations,

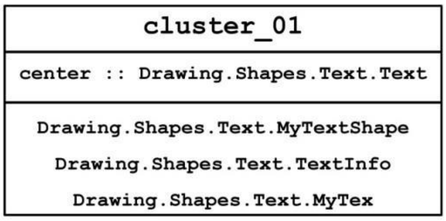

cluster_01
center :: Drawing.Shapes.Text.Text
Drawing.Shapes.Text.MyTextShape
Drawing.Shapes.Text.TextInfo
Drawing.Shapes.Text.MyTex

Figure 3: Example of a mined feature.

clusters which contain only APEs are, consequently, considered as irrelevant and pruned from the result set. In the case of a mixed content cluster, each APE has been replaced by its source class. Indeed, APEs involved in such clusters are only used to provide an optional detailed view. Hence, the final result is a set of relevant clusters composed only by classes. Figure 3 represents a feature (i.e. cluster) mined by our proposed approach for the Drawing Shapes case study (see Section 4.1.1). Given that a cluster computed using OclustR is mainly a ws-graph, it's, consequently, defined by his name which is a unique reference given by the system, his center, and his satellites.

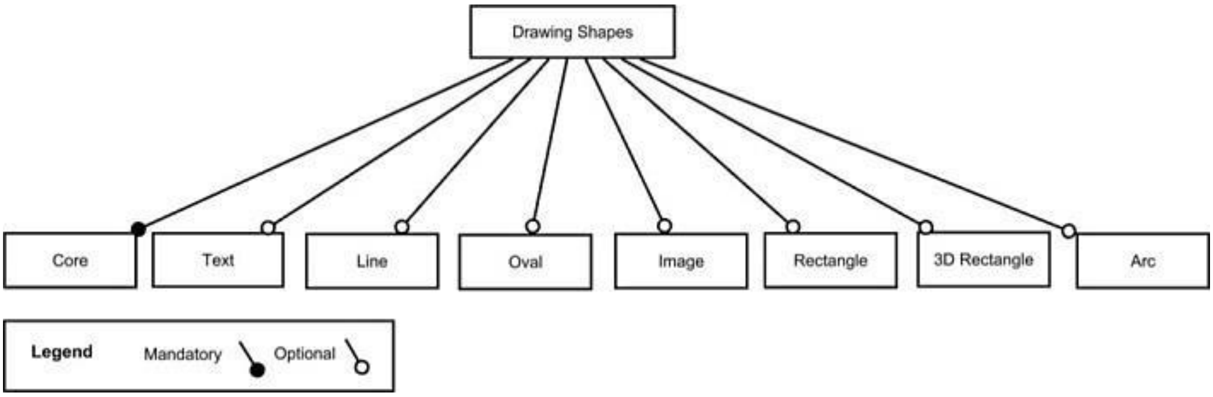


Figure 4: The Drawing Shapes FM.

4 Experimentations

In this section, the experimental setup is described and subsequently, the empirical results are presented in detail, together with a discussion of possible limitations and threats to validity of this study.

4.1 Experimental setup

We fully implemented the steps described in Section 3.2 as a Java tool. We tried to develop the features inference engine as an independent component so that it can be used in a generic and efficient manner. The feature inference engine reads dependency graphs generated using a static analyzer, and seeks to discover software features. In our experiments, we tested our implemented tool on two Java programs, but software written in other OO languages (i.e. C++ or C#) can also be analyzed using our tool, if their dependency graphs are delivered in XML files respecting the DTD[2] used by the *DependencyExtractor* tool. Experiments were made on Windows 7 based PC with Intel i5-4200U processor and 8G of RAM.

4.1.1 Case studies

In order to validate the proposed approach, we conducted experiments on two different Java open-source applications: *Mobile media*[3] and *Drawing Shapes*[4]. The advantage of having two case studies is that they implement variability at different levels. In addition, the corresponding documentations and FMs are available which facilitate the comparison with our results. Moreover, using these two case studies we target two different categories of FMs: (1) a flat FM which is related with the Drawing Shapes case study, and (2) a nested FM related with the Mobile Media case study. Figure 4 and Figure 5 present the corresponding FMs, following the notation proposed by Ferber et al. [10].

The Drawing Shapes SPL represents a small case study (version 5 consists of 8 packages and 25 classes with about 0,6 KlOC). The Drawing Shapes application allows a user to draw seven different kinds of shapes in a variety of colors. The user chooses the shape and the color, and then presses the mouse button and drag the mouse to create the shape. The user can draw as many shapes as desired. The Drawing Shapes software variants were developed based on the copy paste modify technique. In this example, we use version 5 (the full version) which supports *draw 3D rectangle*, *draw rectangle*, *draw oval*, *draw string* and *draw arc* features, together with the *core* one.

The Mobile Media [32] SPL is a benchmark used by researchers in the area of program analysis and SPL research [2, 11, 29]. It manipulates photo, music, and video on mobile devices, such as mobile phones. Mobile Media endured seven evolution scenarios, which led to eight releases, comprising different types of changes involving mandatory, optional, and alternative features, as well as non-functional concerns. In this example, we used the sixth release (R6) which contains OO implementation of all the optional and mandatory features for managing Photos. The used release of

[2] http://depfind.sourceforge.net/dtd/dependencies.dtd

[3] http://homepages.dcc.ufmg.br/~figueiredo/spl/icse08/

[4] https://code.google.com/p/svariants/

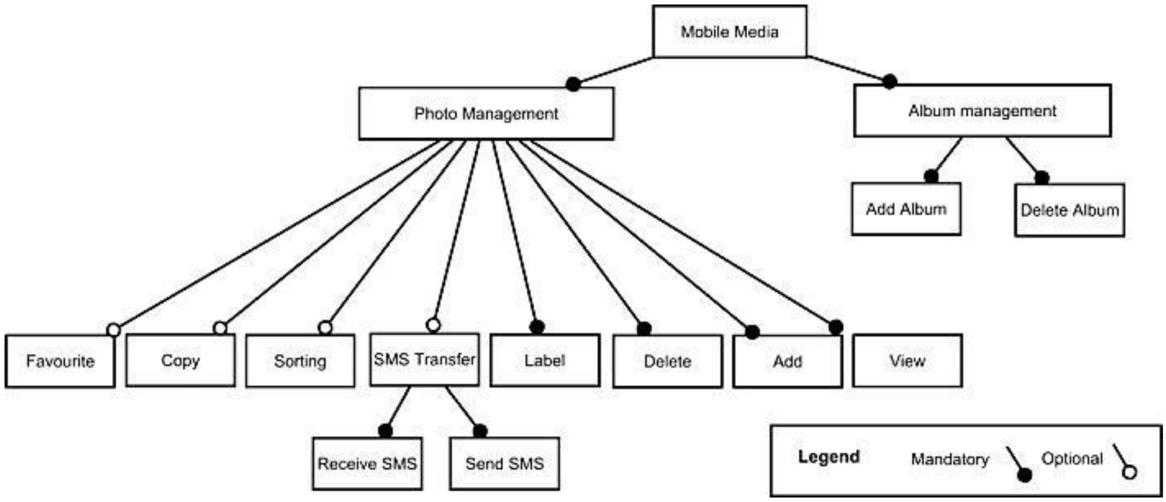


Figure 5: The Mobile Media FM.

Mobile Media consists of 9 packages, 38 classes with about 3 KLOC.

The PEs used to implement the *Exception handling* feature in this Mobile Media release were discarded from the entry set in the pre-processing phase. Indeed, as mentioned before, this paper deals only with functional feature implementations. Additionally, we used the default similarity threshold value for OclustR, i.e. $\beta = 0$.

4.1.2 Evaluation measures

The accuracy and performance of the proposed method is evaluated using an external validation measure. The F-measure [8] is a well-known tool in the information retrieval (IR) domain, which can also be used as a measure for flat clustering quality. We assume that for a given set of program elements $O = \{o_1, o_2, \dots, o_n\}$ we have both: the real, true partition of this set $L = \{L_1, L_2, \dots, L_{K^L}\}$ (we can call L sets as classes, K^L is the number of classes) and clustering partition, the result of OclustR algorithm $C = \{C_1, C_2, \dots, C_{K^C}\}$ (C sets as clusters, K^C is the number of clusters), in order to compute how similar they are.

For the two aforementioned case studies, the true partitions (i.e. listing of real features' PEs) were prepared manually by inspecting the documentation provided by their authors.

F-measure is a mixture of two indices: precision (P), which measures the homogeneity of clusters with respect to a priori known classes, and recall (R), that evaluates the completeness of clusters relatively to classes. A higher precision shows that almost all the cluster elements correspond to expected class. A lower recall, in the other hand, indicates that there are various actual elements that were not retrieved. The convenience of F-measure at this stage of our work is justified by the behavior of this metric. Indeed, F-measure computes the quality of every cluster independently with respect to each class, which allows us to automatically determine the most representative cluster for each class in the gold standard.

Having the previously introduced notation, *precision* of cluster C_i with regard to class L_j is computed as follows:

$$P(C_i, L_j) = \frac{|C_i \cap L_j|}{|C_i|}$$

Recall of cluster C_i with respect to class L_j is computed as follows:

$$R(C_i, L_j) = \frac{|C_i \cap L_j|}{|L_j|}$$

Thus, the F value of the cluster C_i with respect to class L_j is the combination of these two:

$$F(C_i, L_j) = \frac{2 \times P(C_i, L_j) \times R(C_i, L_j)}{P(C_i, L_j) + R(C_i, L_j)}$$

Hence, the F-measure for a cluster C_i is the highest of F values obtained by comparing this cluster with each of known classes:

$$F(C_i) = \max_{L_j \in L} F(C_i, L_j)$$

Despite the fact that the usage of F-measure at this point decreases the expert involvement, this metric seems to be inappropriate when assessing the overall effectiveness of an overlapping clustering solution. According to [3], F-measure does not always detect small improvements in the clustering distribution, and that might have negative implications in the system evaluation/refinement cycles. The authors in [3] have proposed a new metric, i.e. $FBCubed$, that gives a good estimation of the clustering system effectiveness while taking into account the overlapping among clusters. Thus, we decided to evaluate the overall accuracy and performance of our proposed method using $FBCubed$. The $FBCubed$ is calculated using the BCubed Precision and BCubed Recall metrics as proposed in [3]. The BCubed Precision and BCubed Recall are based on the Multiplicity Precision and Multiplicity Recall metrics respectively; which are defined as:

$$MP(o_1, o_2) = \frac{Min(|C(o_1) \cap C(o_2)|, |L(o_1) \cap L(o_2)|)}{|C(o_1) \cap C(o_2)|}$$

$$MR(o_1, o_2) = \frac{Min(|C(o_1) \cap C(o_2)|, |L(o_1) \cap L(o_2)|)}{|L(o_1) \cap L(o_2)|}$$

Where o_1 and o_2 are two program elements, $L(o_1)$ are the classes associated to o_1, $C(o_1)$ are the clusters associated to o_1. $MP(o_1, o_2)$ is the Multiplicity Precision of o_1 wrt o_2, such that o_1 and o_2 share at least one cluster. $MR(o_1, o_2)$ is the Multiplicity Recall of o_1 wrt o_2, such that o_1 and o_2 share at least one class.

Let $D(o_i)$ be the set of PEs that share at least one cluster with o_i including o_i. The BCubed Precision metric of o_i is defined as:

$$BCubed_{Precision}(o_i) = \frac{\sum_{o_j \in D(o_i)} MP(o_i, o_j)}{|D(o_i)|}$$

Let $H(o_i)$ be the set of PEs that share at least one class with o_i including o_i. The BCubed Recall metric of o_i is defined as:

$$BCubed_{Recall}(o_i) = \frac{\sum_{o_j \in H(o_i)} MR(o_i, o_j)}{|H(o_i)|}$$

The overall BCubed Precision of the clustering solution, denoted as $BCubed_{Precision}$, is computed as the average of the BCubed precision of all PEs in the distribution O; the overall BCubed recall of the clustering solution, denoted as $BCubed_{Recall}$, is defined analogously but using the BCubed recall of all PEs. Finally, the FBCubed measure of the clustering solution is computed as the harmonic mean of $BCubed_{Precision}$ and $BCubed_{Recall}$ as follows:

$$FBCubed = \frac{2 \times BCubed_{Precision} \times BCubed_{Recall}}{BCubed_{Precision} + BCubed_{Recall}}$$

4.2 Results and discussions

Our implemented tool has derived features quickly from the used case studies in a reasonable amount of time (about 1 second). Table 1 summarizes the obtained results. For the sake of readability, we manually associated feature names to clusters, based on their content. Of course, this does not impact the quality of our results.

Firstly, we observe that the F values obtained in the Mobile media case study have been greatly influenced by the recall values. The precision values however, remain high for the majority of the features, indicating the relevance of their composing PEs.

These observations can be explained by the operating mechanism of OclustR, which produces partitions having strong cohesion and low overlap. Indeed, we assumed that the features are simulated using one or more classes at the code level. Considering this assumption, implementations of Mobile Media features are therefore strongly overlapped. Indeed, many classes are shared by the implementations of numerous features at the same time, because of the use of several design patterns by the Mobile Media authors such as *Model-Vue-Controller (MVC)* and *Chain of responsibility*. For example, the classes *PhotoController*, *ImageAccessor* and *ImageData* encapsulate all the photo management methods. Thus, features such as *send picture*, *sorting* and *add picture* share most of their PEs (i.e. classes) and, therefore, have reached a low Recall and F values. The *Splash screen* feature however, has not suffered from this problem since all functionalities related to this feature were encapsulated in distinguishable classes. Thus, the *splash screen* feature obtained a maximum F value.

Feature	Evaluation metrics		
	P %	R %	F %
Drawing Shapes (version 5)			
Core	100	57	73
Drawing Text	100	100	100
Drawing Oval	100	100	100
Drawing Line	100	100	100
Drawing Rectangle	100	100	100
Drawing Image	100	100	100
Drawing Arc	100	100	100
Drawing 3D-Rectangle	11	33	17
Mobile Media (version 6)			
Splash screen	100	100	100
SMS transfer	64	41	50
Photo management	100	35	52
Album management	40	22	29
View photo	54	58	56
Edit photo Label	70	39	50
Add photo	100	9	17
Delete photo	40	40	40
Favourites	100	8	15
Sorting	100	8	15
Add album	67	22	33
Delete album	38	38	38
Send photo	100	14	25
Receive photo	40	17	24

Table 1: Features mined from Mobile Media and Drawing Shapes softwares.

The conclusions drawn from the Mobile Media analysis comply with those obtained for the Drawing Shapes case study. The Drawing Shapes features are slightly overlapped, so we reached a maximum *F* value, except for the *3D-Rectangle* and *Core* features. The low F value of *3D-Rectangle* is caused by the low cohesion between its classes in the source code level. In fact, the manual verification of the *3D-Rectangle* source code revealed that the Drawing Shapes author has intentionally or accidently caused this low cohesion by creating *3D-Rectangle* classes without creating dependencies between them (i.e. classes instantiations are missing). Thus, the PEs involved in the implementation of *3D-Rectangle* were scattered throughout the results, so that a low *F* value was reached. However, in the case of *Core* feature, some relevant PEs were mined and mistakenly mapped to another cluster so that the F value was slightly affected. Anyway, the F-measure for this latter is 0.73 on average which is an acceptable value.

