Vol. 20, No. 7, pp. 807-978 **Neural Network World** 2010

VOLUME 20 NUMBER 7 ◯ 2010 ISSN 1210-0552

Special Issue on 6th International Conference on Hybrid Artificial Intelligence Systems

NEURAL NETWORK WORLD

Editor-in-Chief: M. Novák

Associate Editors:

P. Bouchner	D. Harmancová
D. Húsek	M. Jiřina
P. Musílek	R. Neruda
L. J. M. Rothkrantz	M. Svítek
V. Šebesta	M. Vlček
Z. Votruba	J. Wiedermann

Published by: *Institute of Computer Science,* Academy of Sciences of the Czech Republic,
182 07 Prague 8, Pod Vodárenskou věží 2, Czech Republic
Phone: (00420) 2 8689 0639, (00420) 2 6605 2080, (00420) 2 6605 3100
Fax: (00420) 2 8658 5789, **E-Mail:** nnw@cs.cas.cz

Czech Technical University Prague, Faculty of Transportation Sciences
110 00 Prague 1, Konviktská 20, Czech Republic
Phone: (00420) 2 2435 9548

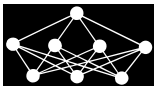
The journal is monitored in the following Thomson Scientific indexes:
- Science Citation Index Expanded (also known as SciSearch®)
- CompuMath Citation Index®
- Current Contents®/Engineering Computing and Technology
- Neuroscience Citation Index®

World-Wide Web: `http://www.nnw.cz`

Neural Network World is published in 6 issues per annum under Ministry of Culture of the Czech Republic registration number MK ČR E 6099.

Distributor: *Distributed by the publisher.*

Printed in the Czech