The evaluation of overlapping clustering solution for each of the case studies using the FBCubed metric clearly confirms the above findings (see Table 2). The overall BCubed Precision of the Mobile Media case study remains high and the low value of the Bcubed Recall affected consequently the global FBCubed value. In the other hand, balanced and high values were reached for the Drawing Shapes' Bcubed metrics.

Software	BCubed Precision	Bcubed Recall	FBCubed
Drawing Shapes	80%	67%	73%
Mobile Media	78%	25%	38%

Table 2: The BCubed evaluation results of the overlapping clustering solutions.

Despite F-measure and FBCubed proved to be appropriate, our implemented tool generates a small number of clusters most of which are relevant. This latter characteristic makes our results easily understandable and effectively handleable by the user. Table 3 shows that our tool generated 50% of relevant clusters for Drawing Shapes case study, and 88% for Mobile Media.

Software	Mined clusters	Relevant clusters	Relevance ratio
Drawing Shapes	16	8	50%
Mobile Media	16	14	88%

Table 3: Number of reliable mapping.

The results show that our proposed approach has generated a reasonable number of features with an acceptable precision. According to the case studies that we have conducted, our proposed method operates efficiently when dealing with programs having flat FMs. Unlike documentation-based approaches [15, 26, 33], our proposed approach relies on source code as it seems to be the most reliable source that can capitalize on the knowledge and the expertise of experts who participated in the development of the analyzed systems. Since they consist of sets of program elements, the features generated by the proposed approach are of a more formal nature which, thereby, facilitates their interpretation and further manipulation.

The proposed approach provided a feature catalogue instead of generating one preferred hierarchy. This latter characteristic, together with the formal nature of the generated features, represents the key strength of our proposed approach, so it can operate in a generic and efficient manner. Thus, results obtained by our proposed method can be manipulated by other complementary tools in order to get additional information and, therefore, construct a reliable FM. Our proposed method is also complementary to other approaches such as software transplantation [4]. Inspired from human organ transplantation, this latter works by isolating the code of a useful feature in a "donor" program and transplanting this "organ" to the right "vein" in software lacking the feature. Our proposed approach can act in such case by delimiting and extracting a feature before its transplantation.

In addition, our proposed method addressed the information loss problem that characterizes most of FCA-based methods by the usage of a similarity measure. This latter can be tuned by the user to change the granularity of outputs. Compared to other clustering approaches [22, 23], our proposed approach used a new partitioning algorithm that provides overlapping clusters in an efficient manner. The user involvement was negligible during all the steps of our experimentation. Hence, the method can be potentially very useful and it can save stakeholder from a lot of effort and time required to specify features composing each software variant during the SPL reverse engineering task.

4.3 Threats to validity

There is a limit to the use of Floyd's algorithm to infer similarity between PEs. In fact, the complexity determined by this algorithm is of $O(n^3)$ [18]. In addition, precomputing all the shortest paths and storing them explicitly in a huge dependency matrix seems to be challenging in terms of space complexity. These two factors affect the applicability of the proposed approach on larger software systems. In fact, even if computing shortest paths is a well-studied problem, exact algorithms cannot be adopted for a massive dependency graph.

Moreover, as illustrated above, the OclustR algorithm manages clusters' overlapping but still represents several restrictions when dealing with clusters that are strongly overlapped, which limits the usability of the proposed approach to systems with a nested FM. Another problem related to OclustR that may affect the results accuracy happens when a given class, i.e. an abstract class, is inherited by most of the system classes and, thus, will be considered as the center c of a ws-graph ($G_c^\star$) having all the inheriting classes as satellites. Hence, during the improvement phase, each ws-graph having as center one of the $G_c^\star$ satellites will be judged as irrelevant. In this case, we call $G_c^\star$ a *predatory ws-graph* and his center c a *predatory center*. Such predatory cluster phenomena may affect the results accuracy.

Finally, structural distance-based measure used in the proposed approach still has some restrictions. Indeed, we used a simple technique to compute similarity between PEs based on the number of steps on the shortest path relating them in the graph. Even that such a strategy has given acceptable results, it still has some limitations since it does not consider the multiplicity in paths (i.e. connectivity) between a pair of nodes.

5 Conclusion and perspectives

In this paper, we proposed a new method for reverse engineering software functional features from source code. We used dependencies that exist between program elements at the source code level in order to apply a graph clustering algorithm in an efficient way. We tested our implemented tool to recover features from source code of two existing java programs. We obtained promising results that are consistent with the main objectives of our study, which makes the proposed approach useful for mining features from software source code.

In future work, we would like to improve output quality using other overlapping clustering techniques, in order to overcome the aforementioned OclustR limitations. We also plan to automatically extract mined feature names, based on features contents, in order to facilitate their interpretation and manipulation in further tasks.

Furthermore, in order to tackle the complexity problem when computing structural distance-based measures, we plan to use approximation methods based on random walks [1], such as random walk with restart. Besides complexity optimization, random walk-based measure provides a result that is different from that of the shortest-path measure because the multiplicity in paths between a pair of nodes is also leveraged when computing similarity. Such a measure is likely to enhance accuracy of our results and to reduce the effects of the predatory clusters phenomena.

Moreover, since software features are associated with its behavior, we intend to enrich input data using dynamic information. Indeed, even if they are based on different operating strategies, dynamic and static analyses can be complementary in certain points [28]. Hence, a dynamically collected data is likely to enhance the result set by additional information.

6 References

[1] C. C. Aggarwal (2015), "*Similarity and Distances*", *Data Mining: The text book*, pp. 63-91: Springer International Publishing.

[2] R. Al-Msie'deen *et al.* (2014), "*Automatic Documentation of [Mined] Feature Implementations from Source Code Elements and Use-Case Diagrams with the REVPLINE Approach*", International Journal of Software Engineering and Knowledge Engineering, vol. 24, n°. 10, pp. 1413-1438.

[3] E. Amigó *et al.* (2009), "*A comparison of extrinsic clustering evaluation metrics based on formal constraints*", Information Retrieval, vol. 12, n°. 4, pp. 461-486.

[4] E. T. Barr *et al.* (2015), "*Automated software transplantation*", in International Symposium on Software Testing and Analysis Baltimore, MD, USA, pp. 257-269.

[5] Y.-L. Chen, and H.-L. Hu (2006), "*An overlapping cluster algorithm to provide non-exhaustive clustering*", European Journal of Operational Research, vol. 173, n°. 3, pp. 762-780.

[6] P. Clements, and L. Northrop (2001), "*Software product lines: practices and patterns*", Addison-Wesley.

[7] J. Dietrich *et al.* (2008), "*Cluster analysis of Java dependency graphs*", in Proceedings of the 4th ACM symposium on Software visualization, Ammersee, Germany, pp. 91-94.

[8] K. Draszawka, and J. Szymański (2011), "*External Validation Measures for Nested Clustering of Text Documents*", *Emerging Intelligent Technologies in Industry*, Studies in Computational Intelligence pp. 207-225: Springer Berlin Heidelberg.

[9] H. Eyal-Salman, A.-D. Seriai, and C. Dony (2013), "*Feature-to-Code Traceability in Legacy Software Variants*", in 39th EUROMICRO Conference on Software Engineering and Advanced Applications Santander, Spain, pp. 57-61.

[10] S. Ferber, J. Haag, and J. Savolainen (2002), "*Feature Interaction and Dependencies: Modeling Features for Reengineering a Legacy Product Line*", in Proceedings of the Second International Conference on Software Product Lines, pp. 235-256.

[11] E. Figueiredo *et al.* (2008), "*Evolving Software Product Lines with Aspects: An Empirical Study on Design Stability*", in 30th International Conference on Software Engineering, Leipzig, Germany, pp. 261-270.

[12] D. Fisher (1987), "*Knowledge Acquisition Via Incremental Conceptual Clustering*", Machine Learning, vol. 2, n°. 2, pp. 139-172.

[13] R. W. Floyd (1962), "*Algorithm 97: Shortest path*", Communications of the ACM, vol. 5, n°. 6, pp. 345.

[14] B. Graaf, S. Weber, and A. van Deursen (2006), "*Migrating supervisory control architectures using model transformations*", The 10th European Conference on Software Maintenance and Reengineering. pp. 153-164.

[15] E. N. Haslinger, R. E. Lopez-Herrejon, and A. Egyed (2011), "*Reverse Engineering Feature Models from Programs' Feature Sets*", in 18th Working Conference on Reverse Engineering, Limerick, Ireland, pp. 308-312.

[16] K. Kang *et al.* (1990), *Feature Oriented Domain Analysis (FODA) Feasibility Study*, Report CMU/SEI-90-TR-21, Carnegie-Mellon University Software Engineering Institute, United States.

[17] C. Kästner, M. Kuhlemann, and D. Batory (2007), "*Automating feature-oriented refactoring of legacy applications*", in ECOOP Workshop on Refactoring Tools, pp. 62-63.

[18] S. Khuller, and B. Raghavachari (2009), "*Basic graph algorithms*", *Algorithms and Theory of Computation Handbook, Second Edition, Volume 1*, Chapman & Hall/CRC Applied Algorithms and Data Structures series: Chapman & Hall/CRC.

[19] C. W. Krueger (2002), "*Easing the Transition to Software Mass Customization*", *Software Product-Family Engineering : Revised Papers from the 4th International Workshop on Software Product-Family Engineering*, Lecture Notes in Computer Science pp. 282-293: Springer Berlin / Heidelberg.

[20] F. Loesch, and E. Ploedereder (2007), "*Restructuring variability in software product lines*

using concept analysis of product configurations", Proceedings of 11th European Conference on Software Maintenance and Reengineering CSMR '07. pp. 159-168.

[21] B. Meyer (1985), "*On Formalism in Specifications*", IEEE Software, vol. 2, n°. 1, pp. 6-26.

[22] N. Niu, and S. Easterbrook (2008), "*On-Demand Cluster Analysis for Product Line Functional Requirements*", Proceedings of 12th International Software Product Line Conference SPLC '08. pp. 87-96.

[23] P. Paskevicius *et al.* (2012), "*Automatic Extraction of Features and Generation of Feature Models from Java Programs*", Information Technology and Control, vol. 41, n°. 4, pp. 376-384.

[24] A. Pérez-Suárez *et al.* (2013), "*OClustR: A new graph-based algorithm for overlapping clustering*", Neurocomputing, vol. 121, pp. 234-247.

[25] R. Al-Msie'Deen *et al.* (2013), "*Mining Features from the Object-Oriented Source Code of Software Variants by Combining Lexical and Structural Similarity*", in IEEE 14th International Conference on Information Reuse & Integration, Las Vegas, NV, USA, pp. 586-593.

[26] A. Rashid, J. C. Royer, and A. Rummler (2011), "*Aspect-Oriented, Model-Driven Software Product Lines: The AMPLE Way*", Cambridge University Press.

[27] U. Ryssel, J. Ploennigs, and K. Kabitzsch (2011), "*Extraction of feature models from formal contexts*", in Proceedings of the 15th International Software Product Line Conference, Volume 2, Munich, Germany, pp. 1-8.

[28] E. Stroulia, and T. Systä (2002), "*Dynamic analysis for reverse engineering and program understanding*", ACM SIGAPP Applied Computing Review, vol. 10, n°. 1, pp. 8-17.

[29] L. P. Tizzei *et al.* (2011), "*Components meet aspects: Assessing design stability of a software product line*", Information and Software Technology, vol. 53, n°. 2, pp. 121-136.

[30] P. Tonella, and A. Potrich (2007), "*Reverse Engineering of Object Oriented Code*", Springer-Verlag New York, 1 ed.

[31] S. Warshall (1962), "*A Theorem on Boolean Matrices*", Journal of the ACM (JACM), vol. 9, n°. 1, pp. 11-12.

[32] T. J. Young (2005), "*Using aspectj to build a software product line for mobile devices*", Master Thesis, The University of British Columbia.

[33] T. Ziadi *et al.* (2012), "*Feature Identification from the Source Code of Product Variants*", Proceedings of 16th European Conference on Software Maintenance and Reengineering (CSMR). pp. 417-422.

PrefWS3: Web Services Selection System Based on Semantics and User Preferences

Rohallah Benaboud
Department of Mathematics and Computer Science, University of Oum El Bouaghi, Algeria
LIRE Laboratory, University of Constantine 2 - Abdelhamid Mehri, Constantine, Algeria
E-mail: r_benaboud@yahoo.fr

Ramdane Maamri and Zaidi Sahnoun
LIRE Laboratory, University of Constantine 2 - Abdelhamid Mehri, Constantine, Algeria
E-mail: rmaamri@yahoo.fr, sahnounz@yahoo.fr

Keywords: semantic web service discovery, domain ontology, OWL-S, QoS, user preferences, reputation

Received: January 21, 2016

With the growing number of web services on the Web, many approaches have been proposed to help users discover and select the desired services. Nevertheless, finding the best service that meets the user needs and preferences is still a problem. In this article, we introduce a user preferences based semantic web services discovery and selection system (PrefWS3). PrefWS3 is considered to be a user-centric system which helps users in formulating their requirements and preferences. This system involves semantic enhancement of both request and web services and provides an efficient semantic-based matching mechanism, which calculates the semantic similarity between the request and the web service. Furthermore, PrefWS3 includes a QoS-aware process and provides a reputation mechanism that enables users to evaluate the credibility of the web services they use. In this article, we also present the results of a comparison of the PrefWS3 and some other published approaches to evaluate its effectiveness.

Povzetek: Prispevek obravnava izbiro spletnih storitev na osnovi semantike in preferenc uporabnika.

1 Introduction

Web services have emerged as a key technology for implementing Service Oriented Architectures (SOA), aiming at providing interoperability among heterogeneous systems and integrating inter-organization applications [1]. Web services are designed to be selected via discovery mechanisms. Web Service discovery mechanisms include a series of registries, indexes, catalogues, agent-based and Peer to Peer solutions. The most dominating among them is the Universal Description Discovery and Integration (UDDI) [2] which is essentially based on keywords search on WSDL descriptions of Web services. Simple keyword matching does not capture the underlying semantics of web services [3]. As a result, only the services which have same syntactic description with the user request may be considered for selection. For example, when searching services with the keyword ‘vehicle’, the ones whose descriptions contain the word ‘car’ will not be returned. Thus, the discovery process is also constrained by its dependency up on human intervention in choosing the appropriate service based on its semantics [4].

In order to solve the above-mentioned problem, a variety of conceptual models have been proposed over these past years to add semantics to Web Services descriptions. These include WSDL-S [5], WSMO [6], and OWL-S [7]. These so-called Semantic Web services (SWS) are Web services that are annotated with semantic descriptions. This semantic is made through ontologies; one of the important technologies of the Semantic Web.

The discovery of SWS is mainly based on their functional aspects (Inputs, Outputs, Pre-conditions and effects). However, due to the increasing availability of Web services that offer similar functionalities, other parameters have to be considered during the discovery process, especially user preferences that are expressed in term of constraints on quality of service (QoS), i.e., execution time, cost, reliability, availability, etc.

Several approaches of Web Services discovery have been proposed in the literature; however, finding the best and the right web service that meets user needs and preferences is still a problem. This is due to a number of challenges. Some of them include [4] [8]:

- Descriptions of the vast majority of already existing web services are specified using WSDL and do not have associated semantics.
- From the user’s point of view, expressing a request can be a disturbing burden, because he may not have the required expertise or skills.
- Searching is a simple keyword based search; as a consequence, matching does not capture the underlying semantics of web services.
- Accurate service matchmaking for service discovery can be computationally very expensive.
- Dishonest service provider may advertise fake QoS.

In this paper, we present a complete system for web service discovery and selection named PrefWS3, which is able to cope with most of the challenges mentioned above. The proposed system covers the entire spectrum of tasks from service request to service invocation, and also after service invocation.

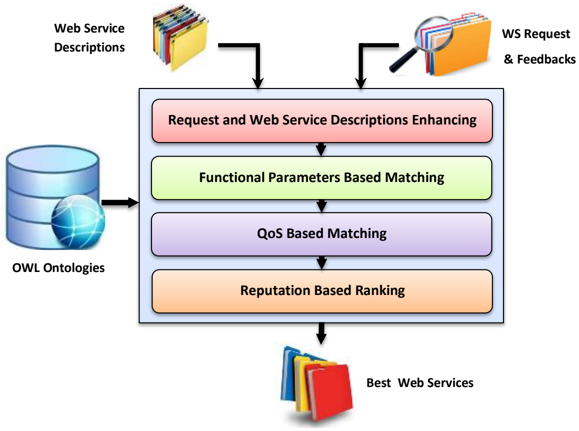


Figure 1: The key steps of PrefWS3.

Figure 1 illustrates the key steps of PrefWS3: The first step involves semantic and QoS enhancing of the request and web service description. The second step deals with the functional parameters based matching of the request against the advertisement services. In the third step, we perform a QoS based matching. In the last step, the user feedback is taken into account for the selection of the best web services. These steps make PrefWS3 a cascading filtering mechanism that finds the best web services from a set of raw web services.

Several approaches have discussed separately the previous four steps, but not all at the same approach. In addition, these approaches differ in the way of each step is implemented. PrefWS3 aims to provide a more "user-centric" system simplifying the service discovery using semantics while satisfying QoS requirements, and to free users from time consuming human computer interactions and Web search. To show the effectiveness of PrefWS3, we compare it with other approaches. The contributions of this paper regarding the different steps can be summarized as follows:

1) Request and web service descriptions enhancing:

- Enhancement of OWL-S profile with QoS information.
- Provide users a way to specify their requirements and preferences expressively and flexibly.

2) Functional parameters based matching: Presenting an efficient matchmaking mechanism that captures semantic similarity between requests and services in a more efficient way with less time.

3) QoS-aware service selection:

- Provide a QoS based filtering mechanism that aims to filter out services that do not meet the service user preferences.
- Introduce a QoS monitoring mechanism that aims to measure the QoS values in order to verify whether the measured values comply with QoS values published by the Web service provider.

4) Reputation based ranking: Provide a mechanism that gives confidence to a user when selecting a web service.

In a previous paper [9], we have addressed some aspects of the PrefWS3 system. The present paper extends the last one by introducing new important mechanisms such as WSLD2OWLS translating, QoS weights calculating, and QoS monitoring mechanisms. Furthermore, the ontology-based OWL-S extension, the QoS based services filtering and the reputation mechanisms, which are previously addressed, are extended in order to make the service selection process more accurate and practical.

The rest of the paper is organized as follows: We present the related works in Section 2. Section 3 gives an overview of the proposed system, and section 4 provides a detailed discussion on request and web services descriptions enhancing. The detailed description of functional parameters based matching is presented in Section 5. Section 6 and 7 include a discussion on QoS based matching and Reputation based ranking respectively. The evaluation of the proposed system is presented in Section 8. Finally, a conclusion and future work are presented in Section 9.

2 Related works

Researches in Web services discovery have been necessary since the number of available services on internet has increased and the user gets tired to find desired service. In this section, we present and analyze the related works in order to comprehend the benefits that may be obtained and to put our contributions in the context of service web discovery.

Most current approaches for web service discovery depend on the measurement of the similarity degrees between service request and service advertisement. The work in [10] presents a matchmaking algorithm which compares input and output concepts of the user request to the service description and defines four levels of matching: Exact, Plug in, Subsumes, and Fail. However, the use of such discrete scale classification of matching is not sufficient to best rank services. Some of the relevant services might be eliminated due to not fitting those discrete scales. PrefWS3 calculates the total similarity score of the web services according to their relevancy to the user request. OWL-S matchmakers are the mainstream in contemporary SWS matchmakers [11]. iSeM [12] performs structural matching between the signatures of a given Web service and request using the logic-based input/output concept matching, the text similarity-based approach, the ontology-structure-based approach, and the SVM-based approach and, after that, adjusts its aggregation and ranking parameters using machine learning. iMatcher2 [13] combines the SPARQL query language for logical referencing and the syntactic similarity measure to calculates the degree of semantic matching between two OWL-S service profiles. OWLS-MX3 [14] takes into account the shortest distance and the common parent classes between the concepts in an ontology to compute the semantic similarity between input/output concepts of service and requests.

The work in [15] introduces a Semantic Advanced Matchmaker (SAM), which provides ranking and scoring based on concept similarity. The authors created their own similarity distance ontologies to find the distance between objects. This ontology is supposed to contain proper similarity scores through the assignment of concept-similarity ratings of all the concepts in the ontology by a similarity ranking mechanism. They perform the matchmaking considering the input/output interface of services. In [16], the authors present a semantic matching approach for discovering semantic web services through a broker-based semantic agent (BSA). The BSA performs semantic matching according to the concepts meanings, the concepts similarities, and distance of concept relations. The semantic distance calculation is based on subsumption-based similarity and hasSynonym, hasIsa relationships. Against the two latter works, PrefWS3 don't use only the subsumption relationships between concepts to calculate their similarity but it also takes into account common properties between them. Additionally, the semantic distance between ontology concepts is not necessarily determined according to the distance between concepts. Two concepts that are directly attached may be semantically very different. This case may take place when a concept extends another one by introducing several new properties. In the paper [17], the authors present the application of Case-based Reasoning (CBR) to the problem of service discovery and selection by introducing a case representation, learning heuristics and different similarity functions. The proposed approach combines notions of CBR with the use of WordNet as lightweight semantic basis. The major disadvantage of CBR is that users might rely on previous experience without validating it in the new situation [18]. This is clearly a problem in changing web services functionalities where past descriptions may not reflect current descriptions. In addition, the system requires a large memory space to store all the previous cases in the form of problem-solution pairs. Semantic web service technology is already adopted in several web based applications and solutions, the authors of [19] propose an intelligent system in order to facilitate semantic discovery and interoperability of Web Educational Services that manage and deliver Web media content.

Unlike the aforementioned matchmakers, the matchmaking mechanism of PrefWS3 takes into account the role of concepts in the request and the web service, i.e, concepts are inputs or outputs, to calculate the degree of similarity between them. We think that an output in the request should not be considered as similar to a more generic output in the advertised service, and an input in the advertised service should not be considered as similar to a more generic input in the request.

Since there are many functionally similar Web Services available in the Web, it is an absolute requirement to distinguish them using a set of non-functional criteria such as Quality of Service (QoS). The work in [20] presents a QoS-based model for web service discovery by extending the UDDI's data structure types in order to enhance UDDI model with QoS information. However service discovery and selection are still done by human consumer. Furthermore, this approach is impractical with a huge number of Web services available for selection. The authors of [21] propose a QoS-based web service selection system that handles QoS requests with both exact values and fuzzy values, return two categories of matching offers: super-exact and partial matches. In [22], users' preferences are defined by a lexical ordering in accordance with their perceived importance. They presented an algorithm to compare web services based on their qualities (QoS). According to the proposed algorithm, a web service WS1 is considered better than a web service WS2 if the first QoS attribute that distinguishes between WS1 and WS2 ranks WS1 higher than WS2. This algorithm is simple, but if the user indicates that tow preferences are equally important, the algorithm will take into account only the first preference in the lexical ordering and ignores the second preference. In our system, the use of the weighted sum method allows to take into account all preferences each with its importance degrees. The authors of [23] present a web service selection framework, which takes into account implicit preferences that are inferred from information related to context and profile of the user. Similarities between different attributes of service request and service are captured thanks to fuzzy-set-based techniques. It is argued that augmenting the user's query by preferences depending on their context and their profile allows for highly improving the result's quality. However, the proposed framework requires too much information and a large number of fuzzy rules to infer implicit preferences in each specific domain. Moreover, the proposed framework doesn't take into account the importance of each QoS.

A major problem in using QoS for service discovery is the specification and storage of the QoS information. Most of QoS-aware discovery mechanisms, described above, ignore this problem. In our work, we propose an ontology-based OWL-S extension to add QoS to OWL-S descriptions. Furthermore, PrefWS3 proposes a QoS-based filtering mechanism which filters out services that do not meet the user preferences described as QoS constraints. This filtering mechanism takes into account the degree of confidence that the user has on the specified constraints.

Some approaches already exist about involving the user in the process of service discovery. Those approaches are variants of reputation systems in which the users rate the service providers and share these ratings with other users. The work in [20] presents a reputation-enhanced model that contains a reputation manager which assigns reputation scores to the services based on user feedback regarding their performance. Then, a discovery agent uses the reputation scores for service matching, ranking and selection. The authors of [24] introduce a Web service selection mechanism based on user ratings and collaborative filtering. Services are ranked based on similarity to the user's ratings from the collected feedback database from the users. The similarity mechanism is calculated based on Pearson correlation coefficient. A Bayesian network trust and reputation model for web services is introduced in [25], which considers several factors when assessing web services' trust: direct opinion from the truster, user rating (subjective view) and QoS monitoring information (objective view). In [26], the authors present a QoS-based semantic web service selection and ranking solution with the application of a trust and reputation management method, which detects and deals with false ratings by dishonest providers and users.

In most of the aforementioned reputation mechanisms, the satisfaction criterion of the rater is unknown since the service user gives one rating score for all QoS of the invoked service. Without knowing the intention of the rater, it is almost impossible to make a given rating meaningful. In the reputation mechanism of PrefWS3, the service user gives a rate score for each QoS attribute of the used Web services. Furthermore, PrefWS3 gives more importance to recent ratings.

3 PrefWS3 overview

PrefWS3 aims to provide a complete system for web service selection that simplifies the service discovery using semantics while satisfying the user preferences. PrefWS3 covers the entire discovery process through several components that take charge of the request and the web services descriptions process, functional based matchmaking, filtering and matching of quality of service parameters, and reputation and rating mechanism. There are four types of data needs to be collected in PrefWS3 to deal with the service request: Web service description in both WSDL or OWL-S files, public open OWL ontologies on the Internet, QoS data, and ratings.

Figure 2 illustrates the main components of the PrefWS3 system, which are described as follows:

Interface Management: This module manages the different interactions with the system users. Depending on the nature of the user and type of his request, the required scenario occurs. These requests include the following parts:

- Requests from service providers in order to register their web services.
- Requests from service consumers, which can be either web services lookup requests or service ratings.

WSDL to OWL-S translating: A large number of service providers use WSDL based syntactic description to describe their services. Therefore in our system, we use a translator to translate WSDL files into OWL-S files and provide semantically enriched description. The translating mechanism uses domain ontologies for mapping complex types, inputs, and outputs of WSDL into OWL ontology concepts.

OWL-S Extending with QoS Information: We use OWL-S service profile as a model for semantic annotation of Web service descriptions. However, OWL-S mainly focuses on describing functional aspects of a Web service and does not describe QoS aspects. After the translation from WSDL files to OWL-S files, the OWL-S profile must be enhanced with QoS information to enable selecting the best services that meet user preferences.

OWL-S repository: This component is responsible for storing the semantic service descriptions as OWL-S files, which could be used for service discovery process.

QoS Weight Calculation: Service consumers have different preferences. For example, a service consumer may want a Web service with lower cost while for another one; the execution time could be his most important parameter. For this raison, we propose that the service consumer may specify that a QoS attribute is more important than another one. Indeed, a weight is given for each QoS attribute. Weights are in [0, 1] where higher weights represent greater importance. Because weights are an important factor for determining the overall quality of a Web service, we calculate the weights for each QoS attribute according to the service consumer QoS preferences. However, distributing the weight of many QoS attributes overburdens the service consumer. Weights calculation can be consider as a multiple decision criteria problem and therefore, we can apply an Analytic Hierarchy Process (AHP) which becomes one of the best known and most widely used multi-criteria decision making methods [27]. By applying this method, the system can easily calculate the weights because it requires only a simple evaluation between two QoS attributes.

Request Generating: In PrefWS3, a request is described in the same manner as a service to facilitate the matching of their descriptions. Request input and output parameters are assumed to be mapped to concepts from domain ontology. Therefore, when a service consumer wants to insert his request, an Ontology-Guided Interface is offered and the service consumer must select the desired terms he wants to use in his request from the list of terms provided in a pop-up by the interface.

Domain Ontologies: Service and request are described using relevant ontology concepts. Different domains may need different ontological representations. Therefore, to avoid the semantic heterogeneity due to the use of different concepts, we use a common ontological basis which contains comprehensive ontologies of different domains of Web services development. Input and output parameters of both request and service are defined using concepts from the same domain ontology.

OWL Public Ontologies: Several websites provide open public ontologies such as DAML Ontology Library[1], which contains more than 280 ontologies written in OWL or DAML+OIL. OWL public ontologies are used to create new ontologies or update already existing one in the domain ontologies repository. These public ontologies will be consulted periodically to develop or enrich the domain ontologies.

[1] www.daml.org/ontologies/

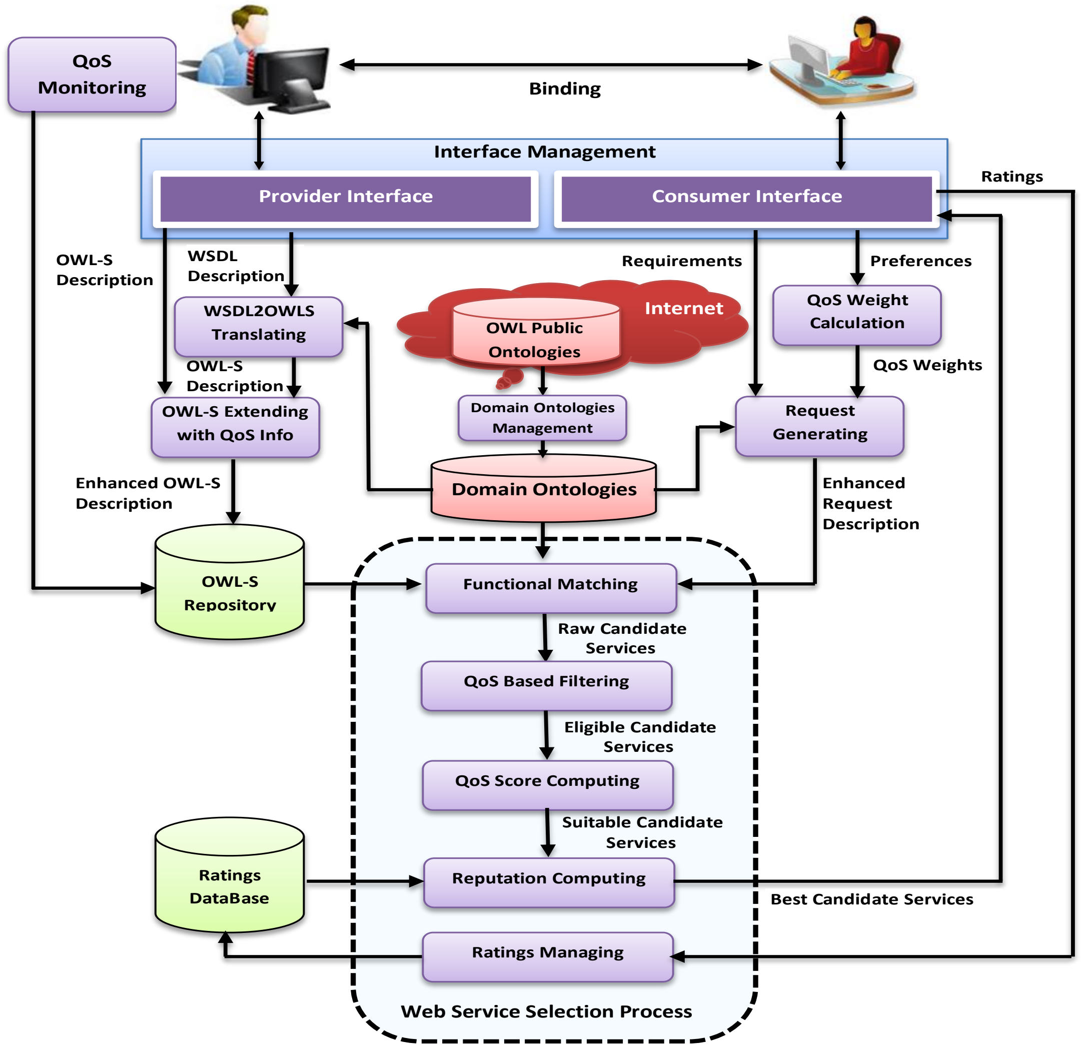


Figure 2. PrefWS3 architecture.

Domain Ontologies Management: This component takes charge of maintaining and updating domain ontologies with additional entities. The main challenges in updating domain ontologies are: 1) finding new information (concepts, relationships…), and 2) incorporating the new information in ontologies.

Functional Matching: The web service input and output parameters contain the underlying functional knowledge that is extracted for improving functional service discovery. The main concept of service discovery is semantic-based matching between requests and services. It establishes a mapping between the input of the request and the input of the service and a mapping between the output of the request and the output of the service.

QoS Based Filtering: Sometimes, the service consumer indicates that he refuses a Web service with a QoS having a value below or above a threshold specified in his query. QoS based filtering aims to filter out services that do not meet the service consumer preferences described as QoS constraints.

QoS Score Computing: The role of the QoS score computing step is to find the degree of quality of the candidate services after completing the functional matching through their QoS metric information and user preferences.

QoS Monitoring: The QoS monitoring mechanism aims to monitor and measure the QoS values of services in order to verify whether the measured values are in compliance with QoS values published by the Web service provider. QoS monitoring becomes the determining factors for customers to whether continue using the service or not [28].

Ratings Managing: Once the web service is selected, the service user should provide a rating score to show his satisfaction level of the invoked web service. The reputation mechanism enables service consumers to evaluate the credibility of web services they use, and takes into account the satisfaction criteria of each service consumer.

Ratings Database: User ratings are stored in an RDF triple store and kept in Ratings database.

Reputation Computing: The reputation of a service is a collective measure of the opinion of a community of users regarding their experience with the service [29]. It is computed as an aggregation of users' feedbacks and kept in Ratings Database. Web services are ranked using their reputation and as a final step, best ranked candidate services are shown to the service consumer.

4 Web service and request model

Typically, Web services are described using functional and non-functional properties. Functional properties represent the description of the service functionalities. In our work, functional properties contain Service Name, Textual description, a set of Inputs and a set of Outputs. Non-functional properties represent the description of the service characteristics (e.g. QoS). Generally, QoS may cover a lot of attributes hosted by different roles. In this paper, we adopt three key attributes that the service customers mostly care about when they use a Web service. These are: Execution time, Execution price, and Reliability. Note that other QoS attributes can be applied to our system without fundamental modification.

A request signifies a service demand. A request description includes functional and non-functional requirements. The former describes the functional characteristic of the service demand, such as inputs and outputs. The latter mainly focuses on the customer's preferences, namely quality of service (QoS). In our work, a service consumer doesn't have to give the value of each desired QoS attribute; he should get instead the means to specify that a QoS attribute is more important than another one.

4.1 Translation from WSDL to OWL-S descriptions

The WSLD2OWLS translating component of PrefWS3 system translates WSDL files of the already existing Web Services into a semantic definition using OWL-S. This translation aims to add semantic annotations to Web Service specifications. In PrefWS3 system, we use only the OWL-S service profile in the discovery mechanism, so that the translator is responsible for translating WSDL files into OWL-S service profile files. Much research work has been done in mapping WSDL to OWL-S [30] [31] [32], but it is important to note here that until nowadays, the mapping process is not functioning fully automatically. The main raison is that the OWL-S description contains more information than the WSDL description. WSDL description provides only input and output information, while OWL-S description can provide inputs, outputs, preconditions and effects. Therefore this additional information must be set manually. In our work, we are limited to use only service name, textual description, inputs and outputs to functionally describe a web service. Because that information are already provided by WSDL description, the translator process is fully automatic.

The benefit of describing Web services in WSDL format is that WSDL is machine-readable, namely it can be parsed automatically. WSDL description contains service name, service textual description, types, inputs, outputs and binding. Based on the mapping process presented in [31], we can summarize our translator mechanism, as depicted in Figure 3, as follows:

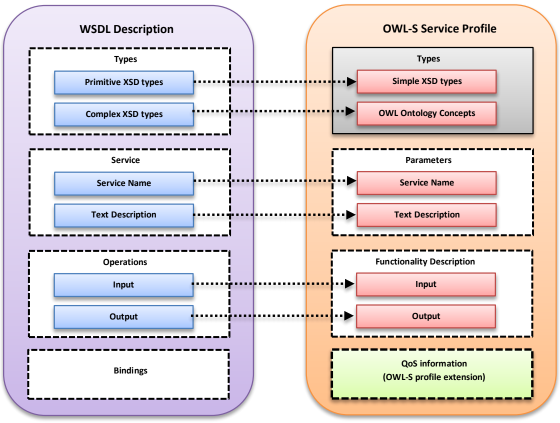


Figure 3: Translation from WSDL to OWL-S descriptions.

- Service name in WSDL is translated into service name in OWL-S service profile.
- Service textual description in WSDL is translated into Service textual description in OWL-S service profile.
- Primitive XSD types in WSDL are translated into XSD simple types.
- Complex XSD types in WSDL are translated into OWL ontology concepts.
- Inputs and outputs of WSDL are mapping to OWL ontology concepts.

4.2 Embedding QoS properties in the OWL-S service profile

PrefWS3 system uses OWL-S service profile as a model for semantic annotation of Web service descriptions. However, OWL-S mainly focuses on describing functional aspects of a Web service and does not describe QoS aspects. Many approaches based on ontologies have been proposed for QoS [33] [34]. However, existing approaches are difficult for users to define their QoS based preferences. They usually assume that users could formulate their preferences easily and are accurately using the QoS languages.

Based on our previous work [9], we propose an ontology based OWL-S extension to add non-functional description, referred to as QoS, to Web service description. The new service profile model is depicted in Figure 4. In OWL-S service profile we use a set of ServiceParameter which has a name (serviceParameterName) and a value (sParameter). For the connection of OWL-S and QoS ontology, the QoSProperty is a subclass of OWL-S ServiceParameter, and QoSParameterName and qosParameter are subproperties of OWL-S ServiceParmaerterName and sParameter property respectively. This method is open to apply any QoS ontologies.

Each QoS property (QoSProperty) is defined by a name (qosParameterName) as a "String" and a set of characteristics (QoSCharacterisitic) that we describe as follows:

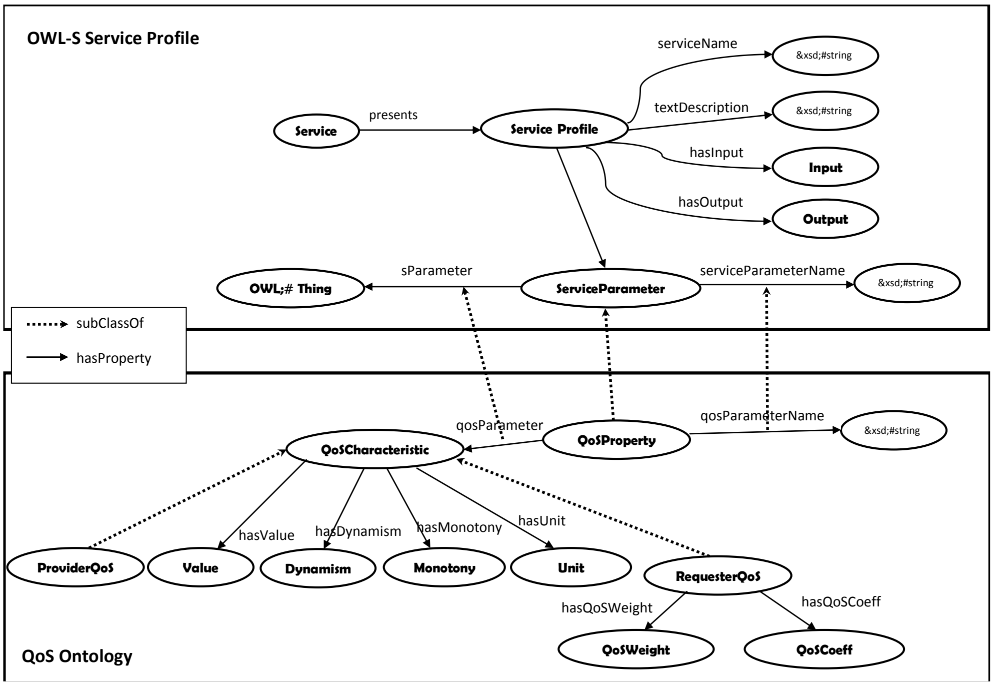


Figure. 4: OWL-S extension to support QoS.

- *Value:* Represents the value of a QoS property. From the provider viewpoint, this value represents the one of a QoS attribute of the provided service; but from the consumer viewpoint, it represents a threshold QoS value.
- *Monotony: This feature is used to distinguish between two types of QoS:*
  - ➢ Quality with increasing monotony (e.g, execution time). In this case, the QoS property value indicated by the service user represents the minimum value to be taken into account.
  - ➢ Quality with decreasing monotony (e.g, execution price."). In this case, the QoS property value indicated by the service user represents the maximum value to be taken into account.
- *Unit*: Each value of the QoS property is provided together with a measuring unit (e.g, Dollars, Seconds)
- *Dynamism*: We distinguish two different types of QoS: The static and dynamic. A static QoS is a quality whose value is known before the Web service execution (eg, the execution price). Dynamic QoS is a quality whose value is known only after the Web service execution (eg, execution time).
- *QoSWeight*: As described in previous section, the QoS weight allows specifying that a QoS property is more important than another one.
- *QoSCoeff*: This coefficient represents the degree of confidence that a customer has on his preference. The use of this coefficient will be detailed in subsection 6.1.

The proposed OWL-S extension is particularly useful for the different actors involved in the publication, the discovery and the invocation of web services, mainly service provider and service consumer (or user). Service provider can enter the services by filling properties values of the different service qualities (ProviderQoS). To facilitate this task, PrefWS3 displays the definition and comments on the quality property whose value must be entered. The service consumer can query the OWL-S extension ontology to find services that best meet their QoS requirements (RequesterQoS).

4.3 QoS weights calculating using AHP method

To help the service user on determining the weights according to their QoS preferences easily, the "QoS weight calculation" component of PrefWS3 uses a mechanism based on an Analytic Hierarchy Process (AHP) method which allows the calculation of weights only by a simple evaluation between two QoS attributes.

The Analytic Hierarchy Process, presented in [35], is a multi-criteria decision-making approach which can be used to solve complex decision problems. Basically, this approach involves the construction of a pair-wise comparison matrix where each element is rated against every other element by means of predefined scores (from 1 to 9) indicating their relative importance as shown in Table 1. These comparisons are used to obtain the weights of importance of the decision criteria. If the comparisons are not perfectly consistent, then it provides a mechanism for improving consistency.

In PrefWS3 system, the main steps for using the AHP method can be described as follows:

1. In the first step, we identify the criteria to be used by the method. As an illustration, we choose three QoS attributes as criteria, which are: execution time, execution price and availability.

2. In the second step, we establish the pairwise matrix based on service user preferences. By applying the AHP method, since we have three QoS attributes, a pairwise comparison matrix, containing nine elements, has been constructed. Suppose that the matrix, depicted in Table 2, represents the corresponding judgments with the pairwise comparisons.

Scales	**Degree of preferences**	**Explanation**
1	Equally	Two activities contribute equally to the objective
3	Moderately	Experience and judgment slightly favor one over the another
5	Strongly	Experience and judgment strongly or essentially favor one activity over another
7	Very strongly	An activity is strongly favored over another and its dominance is shown in practice
9	Extremely	The dominance of one over another is affirmed on the highest possible order
2, 4, 6, 8	Intermediate values	Used to represent compromises between the preferences in weights 1, 3, 5, 7 and 9
Reciprocals	Opposites	Used for inverse comparisons

Table 1: Pairwise comparison scale for AHP preferences. [35]

QoS attribute	**Execution time**	**Availability**	**Execution price**
Execution time	1	9	3
Availability	1/9	1	1/5
Execution price	1/3	5	1

Table 2: Example of a pairwise matrix.

3. In the third step, we calculate the weight of importance of each QoS attribute based on the pairwise comparison matrix and many normalization operations. The weighted values are calculated by Algorithm 1.

4. In the fourth step, we verify the consistency of the service user judgments. In the AHP method, judgments are considered to be adequately consistent if the corresponding consistency ratio (CR) is less than 0.1; otherwise it is necessary to review the subjective judgments. The CR is calculated as follows. First the consistency index (CI) needs to be calculated. This is done by algorithm 2. Next the consistency ratio CR is obtained by dividing the CI value by the Random index (RI) as given in Table 4 where *n* is the number of criteria.

When applying the algorithm on the above example of pairwise comparison matrix, we get the weights presented in Table 3.

Algorithm 1: Weights Calculation

```
Input: C: matrix n × n
// pairwise comparison matrix obtained in step 2.
Output: W: vector with size n // weights vector
Variables: P: matrix n × n initialized with 0 for each element.
S, W: vectors with size n initialized with 0 for each element.
Begin
1:  for j := 1 to n  do
        for i := 1 to n  do
            $S[j] \leftarrow S[j] + C[i][j]$;
2:  for j := 1 to n  do
        for i := 1 to n  do
            $P[i][j] \leftarrow C[i][j] / S[j]$;
3:  for i := 1 to n  do
        for j := 1 to n  do
            $W[i] \leftarrow W[i] + P[i][j]$;
4:  for i := 1 to n  do
        $W[i] \leftarrow W[i]/n$;
End
```

QoS attribute	**Execution time**	**Availability**	**Execution price**
Weight	0.67	0.06	0.27

Table 3: Example of weight scores.

n	1	2	3	4	5	6	7	8	9
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45

Table 4: RI values for different values of n. [35]

When the algorithm 2 is applied to the previous judgment matrix, it can be verified that the following are derived: $\lambda_{max} = 3.056$, CI = 0.028, and CR = 0.048. The CR value is less than 0.10, so weights are accepted.

Cases	Concept A	Concept B	The role of concepts in request/web service	Relationship between Concepts in Domain Ontology	ConceptSim(A, B)
1 (line 1)	Location	Location	/	Location = Location	1
2 (line 3)	PhdStudent	Person	PhdStudent ∈R1.Inputs and Person ∈ S1.Inputs	PhdStudent < Person	1
3 (line 4)	AlgUniversity	University	AlgUniversity ∈ R1.Outputs and University ∈ S1.Outputs	AlgUniversity < University	0,8
4 (line 7)	University	AlgUniversity	University ∈ R2.Outputs and AlgUniversity ∈ S2.Outputs	AlgUniversity < University	1
5 (line 8)	Person	PhdStudent	Person ∈ R2.Inputs and PhdStudent ∈ S2.Inputs	PhdStudent < Person	0,6
6 (line 10)	PhdStudent	Employer	/	PhdStudent <> Employer	0,5
7 (line 11)	Person	University	/	Person ≠ University	0

Table 5: Example of conceptSim calculation.

5 Service functional matching

The main concept of service functional matching is semantic matching between request and web services, namely, inputs and outputs of the request are both matched with the ones of the web service. We consider that all inputs and outputs refer to concepts of domain ontology. In fact, matching inputs (outputs) of the request and the web service is nothing other than the matching of concepts associated to inputs (outputs). To calculate the similarity of two concepts A and B, we take into account two parameters. The first is the relationship between the two concepts in the domain ontology. The second is the role of concepts in the request and the web service, i.e, concepts are inputs or outputs.

Algorithm 2: CI Calculation

```
Inputs: C: matrix n × n   // pairwise comparison matrix obtained in step 2.
W: vector with size n   // weights vector obtained by Algorithm 1
Outputs: CI: float    // Consistency Index
Variables: P: matrix n × n initialized with 0 for each element.
 S: vector with size n initialized with 0 for each element.
 $\lambda_{max}$: float.
Begin
1:  for j := 1 to n  do
          for i := 1 to n  do
            $P[i][j] \leftarrow C[i][j]*W[j]$;
2:  for i := 1 to n  do
          for j := 1 to n  do
            $S[i] \leftarrow S[i] + P[i][j]$;
3:  for i := 1 to n  do
          $S[i] \leftarrow S[i]/W[i]$;
4:  $\lambda_{max} \leftarrow Max(S[1], S[2],\ldots, S[n])$;
5:  $CI \leftarrow (\lambda_{max} - n)/(n - 1)$;
End
```

Based on the relationship between the two concepts A and B in the domain ontology, we distinguish the following scenarios:

- A = B: The concepts A and B are the same or they are declared as equivalent classes.
- A < B: The concept A is a subclass of the concept B directly or indirectly.
- B < A: The concept B is a subclass of the concept A directly or indirectly.
- A <> B: The concept A does not have a parent/child relationship with the concept B, but both concepts have a parent concept C in common directly or indirectly.
- A≠B : Otherwise.

Based on the role of concepts in both request and web service, we think that an output in the request should not be considered as similar to a more generic output in the advertised service, while a request input could be considered as similar to a more generic advertised input. For example, if a user requests a web service that gives as an output the list of “Algerian universities”, then the web service that gives as an output the list of all universities, cannot be considered as a suitable service because; it can return a set of “European universities” that do not interest the user. We think also that an input in the advertised service should not be considered as similar to a more generic input in the request, while an output in the advertised service could be considered as such. For example, if a user requests a web service that takes as an input the ID of a student, then the Web service that takes as an input only the ID of a PHD student cannot be considered as a suitable service, because it ignores a much of the request's inputs.

To calculate the semantic similarity between two concepts A and B, we use the function ConceptSim(A, B). Our definition of this function is based on the constraints described above and on the information theoretic based measure presented in [36]. Semantic similarity is defined as the amount of common information that is shared between the concepts. Algorithm 3 gives the exact definition of the function ConceptSim(A, B), where:

- The concept A annotates an input/output of the request, while the concept B annotates an input/output of the Web Service.
- All inputs and outputs refer to concepts of the domain ontology, an example portion of which is shown in Figure 5.

Algorithm 3 : ConceptSim(A, B)

```
Begin
1:  if A = B then  ConceptSim(A, B) = 1
2:  else if A <  B  then
3:        if A, B are Inputs  then ConceptSim(A,B)= 1
4:        else  if A, B are Outputs then
              ConceptSim(A, B) = $\frac{\text{Size(prop(B))}}{\text{Size(prop(A))}}$  endif
5:        endif
6:     else if B < A then
7:        if A, B are Outputs then ConceptSim(A,B) = 1
8:        else if A, B are Inputs then
              ConceptSim (A, B) = $\frac{\text{Size(prop(A))}}{\text{Size(prop(B))}}$  endif
9:        endif
10:    else if A <> B then
              ConceptSim(A, B) = $\frac{\text{Size(prop(A)}\cap\text{prop(B))}}{\text{Size(prop(A)}\cup\text{prop(B))}}$
11:    else  ConceptSim (A, B) = 0  endif
12:    endif
13:   endif
14: endif
13: return ConceptSim (A, B).
End
```

- The function prop(C) denotes the set of properties of the concept C.
- The function Size(S) denotes the number of elements of the set S.
- If a concept A is a subclass of a concept B (A < B), then all properties of B are added to the properties of A (inheritance property).

Example: For illustration, let us take two requests (R1, R2) and two web services (S1, S2). All inputs and outputs refer to concepts of the domain ontology shown in Figure 5.

- R1: Inputs = { PhdStudent}, and
  Outputs = { Location, AlgUniversity }
- R2: Inputs = { GeographicArea, Person }, and
  Outputs = { University }
- S1: Inputs = { Person }, and
  Outputs = { Location, University }
- S2 : Inputs = { Location, PhStudent }, and
  Outputs = { AlgUniversity}

The different cases can be illustrated in Table 5.

After describing the semantic similarity between concepts, we give now the algorithm of inputs matching (algorithm 4). Where R.Inputs and S.Inputs denote the set of inputs in the request R and the set of inputs in the service S respectively, Card(E) denotes the cardinality of the set E, Sort(A) allow to sort the elements of the array A in descending order. In lines 1, 2, 3 and 4, the algorithm matches each request input with all Web service inputs, and keeps the best mapping for each request input. In lines 9, 10, 11 and 12, it distinguishes between the situation when the number of request inputs is less than the number of service inputs and when the inverse situation is presented. In the first case, we have a miss of information; therefore InputsSim value is decreased (line 10).

The outputs similarity given by OutputsSim(R.Outputs, S.Outputs) function is also calculated, by algorithm 5, in the same way as inputs similarity. But when the number of service outputs is less than the number of request outputs, the value of OutputsSim is decreased. Therefore we inverse line 10

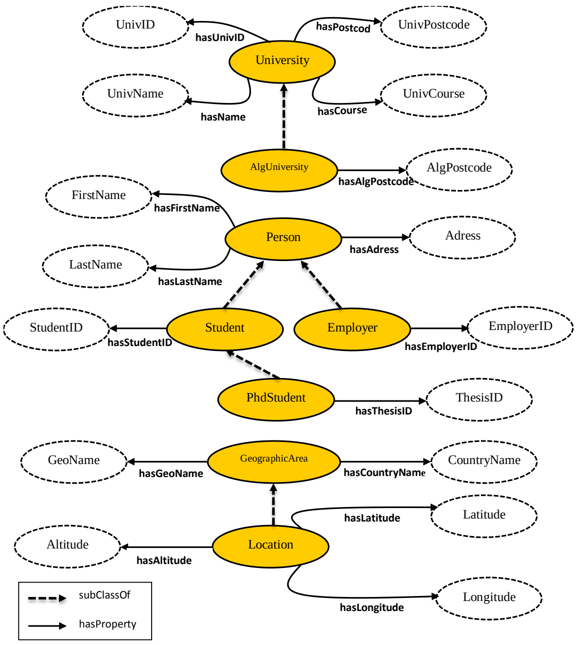


Figure 5: Part of simple Ontology.

with 12 and perform changes in variable names in the algorithm 3.

For example, let us calculate the Inputs and Outputs similarity between Req1 and WSer1 shown previously. InputsSim = ConceptSim(PhdStudent, Person) = 1.

OutputsSim= $\frac{\text{ConceptSim(Location,Location)+ConceptSim(AlgUniversity,University)}}{2}$

$= \frac{1+0,8}{2} = 0,9.$

After calculating inputs and outputs similarity, functional similarity can be calculated using Equation 1. Where weights w_1 and w_2 are real values between 0 and 1 and must sum to 1; they indicate the degree of confidence that the service consumer has in the input similarity and output similarity. By default, w_1 and w_2 are set to 0.5.

FunctionalSim(R, S) = w_1*InputsSim(R.Inputs, S.Inputs) + w_2*OutputsSim(R.Outputs, S.Outputs) (1)

In the previous example, FunctionalSim(R1, S1)= 0.5*1 + 0.5*0.9= 0.95. This value indicates that R1 and S1 are semantically very close.

Algorithm 4 : InputsSim(R.Inputs, S.Inputs)

```
InSim: array of float; // initialized with 0 for each element
Begin
1:  foreach e$_1$ in R.Inputs do
2:      foreach e$_2$ in S.Inputs do
3:          InSim$_i$ = Max(InSim$_i$ , ConceptSim(e$_1$, e$_2$));
4:      end for
5:      i = i + 1;
6:  end for
7:  Sort(InSim);
8:  m = Card(R.Inputs) – Card(S.Inputs);
9:  if m<0 then
10:     InputsSim = $\frac{\sum_{j=1}^{Card(R.Inputs)} InSim_i}{Card(R.Inputs)} /(|m|+1)$
11: else
12:     InputsSim = $\frac{\sum_{j=1}^{Card(S.Inputs)} InSim_i}{Card(S.Inputs)}$
13: end if
14: return InputsSim
End
```

6 The QoS-based matching phase

6.1 QoS based services filtering

Sometimes, the service user indicates that he refuses a Web service with a QoS having a value below or above a threshold specified in his query. For example, a service consumer may want a service with an execution price not exceeding 100 units. So, candidate services which are over this threshold value will be eliminated.

This type of filtering is effective to meet user preferences, but suppose for the previous query, the discovery process has found a good Web service from the functional point of view but offers execution price equal to 101 units. This Web service will be ignored although 1unit may not make a difference to the user. In such case, we propose that when the user indicates a threshold for a QoS attribute, it associates a confidence coefficient "QoSCoeff". This coefficient represents the degree of confidence that the user has on the specified threshold. The value of this coefficient should be in the range [0, 1]. The value 1 means that the filtering algorithm must strictly observe the specified threshold. The value 0 means that the filtering algorithm must ignore this threshold. Therefore, our system uses algorithm 6 as a QoS-based services filtering algorithm. With this algorithm we can avoid the selection of web services that does not meet the service consumer preference.

Algorithm 6 takes as inputs a set of candidate web services and a set of QoS based constraints, thresholds (QoSConstraints.value) and confidence coefficient (QoSConstraints.QoScoeff), then filter out unwanted services taking into account that each QoS attribute can be monotonically increased or decreased.

For each Web service from the candidate Web services, we check the offered QoS properties to compare it with user constraints.

Line 9: If the QoS property is a positive quality (QoSCharacteristic.Monotony = "increase"), then multiply the value of the threshold by the coefficient to further decrease the threshold value.

For example, if the user indicates a threshold equal to 50 units for the execution time QoS property, with a confidence coefficient equal to 0.7, then all Web services with an execution time more than 50 * 0.7 = 35 units are maintained. The others are filtered out.

Line 11: If the QoS property is a negative quality (QoSCharacteristic.Monotony = "decrease"), then we divide the value of the threshold by the coefficient to further increase the threshold value.

For example, if the user indicates a threshold equal to

Algorithm 5 : OutputsSim(R.Outputs, S.Outputs)

```
OutSim: array of float; // initialized with 0 for each element
Begin
1:  foreach e$_1$ in R.Outputs do
2:      foreach e$_2$ in S.Outputs do
3:          OutSim$_i$ = Max(OutSim$_i$ , ConceptSim(e$_1$, e$_2$));
4:      end for
5:      i = i + 1;
6:  end for
7:  Sort(OutSim);
8:  m = Card(R.Outputs) – Card(S.Outputs);
9:  if m<0 then
10:     OutputsSim = $\frac{\sum_{j=1}^{Card(S.Outputs)} OutSim_i}{Card(S.Outputs)}$
11: else
12: OutputsSim = $\frac{\sum_{j=1}^{Card(R.Outputs)} OutSim_i}{Card(R.Outputs)} /(|m|+1)$
13: end if
14: return OutputsSim
End
```

100 units for the execution price QoS property, with a confidence coefficient equal to 0.8, then all Web services with an execution price less than 100/0.8 = 125 units are maintained. The others are filtered out.

Algorithm 6: QoSServicesFiltering(CandidateServices, QoSConstraints)

```
Begin
 foreach service S in CandidateServices do
   foreach QoSParameter in S do
    Coeff := QoSConstraints.QoScoeff
    Mon:= QoSCharacteristic.Monotony
    SVal:= QoSCharacteristic.Value
    RVal:= QoSConstraints.Value
   if (Coeff < > 0) then
     if (QoSParameter.name = QoSConstraints.name)
     then
        if (Mon = "increase" ) and
           (SVal < (RVal × Coeff)) then
             FilterOut (S) from CandidateServices.
      elseif (Mon = "decrease" ) and
           (SVal > RVal / Coeff)) then
               FilterOut (S) from CandidateServices.
       endif.
     endif
    endfor
   endfor
End
```

6.2 QoS score computing

Each QoS value (qosValue) needs to be normalized to have a value in the range of 0 to 1. This step normalizes them in [0, 1] to guarantee they are evaluated by the same span. To normalize the QoS value, we take into account that each QoS attribute is monotonically increasing or decreasing.

	QoS attribute nature	qosMaxValue and qosMinValue
	monotonically increasing	qosMaxValue ≠ qosMinValue
Normalized QoS value	$1 - \frac{qosMaxValue - qosValue}{qosMaxValue - qosMinValue}$	
	monotonically increasing	qosMaxValue = qosMinValue
Normalized QoS value	1	
	monotonically decreasing	qosMaxValue ≠ qosMinValue
Normalized QoS value	$1 - \frac{qosValue - qosMinValue}{qosMaxValue - qosMinValue}$	
	monotonically decreasing	qosMaxValue = qosMinValue
Normalized QoS value	1	

Table 6: QoS value normalization.

Table 6 shows how to normalize QoS value, where qosMaxValue and qosMinValue values show the maximum and minimum values of the QoS attribute between all candidate services.Algorithm 6 takes as inputs a set of candidate services in "CandidateServices " and calculated QoS weights from a service user request and establishes the QoSServices matrix of QoS scores, and gives as output a vector QoSScore which contains the overall QoS score of each candidate Web service. QoSServices is a matrix where rows represent candidate Web services, and columns represent QoS attributes.

Algorithm 7: QoSScoreComputing(CandidateServices, QoSConstraints)

```
MtxServices: Matrix of float ;
QoSScore: Vector of float initialized by <0, 0,……, 0> ;
Begin
1:  foreach service S in CandidateServices do
      begin
2:     foreach QoSParameter in S do
        begin
3:       MtxServices[i, j] : =
           NormalizedValue(QoSCharacteristic.Value);
4:      QoSScore[i] := QoSScore[i] + (MtxServices[i, j] ×
           QoSConstraints.QoSWeight);
         j:= j +1;
        endfor
5:    i:= i +1;
      end for
End
```

In line 3, we calculate for each candidate Web service the normalized value of each QoS attribute.

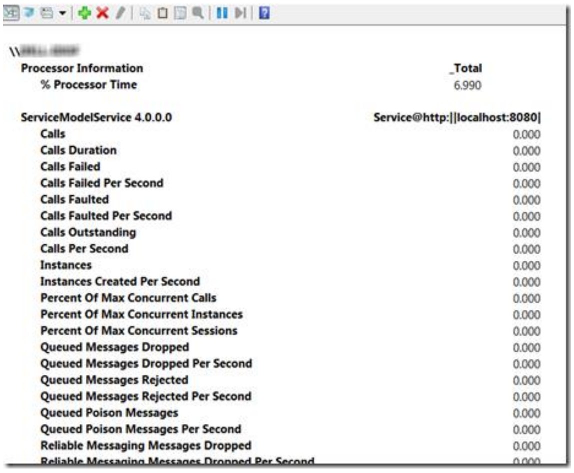
In line 4, we calculate for each candidate Web service the overall QoS score which is the sum of each normalized QoS value multiplied by the weight given in the service user request.

6.3 QoS monitoring

The QoS monitoring process aims to monitor and measure the QoS values in order to verify whether the measured values comply with QoS values published by the Web service provider. As it is mentioned in Section 4.2, we distinguish two different types of QoS: The static and dynamic. The QoS monitoring process is interested in dynamic QoS monitoring, because QoS values are known only after the Web service execution.

The QoS monitoring in the field of Web services has been studied by many addressed (e.g., [[37] [38] [39] [40] just to name a few). The authors of [37] introduce a QoS model which covers various dimensions of QoS, i.e. availability, accessibility, performance, reliability, security, and regulatory, and propose metrics to enhance QoS measurement on the service side. They realized the monitoring of QoS dimensions above through a monitoring extension of Java system application server developed in Java EE 5.0. In [38], the authors present a Probe-based Observability Mechanism required for the monitoring of the web services that facilitates observation of internal execution details of the web services during testing and execution. The authors in [39] carry out a research to develop a monitoring method for web services response time. The method proposed in this research is based on creating a proxy for connecting to the required Web service, and then calculating the Web services response time via the proxy. The work in [40] presents the Vienna Runtime Environment for Service-oriented Computing (VRESCo) that addresses some issues of current Web service technologies, with a special emphasis on service metadata, quality of service, service querying, dynamic binding and service mediation. The QoS monitoring is performed in their work to evaluate the framework through performance measurements on service querying, binding, mediation and invocation performances.

According to these studies, QoS monitoring can be performed into two approaches: (1) Client-side monitoring: the measurement of QoS is run on the client side [39], (2) Server-side monitoring: the measurement of QoS is run on the server side [37] [40]. On one hand, client-side monitoring usually gives less accurate monitoring results and requires that clients must agree to install monitoring software which may not always be the case. But on the other hand, server-side monitoring is usually accurate but requires access to the actual service implementation which is not always possible.



```
\\[redacted]
  Processor Information                                        _Total
    % Processor Time                                            6.990

  ServiceModelService 4.0.0.0                Service@http:||localhost:8080|
    Calls                                                        0.000
    Calls Duration                                               0.000
    Calls Failed                                                 0.000
    Calls Failed Per Second                                      0.000
    Calls Faulted                                                0.000
    Calls Faulted Per Second                                     0.000
    Calls Outstanding                                            0.000
    Calls Per Second                                             0.000
    Instances                                                    0.000
    Instances Created Per Second                                 0.000
    Percent Of Max Concurrent Calls                              0.000
    Percent Of Max Concurrent Instances                          0.000
    Percent Of Max Concurrent Sessions                           0.000
    Queued Messages Dropped                                      0.000
    Queued Messages Dropped Per Second                           0.000
    Queued Messages Rejected                                     0.000
    Queued Messages Rejected Per Second                          0.000
    Queued Poison Messages                                       0.000
    Queued Poison Messages Per Second                            0.000
    Reliable Messaging Messages Dropped                          0.000
    Reliable Messaging Messages Dropped Per Second               0.000
```

Figure 6: Windows Performance Counters: ServiceModelService Category.

In our work, we choose the use of a server-side monitoring mechanism, while ensuring that it does not affect existing implementations of the observed Web services. For this raison, our QoS monitoring mechanism is based on Windows Performance Counters (WPC) provided by Windows Communication Foundation (WCF) [41], which are part of the .NET Framework and offer a server-side QoS monitoring for Web services. Windows Performance Counters allow measuring the performance of Windows Communication Foundation Web services without altering any existing services.

WPC supports a rich set of counters that can be measured during the execution time of Web services. Performance counters are scoped to three different levels: Service, Endpoint and Operation. Each of these levels has performance counters to analyse the performance of a hosted WCF Web service. Service performance counters measure the service behaviour as a whole and can be used to diagnose the performance of the whole service. They can be found under the **ServiceModelService 4.0.0.0** performance object when viewed with Performance Monitor (Figure 6).

In our work, we focus on the following counters: "*Call Duration”* counter to measure the execution time, "*Calls Per Second”* counter to measure the number of a Web service invocations, and "*Failed Calls Per Second”* counter to measure the number of a Web service failures.

As depicted in figure 7, the way the QoS monitoring mechanism functions can be summarized as follows:

- Initially, the QoS monitor has to be installed on the service provider host. QoS monitor is itself a service which captures the performance counters of the monitored web services.
- Once installed, the QoS monitor has to be configured by setting the required parameters in the **Web.config** file. This configuration allows the operating system to attach the performance counters to the monitored web services.
- By default, the Windows Performance Counters are turned off because they could significantly increase the memory footprint of the WCF application. Performance counters can be enabled for the service from the diagnostics section of the **Web.config** file, as shown in the following sample configuration:

```
<configuration>
  <system.serviceModel>
      <diagnostics
   performanceCounters="All" />
  </system.serviceModel>
</configuration>
```

To specify the web service we want to monitor, we need to add its name in the services section of the Web.config file as follows:

```
<configuration>
  <system.serviceModel>
     <services>
        <service
     name="MonitoredServiceName" >
                ......
         </service>
     </services>
   </system.serviceModel>
</configuration>
```

- Once started, the QoS monitor constantly continues reading the current values of the performance counters (Call Duration, Calls Per Second, Failed Calls Per Second) and transmits them to the QoS aggregator component of the PrefWS3 system. The QoS monitor sends sequentially, to the QoS aggregator, a SOAP message containing information about the service provider, the monitored service and the corresponding measured performances.

- When the QoS aggregator component receives the performance counters values sent by the QoS monitor, it aggregates these values to calculate the execution time and the reliability of the monitored Web service. The performance counter "Calls Duration" of the counter category "ServiceModelService 4.0.0.0" is used to calculate the execution time QoS, and the performance counters "Calls Per Second", "Failed Calls Per Second" of the same category are used to calculate the reliability QoS.
- Finally, the measured QoS values are transmitted to the decision maker component. This latter compares the measured QoS values with the corresponding QoS values published by the Web service provider in the OWL-S repository. If the QoS values published do not comply with the measured QoS values then the service provider will be punished. Several forms of punishments have been proposed. In our work, we propose to temporarily exclude the web service whose QoS are not real.

The QoS monitoring mechanism of the PrefWS3 system makes use of Windows Performance Counters, which are integrated into the operating system and thus, representing an easy way to QoS monitoring.

7 Rating and reputation mechanism

Before paying the execution price of a Web service, the user is always looking to be sure of his choice. One of the mechanisms used to make the user have confidence in the selected web service is to give him the ratings of other users who have already used it. Once the web service is selected, the service user should provide a rating score to show the user satisfaction level of the invoked web service. A rating score is an integer number that ranges from 0 to 4, where the meaning of each value is as follows: 4: very satisfied, 3: satisfied, 2: neither satisfied or dissatisfied, 1: dissatisfied, 0: very dissatisfied.

In existing Rating-based approaches, the satisfaction criterion of the rater is unknown. Without knowing the intendment of the rater, it is almost impossible to make sense of a given rating. For example, a service user may give a high rating to a Web service because its execution time is small. If the execution time is not significant for a second service user, then the first service user's high rating will not be significant either. Hence, it is important to take into account the satisfaction criteria of each service user. This is done by giving a rate score for each QoS attribute of the used Web services.

The user ratings are stored in an RDF triple store. As user ratings refer to a given service request, each Rating instance contains the service user who performed the rating, the rated service, the rating date, and finally the rating scores (one rating score per QoS attribute). New ratings from the same user for the same service replace older ratings.

Over time, the qualities of a service can be changed by the service provider. In this case, old ratings are no longer representative. To address this problem, we give more importance to the recent ratings. This is done using Equation 2, where Δd is the number of days between the current date and the rating submission one. Figure 8 shows the evolution of the rating value over the time where the initial value equals 3.

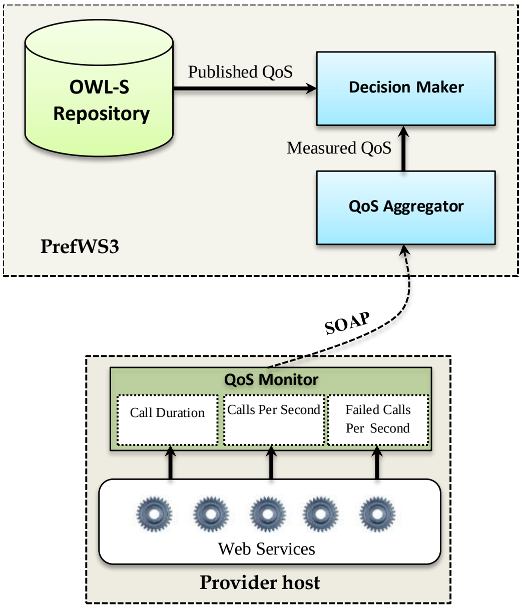


Figure 7. QoS monitoring mechanism.

$$\text{UpdatedRate}(\text{S.QoSProperty}) = \frac{Rate(S.\text{QoSProperty})}{\log_{10}(10+\Delta d\ )} \quad (2)$$

The reputation score of a service S within a single QoS attribute is computed as the average of all ratings the service receives from service users for this QoS attribute as indicated in Equation 3, where N is the number of ratings for the service S. Each rating score is normalized, as a monotonically increasing criterion, to have a value in the range of 0 to 1.

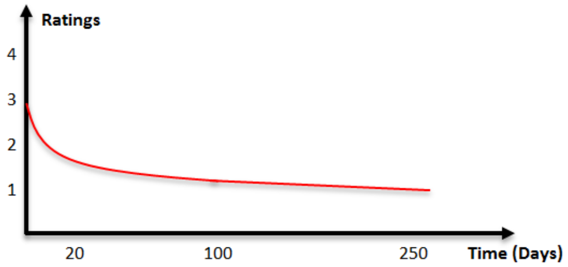


Figure 8: Example of the rating value evolution.

$$\text{ReputationScore}(\text{S. QoSProperty}) =$$

$$\text{NormalizedValue}\left(\frac{\sum \text{UpdatedRate(S.QoSProperty)}}{\text{N}}\right) \quad (3)$$

The reputation score of a service within multiple QoS attributes is computed, by Equation 4, as the weighted sum of the rating score of each quality attribute.

$$\text{OverallReputationScore}(\text{S}) =$$

$$\frac{\sum \text{ReputationScore(S.QoSProperty)}*\text{weight}}{\sum \text{weight}} \quad (4)$$

8 Evaluation

In implementing PrefWS3, we use some software and tools. PrefWS3 is developed with Java under Eclipse IDE platform. PrefWS3 makes use the OWL-S API [42] for OWL-S files parsing. Jena 2.2 [43] is used for reasoning on OWL.

In order to evaluate the performance of our proposed semantic similarity algorithm which calculates the semantic similarity between a request and a web service, we compared it with two semantic matchmakers, the SAM architecture introduced in [15], and the BSA algorithm presented in [16].We use Book, Person and Printed Material ontology presented in [15], which is retrieved from "OWL-S Service Retrieval Test Collection version 2.1" available from the SemWebCentral Website[2]. In addition, we also used request and service definitions presented in the same work.

As shown in Figure 9, Book, Person and Printed Material ontology contains information on printed material classification and related concepts such as publishers, readers, authors, book types and several other concepts.

As the properties of the superclass are inherited by its subclasses, and in order to apply our algorithm, using the ontology described above, we assume that each subclass (or subconcept) in the ontology contains one more property than its superclass (superconcept). The request and the web services input/output parameters are given in Table 7. Request input concepts are Ordinary-Publisher, Novel, and Paper-Back. Request output concepts are Local-Author and Genre.

To demonstrate the value-added features of our semantic similarity algorithm, we present a test case between Request and Web Service 1 for input matching. The input parameters for Web Service 1, as shown in Table 7, are Publisher, ScienceFiction-Book. We calculated the semantic similarity using the ConceptSim function.

By applying the InputsSim algorithm, input concepts in both request and Web service 1 are matched as follows:

- Ordinary-Publisher → Publisher: ConceptSim = 1,

  since: Ordinary-Publisher < Publisher.

- Novel → ScienceFictionBook:

ConceptSim = $\frac{\text{Size(prop(Novel)}\cap\text{prop(ScienceFictionBook))}}{\text{Size(prop(Novel)}\cup\text{prop(ScienceFictionBook))}}$ =

$\frac{4}{6} = 0.666$, since:

Novel <> ScienceFictionBook.

- Paper-Back: No match

- Paper-Back is an extra input of a request, so it can be ignored and thus, the InputSim(Request, Web Service 1) = (1+0.666)/2 = 0.833

Results of the input-output similarity calculation of all services in the test case are listed in Table 8.

Service 2 is found to be the most similar to Request according to input matching, since it has the highest score for input matching of all the other classes. In fact, all the SAM, BSA, and PrefWS3 found this to be the best matched service in input matching with a score of 0.4388 by SAM, a score of 0.77 by BSA, and a score of 0.833 by PrefWS3. However, SAM, BSA, and PrefWS3 found the Service 2 has the weakest match outputs with scores of 0.01447, 0.012, and 0.125 respectively.

On the other hand, in PrefWS3, matching Request and Service 5 should give the highest score according to output matching since {Genre → Genre : ConceptSim = 1} and {Local-Author → Publisher : ConceptSim = 0.25}. Both SAM and BSA found this to be the best matched service for output matching and scored it as 1.00018 by SAM, and 1.2565 by BSA.

Furthermore, SAM and BSA found that Service 3 has the weakest match for inputs, so this places it the latest in the rankings, which was also found as unrelated and scored as 0.541 by PrefWS3.

All the SAM, BSA, and PrefWS3 found that Service 3 and Service 4 have the same output matching scores. Thus, for Service 3 and Service 4, BSA orders the results according to the maximum value of input scores, whereas SAM uses a random selection. Finally, both the BSA and PrefWS3 found the order of total score to be:

Service 5 > Service 1 > Service 4 > Service 2 > Service 3.

The results reveal that, in both BSA and PrefWS3 systems, Service 5 has the highest total score considering both input and output matching, and Service 2 has the lowest total score. As a conclusion, comparing the results given by PrefWS3 with those given by SAM and BSA, we note that PrefWS3 offers good results but with less calculation, and therefore less time.

9 Conclusion

In this article, we introduce a semantic web services discovery and selection system (PrefWS3). An advanced feature of PrefWS3 is that it performs the service discovery and selection based on the matching level of the service advertisements with the user requests in terms of both functional and non-functional parameters. PrefWS3 is considered to be a user-centric system which helps and guides users on formulating their requirements and preferences, and hence, allows to free consumers from time consuming human computer interactions and Web search. Additionally, PrefWS3 uses a translator to translate WSDL files into OWL-S and provides semantically enriched description. As a result, enhancing web services with a semantic description of their functionality will further improve their discovery and selection. PrefWS3 uses an efficient semantic-based

[2] http://projects.semwebcentral.org

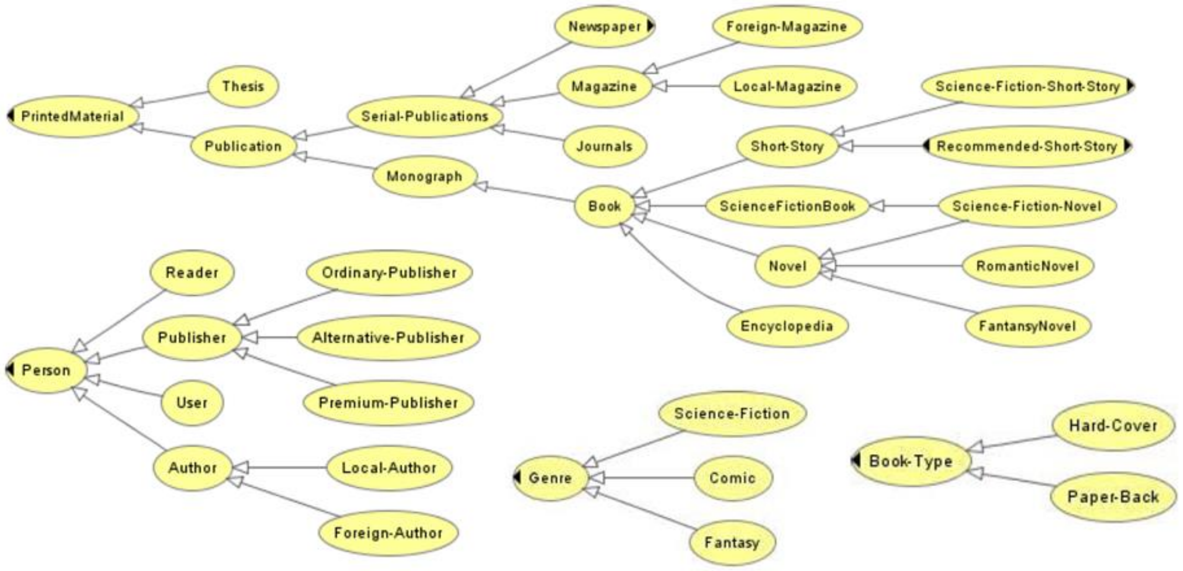


Figure 9: Book, Person and Printed Material ontology section. [15]

Request/Service Name	Inputs	Outputs
Request	Ordinary-Publisher, Novel, Paper-Back	Local-Author, Genre
Web Service 1	Publisher, ScienceFictionBook	Author, Price
Web Service 2	Book, Alternative- Publisher, Book-Type	Publisher, Price, Date
Web Service 3	FantasyNovel, Author	Price, Comic
Web Service 4	Newspaper, Book-Type, Person	Review, Fantasy
Web Service 5	Publication, Book-Type, Reader	Genre, Publisher

Table 7: Request and Services parameters.

Service name	**Scores of SAM**			**Scores of BSA**			**Scores of PrefWS3**		
	Input Sim Score	**Output Sim Score**	**Total Score**	**Input Sim Score**	**Output Sim Score**	**Total Score**	**InputS im Score**	**Output Sim Score**	**Tota l Score**
Service 1	0.35964	0.12229	0.21723	0.640	0.8571	0.7485	0.833	0.5	0.666
Service 2	0.4388	0.01447	0.27771	0.77	0.012	0.391	0.833	0.125	0.479
Service 3	0.18026	0.17033	0.08078	0.47	0.5076	0.4888	0.541	0.5	0.520
Service 4	0.23636	0.12229	0.69465	0.5321	0.5076	0.5198	0.761	0.5	0.630
Service 5	0.31718	1.00018	0.20024	0.575	1.2565	0.9157	0.75	0.625	0.687

Table 8: Comparison of PrefWS3, SAM, and BSA based on input/output parameter matching.

matching mechanism which calculates the semantic similarity between the request and the web service based on the concepts position in the ontology, the common properties between concepts, and also, either concept has annotated an input/output request parameter or an input/output web service parameter. Furthermore, PrefWS3 includes a QoS-aware process and provides a reputation mechanism that enables service users to evaluate the credibility of the web services they use, and takes into account the satisfaction criteria of each service user. In order to evaluate the effectiveness of our system, the results of a comparison of the PrefWS3 and some other published approaches (BSA and SAM) have been presented.

As future directions, we plan to incorporate the Web services composition into PrefWS3 in order to make it more practical in real-world applications. To this end, two main questions need to be asked:

1) How to combine Web services in a suitable way to fulfil the user request?

To answer this question, several approaches have been proposed such as: Constraint based composition, Business rule driven composition, AI Planning based composition, Context information based composition, Process based composition, and Model and aspect driven composition [44]. AI Planning approach has become interesting due to the maturity that the planning area has achieved in AI. We decide to extend our PrefWS3 system to support service composition by combining semantic matching and an AI planning technique. We focus on functional input and output parameters of Web services. The latter are respectively the preconditions and the effects in the planning context. Web service composition is then viewed as an AI planning based composition of semantic relationships between Web service parameters. To this end, we intend to adapt the functional matching mechanism of the PrefWS3 system to support semantic similarities between input and output parameters, and add a composition component that implements an AI planning technique.

2) How to select the best composition among a set of candidates that fulfil the same user request?

It is possible that the composition mechanism generates multiple composite services fulfilling the user request. In that case, the composite services are evaluated and ranked along the non-functional parameters such as QoS and user constraints, and the best composite service is the one which is ranked on top. Selecting a composite service that satisfies user constraints and preferences can be viewed as a Constraint Satisfaction Problem (CSP). To this end, we intend to formulate QoS based web service composition as a CSP, and adapt our QoS computing mechanism to compute the quality of a composite service when it is given the QoS of its underlying services.

References

[1] Averbakh. A, Krause. D and Skoutas. D. (2009). Exploiting User Feedback to Improve Semantic Web Service Discovery. *8th International Semantic Web Conference*. LNCS, Vol.5823, pp.33-48.

[2] Garofalakis. J, Panagis. Y, Sakkopoulos. E and Tsakalidis. A. (2006). Contemporary web service discovery mechanisms. *Journal of Web Engineering*. Vol.5, No.3, pp.265-290.

[3] Hyunkyung. Y. P and TaeDong. L. (2013). Ontology based keyword dictionary server for semantic service discovery. *IEEE Third International Conference on Consumer Electronics*. pp. 295 – 298.

[4] Paliwal. A.V, Shafiq. B, Vaidya. J, Hui. X and Adam. N. (2012). Semantics-Based Automated Service Discovery. *IEEE Transactions on Services Computing*. Vol.5, No.2, pp.260 – 275.

[5] Kopecky. J, Vitvar. T, Bournez. C and Farrell. J . (2007). SAWSDL: Semantic Annotations for WSDL and XML Schema. IEEE Internet Comput. Vol.1, No.11, pp.60-67.

[6] Roman. D, Keller. U, Lausen. H, de Bruijn. J, Lara. R, Stollberg. M, Polleres. A, Feier. C, Bussler. C and Fensel. D. (2005). Web Service Modeling Ontology. *Journal of Applied Ontology*. Vol.1, pp.77-106.

[7] Martin. D et al. (2004). OWL-S: Semantic Markup for Web Services. W3C. From http://www.w3.org/ Submission/OWL-S/.

[8] Dasgupta. S, Aroor. A, Shen. F and Lee. Y. (2014). SMARTSPACE: Multiagent Based Distributed Platform for Semantic Service Discovery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. Vol.44, No.7, pp.805- 821.

[9] Benaboud. R, Maamri. R and Sahnoun. Z. (2013). Agents and owl-s based semantic web service discovery with user preference support. *Int. Journal of Web & Semantic Technology*. Vol.4, No.2, pp.57-75.

[10] Paolucci. M, Kawamura. T, Payne. T and Sycara. K. (2002). Semantic matching of web services capabilities. *Proceedings of the 1st International Semantic Web Conference (ISWC)*, Springer-Verlag. pp.333–347.

[11] Dong. H, Hussain. F and Chang. E. (2012). Semantic Web Service matchmakers: state of the art and challenges. *Concurrency and Computation: Practice and Experience*. Vol.25, No.5, pp.961-988.

[12] Klusch. M and Kapahnke. P. (2010). iSeM: Approximated Reasoning for Adaptive Hybrid Selection of Semantic Services. *In: Aroyo L, Antoniou G, Hyvönen E, ten Teije A, Stuckenschmidt H, et al.., editors. The Semantic Web: Research and Applications*. Springer Berlin. pp.30–44.

[13] Kiefer. C and Bernstein. A. (2008). The Creation and Evaluation of iSPARQL Strategies for Matchmaking. *In: Bechhofer S, Hauswirth M, Hoffmann J, Koubarakis M, editors. The Semantic Web: Research and Applications.* Springer Berlin. pp.463–477.

[14] Klusch. M and Kapahnke. P. (2012). Adaptive Signature-Based Semantic Selection of Services with OWLS-MX3. *Multiagent and Grid Systems*. Vol.8, No.1,pp.69–82.

[15] Erdem. S.I and Ayse. B. B. (2008). SAM: Semantic Advanced Matchmaker. *R. Nayak et al. (Eds.): Evolution of the Web in Artificial Intel. Environ*. Vol.130, pp.163–190.

[16] Çelik. D and Elçi. A. (2013). A broker-based semantic agent for discovering Semantic Web services through process similarity matching and equivalence considering quality of service. *Science China Information Sciences*. Vol.56, No.1, pp.1-24.

[17] De Renzis. A, Garriga. M, Flores. A and Cechich. A. (2016). Case-based Reasoning for Web Service

Discovery and Selection. Electronic Notes in Theoretical Computer Science. Vol.321, pp.89–112.

[18] Kolodner. J. (1993). Case-Based Reasoning, Morgan Kaufmann Publishers, Inc.

[19] Sakkopoulos. E, Kanellopoulos. D and Tsakalidis. A.(2006). Semantic mining and web service discovery techniques for media resources management. *Int. Journal of Metadata, Semantics and Ontologies*, Vol.1, No.1, pp.66-75.

[20] Xu. Z, Martin. P, Powley. W and Zulkernine. F. (2007). Reputation-Enhanced QoS-based Web Service Discovery. *In Proceedings of the International Conference on Web Services*. pp.249 – 256.

[21] Mobedpour. D and Ding. C. (2013). User-centered design of a QoS-based web service selection system. *Service Oriented Computing and Applications*. Vol.7, No.2, pp.117-127.

[22] Iordache. R and Moldoveanu. F. (2014). QoS-Aware Web Service Semantic Selection Based on Preferences. *Procedia Engineering.* Vol.69, pp.1152–1161.

[23] Chouiref. Z, Belkhir. A, Benouaret. K and Hadjali. A. (2016). A fuzzy framework for efficient user-centric Web service selection. Applied Soft Computing, Vol.41, pp.51-65.

[24] Chang. C and Kuo. C. (2013). A Web Service Selection Mechanism Based on User Ratings and Collaborative Filtering. *Smart Innovation, Systems and Technologies.* Springer-Verlag Berlin Heidelberg. Vol.20, pp.439-449.

[25] Nguyen. H. T, Zhao. W and Yang. J. (2010). A trust and reputation model based on bayesian network for web services. *IEEE International Conference on Web Services*. pp.251-258.

[26] Vu. L, Hauswirth. M and Aberer. K. (2005). QoS-based Service Selection and Ranking with Trust and Reputation Management. *Proceedings of the Confederated international conference on the Move to Meaningful Internet Systems*. pp.466-483.

[27] Grozavu. A, Pleşcan. S and Mărgărint. C. (2011). Comparative Methods for the Evaluation of The Natural Risk Factors' Importance. *Present Environment and Sustainable Development*. Vol.5, No.1, pp.41–46.

[28] Zadeh. M, Seyyedi. M. (2010). Qos monitoring for web services by time series forecasting. *3rd IEEE international conference on computer science and information technology (ICCSIT)*. pp.659–663.

[29] Limam. N and Boutaba. R. (2008). QoS and Reputation-aware Service Selection. *IEEE on Network Operations and Management Symposium*. pp.403-410.

[30] Heß. A, Johnston. E and Kushmerick. N. (2004). ASSAM: A Tool for Semi-Automatically Annotating Semantic Web Services. *In: Proceedings of the 3rd International Semantic Web Conference (ISWC)*. pp.320-334.

[31] Farrag. T, Saleh. A and Ali. H. (2013). Towards SWSs Discovery: Mapping from WSDL to OWL-S Based on Ontology Search and Standardization Engine. *IEEE Transactions on Knowledge and Data Engineering.* Vol.25, No.5, pp.1135-1147.

[32] Ashraf. B. (2014). Fast Mapping Algorithm from WSDL to OWL-S. *I.J. Information Technology and Computer Science*. Vol.6, No.9, pp.24-31.

[33] Lin. L, Kai. S and Sen. S. (2008). Ontology-based QoS-Aware Support for Semantic Web Services. *Technical Report at Beijing University of Posts and Telecommunications*.

[34] Zhang. Y, Huang. H, Yang. D, Zhang. H, Chao. H and Huang. Y. (2009). Bring QoS to P2P-based semantic service discovery for the Universal Network. *Journal Personal and Ubiquitous Computing*. Vol.13, No.7, pp.471–477.

[35] Saaty. T. (1995). Decision Making for Leaders. RWS Publications.

[36] [36] Lin. D. (1998). An information-theoretic definition of similarity. *In Proceedings of International Conference on Machine Learning*. pp.296-304.

[37] Artaiam. N and Senivongse. T. (2008). Enhancing service-side qos monitoring for web services. *In SNPD '08: Proceedings of the 2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*. IEEE Computer Society. pp.765-770.

[38] Saxena. N, Goel. A and Singh. D. (2009). A probe-based observability mechanism for monitoring of web services. *Int J Recent Trends Eng*. Vol.1, No.1, pp.600–602.

[39] Asadollah. S. A and Thiam. K. C. (2011). Web service response time monitoring: architecture and validation. Theoretical and Mathematical Foundations of Computer Science. Vol.164, pp.276–282.

[40] Michlmayr. A, Rosenberg. F, Leitner. P and Dustdar. S. (2010). End-to-end support for QoS-aware service selection, binding, and mediation in VRESCo. *IEEE Trans Serv Comput.* Vol.3, No.3, pp.193–205.

[41] Peiris. C, Mulder. D, Bahree. A, Chopra. A, Cicoria. S and Pathak. N. (2007). Pro WCF: Practical Microsoft SOA Implementation". Apress.

[42] Hewlett-Packard Development Company. (2001). Jena RDF API. from: http://www.hpl.hp.com/semweb/jena.htm

[43] Mindswap-Maryland Information and Network Dynamics Lab. (2004). Semantic Web agents project: OWL-S Java API. from: http://www.mindswap.org/2004/owl-s/api/index.shtml

[44] D'Mello. D. N, Ananthanarayana. V. S and Salian. S. (2011). A Review of Dynamic Web Service Composition Techniques. *First International Conference on Computer Science and Information Technology*. Springer Berlin Heidelberg. pp 85-97.

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several scientific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute temperature, known as the Stefan–Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, energy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research departments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 900 staff, has 700 researchers, about 250 of whom are postgraduates, around 500 of whom have doctorates (Ph.D.), and around 200 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the universities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; applied mathematics. Most of the activities are more or less closely connected to information sciences, in particular computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automation and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the independent state of **Slove**nia (or S♡nia). The capital today is considered a crossroad between East, West and Mediterranean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

From the Jožef Stefan Institute, the Technology park “Ljubljana” has been proposed as part of the national strategy for technological development to foster synergies between research and industry, to promote joint ventures between university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

Part of the Institute was reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology park "Ljubljana". The project was developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park is a shareholding company hosting an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Higher Education, Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of the Economy, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel.:+386 1 4773 900, Fax.:+386 1 251 93 85
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Public relations: Polona Strnad

INFORMATICA

AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS

INVITATION, COOPERATION

Submissions and Refereeing

Please submit a manuscript to: http://www.informatica.si/Editors/PaperUpload.asp. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible from typing errors to global philosophical disagreements. The chosen editor will send the author the obtained reviews. If the paper is accepted, the editor will also send an email to the managing editor. The executive board will inform the author that the paper has been accepted, and the author will send the paper to the managing editor. The paper will be published within one year of receipt of email with the text in Informatica MS Word format or Informatica LaTeX format and figures in .eps format. Style and examples of papers can be obtained from http://www.informatica.si. Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the managing editor.

QUESTIONNAIRE

□ Send Informatica free of charge

□ Yes, we subscribe

Please, complete the order form and send it to Dr. Drago Torkar, Informatica, Institut Jožef Stefan, Jamova 39, 1000 Ljubljana, Slovenia. E-mail: drago.torkar@ijs.si

Since 1977, Informatica has been a major Slovenian scientific journal of computing and informatics, including telecommunications, automation and other related areas. In its 16th year (more than twentytwo years ago) it became truly international, although it still remains connected to Central Europe. The basic aim of Informatica is to impose intellectual values (science, engineering) in a distributed organisation.

Informatica is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the Refereeing Board.

Informatica is free of charge for major scientific, educational and governmental institutions. Others should subscribe (see the last page of Informatica).

ORDER FORM – INFORMATICA

Name: ....

Title and Profession (optional): ....

....

Home Address and Telephone (optional): ....

....

Office Address and Telephone (optional): ....

....

E-mail Address (optional): ....

Signature and Date: ....

Informatica WWW:

http://www.informatica.si/

Referees from 2008 on:

A. Abraham, S. Abraham, R. Accornero, A. Adhikari, R. Ahmad, G. Alvarez, N. Anciaux, R. Arora, I. Awan, J. Azimi, C. Badica, Z. Balogh, S. Banerjee, G. Barbier, A. Baruzzo, B. Batagelj, T. Beaubouef, N. Beaulieu, M. ter Beek, P. Bellavista, K. Bilal, S. Bishop, J. Bodlaj, M. Bohanec, D. Bolme, Z. Bonikowski, B. Bošković, M. Botta, P. Brazdil, J. Brest, J. Brichau, A. Brodnik, D. Brown, I. Bruha, M. Bruynooghe, W. Buntine, D.D. Burdescu, J. Buys, X. Cai, Y. Cai, J.C. Cano, T. Cao, J.-V. Capella-Hernández, N. Carver, M. Cavazza, R. Ceylan, A. Chebotko, I. Chekalov, J. Chen, L.-M. Cheng, G. Chiola, Y.-C. Chiou, I. Chorbev, S.R. Choudhary, S.S.M. Chow, K.R. Chowdhury, V. Christlein, W. Chu, L. Chung, M. Ciglarič, J.-N. Colin, V. Cortellessa, J. Cui, P. Cui, Z. Cui, D. Cutting, A. Cuzzocrea, V. Cvjetkovic, J. Cypryjanski, L. Čehovin, D. Čerepnalkoski, I. Ćosić, G. Daniele, G. Danoy, M. Dash, S. Datt, A. Datta, M.-Y. Day, F. Debili, C.J. Debono, J. Dedič, P. Degano, A. Dekdouk, H. Demirel, B. Demoen, S. Dendamrongvit, T. Deng, A. Derezinska, J. Dezert, G. Dias, I. Dimitrovski, S. Dobrišek, Q. Dou, J. Doumen, E. Dovgan, B. Dragovich, D. Drajic, O. Drbohlav, M. Drole, J. Dujmović, O. Ebers, J. Eder, S. Elaluf-Calderwood, E. Engström, U. riza Erturk, A. Farago, C. Fei, L. Feng, Y.X. Feng, B. Filipič, I. Fister, I. Fister Jr., D. Fišer, A. Flores, V.A. Fomichov, S. Forli, A. Freitas, J. Fridrich, S. Friedman, C. Fu, X. Fu, T. Fujimoto, G. Fung, S. Gabrielli, D. Galindo, A. Gambarara, M. Gams, M. Ganzha, J. Garbajosa, R. Gennari, G. Georgeson, N. Gligorić, S. Goel, G.H. Gonnet, D.S. Goodsell, S. Gordillo, J. Gore, M. Grčar, M. Grgurović, D. Grosse, Z.-H. Guan, D. Gubiani, M. Guid, C. Guo, B. Gupta, M. Gusev, M. Hahsler, Z. Haiping, A. Hameed, C. Hamzaçebi, Q.-L. Han, H. Hanping, T. Härder, J.N. Hatzopoulos, S. Hazelhurst, K. Hempstalk, J.M.G. Hidalgo, J. Hodgson, M. Holbl, M.P. Hong, G. Howells, M. Hu, J. Hyvärinen, D. Ienco, B. Ionescu, R. Irfan, N. Jaisankar, D. Jakobović, K. Jassem, I. Jawhar, Y. Jia, T. Jin, I. Jureta, Đ. Juričić, S. K, S. Kalajdziski, Y. Kalantidis, B. Kaluža, D. Kanellopoulos, R. Kapoor, D. Karapetyan, A. Kassler, D.S. Katz, A. Kaveh, S.U. Khan, M. Khattak, V. Khomenko, E.S. Khorasani, I. Kitanovski, D. Kocev, J. Kocijan, J. Kollár, A. Kontostathis, P. Korošec, A. Koschmider, D. Košir, J. Kovač, A. Krajnc, M. Krevs, J. Krogstie, P. Krsek, M. Kubat, M. Kukar, A. Kulis, A.P.S. Kumar, H. Kwaśnicka, W.K. Lai, C.-S. Laih, K.-Y. Lam, N. Landwehr, J. Lanir, A. Lavrov, M. Layouni, G. Leban, A. Lee, Y.-C. Lee, U. Legat, A. Leonardis, G. Li, G.-Z. Li, J. Li, X. Li, X. Li, Y. Li, Y. Li, S. Lian, L. Liao, C. Lim, J.-C. Lin, H. Liu, J. Liu, P. Liu, X. Liu, X. Liu, F. Logist, S. Loskovska, H. Lu, Z. Lu, X. Luo, M. Luštrek, I.V. Lyustig, S.A. Madani, M. Mahoney, S.U.R. Malik, Y. Marinakis, D. Marinčič, J. Marques-Silva, A. Martin, D. Marwede, M. Matijašević, T. Matsui, L. McMillan, A. McPherson, A. McPherson, Z. Meng, M.C. Mihaescu, V. Milea, N. Min-Allah, E. Minisci, V. Mišić, A.-H. Mogos, P. Mohapatra, D.D. Monica, A. Montanari, A. Moroni, J. Mosegaard, M. Moškon, L. de M. Mourelle, H. Moustafa, M. Možina, M. Mrak, Y. Mu, J. Mula, D. Nagamalai, M. Di Natale, A. Navarra, P. Navrat, N. Nedjah, R. Nejabati, W. Ng, Z. Ni, E.S. Nielsen, O. Nouali, F. Novak, B. Novikov, P. Nurmi, D. Obrul, B. Oliboni, X. Pan, M. Pančur, W. Pang, G. Papa, M. Paprzycki, M. Paralič, B.-K. Park, P. Patel, T.B. Pedersen, Z. Peng, R.G. Pensa, J. Perš, D. Petcu, B. Petelin, M. Petkovšek, D. Pevec, M. Pičulin, R. Piltaver, E. Pirogova, V. Podpečan, M. Polo, V. Pomponiu, E. Popescu, D. Poshyvanyk, B. Potočnik, R.J. Povinelli, S.R.M. Prasanna, K. Pripužić, G. Puppis, H. Qian, Y. Qian, L. Qiao, C. Qin, J. Que, J.-J. Quisquater, C. Rafe, S. Rahimi, V. Rajkovič, D. Raković, J. Ramaekers, J. Ramon, R. Ravnik, Y. Reddy, W. Reimche, H. Rezankova, D. Rispoli, B. Ristevski, B. Robič, J.A. Rodriguez-Aguilar, P. Rohatgi, W. Rossak, I. Rožanc, J. Rupnik, S.B. Sadkhan, K. Saeed, M. Saeki, K.S.M. Sahari, C. Sakharwade, E. Sakkopoulos, P. Sala, M.H. Samadzadeh, J.S. Sandhu, P. Scaglioso, V. Schau, W. Schempp, J. Seberry, A. Senanayake, M. Senobari, T.C. Seong, S. Shamala, c. shi, Z. Shi, L. Shiguo, N. Shilov, Z.-E.H. Slimane, F. Smith, H. Sneed, P. Sokolowski, T. Song, A. Soppera, A. Sorniotti, M. Stajdohar, L. Stanescu, D. Strnad, X. Sun, L. Šajn, R. Šenkeřík, M.R. Šikonja, J. Šilc, I. Škrjanc, T. Štajner, B. Šter, V. Štruc, H. Takizawa, C. Talcott, N. Tomasev, D. Torkar, S. Torrente, M. Trampuš, C. Tranoris, K. Trojacanec, M. Tschierschke, F. De Turck, J. Twycross, N. Tziritas, W. Vanhoof, P. Vateekul, L.A. Vese, A. Visconti, B. Vlaovič, V. Vojisavljević, M. Vozalis, P. Vračar, V. Vranić, C.-H. Wang, H. Wang, H. Wang, H. Wang, S. Wang, X.-F. Wang, X. Wang, Y. Wang, A. Wasilewska, S. Wenzel, V. Wickramasinghe, J. Wong, S. Wrobel, K. Wrona, B. Wu, L. Xiang, Y. Xiang, D. Xiao, F. Xie, L. Xie, Z. Xing, H. Yang, X. Yang, N.Y. Yen, C. Yong-Sheng, J.J. You, G. Yu, X. Zabulis, A. Zainal, A. Zamuda, M. Zand, Z. Zhang, Z. Zhao, D. Zheng, J. Zheng, X. Zheng, Z.-H. Zhou, F. Zhuang, A. Zimmermann, M.J. Zuo, B. Zupan, M. Zuqiang, B. Žalik, J. Žižka,

Informatica

An International Journal of Computing and Informatics

Web edition of Informatica may be accessed at: http://www.informatica.si.

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Litostrojska cesta 54, 1000 Ljubljana, Slovenia.
The subscription rate for 2016 (Volume 40) is
– 60 EUR for institutions,
– 30 EUR for individuals, and
– 15 EUR for students
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

Typesetting: Borut Žnidar.
Printing: ABO grafika d.o.o., Ob železnici 16, 1000 Ljubljana.

Orders may be placed by email (drago.torkar@ijs.si), telephone (+386 1 477 3900) or fax (+386 1 251 93 85). The payment should be made to our bank account no.: 02083-0013014662 at NLB d.d., 1520 Ljubljana, Trg republike 2, Slovenija, IBAN no.: SI56020830013014662, SWIFT Code: LJBASI2X.

Informatica is published by Slovene Society Informatika (president Niko Schlamberger) in cooperation with the following societies (and contact persons):
Robotics Society of Slovenia (Jadran Lenarčič)
Slovene Society for Pattern Recognition (Janez Perš)
Slovenian Artificial Intelligence Society (Dunja Mladenić)
Cognitive Science Society (Urban Kordeš)
Slovenian Society of Mathematicians, Physicists and Astronomers (Andrej Likar)
Automatic Control Society of Slovenia (Sašo Blažič)
Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Vojteh Leskovšek)
ACM Slovenia (Andrej Brodnik)

Informatica is financially supported by the Slovenian research agency from the Call for co-financing of scientific periodical publications.

Informatica is surveyed by: ACM Digital Library, Citeseer, COBISS, Compendex, Computer & Information Systems Abstracts, Computer Database, Computer Science Index, Current Mathematical Publications, DBLP Computer Science Bibliography, Directory of Open Access Journals, InfoTrac OneFile, Inspec, Linguistic and Language Behaviour Abstracts, Mathematical Reviews, MatSciNet, MatSci on SilverPlatter, Scopus, Zentralblatt Math

Volume 40 Number 2 June 2016 **ISSN 0350-5596**

Informatica

An International Journal of Computing and Informatics

End of Special Issue / Start of normal papers

Informatica **40** (2016) Number 2, pp. 157–275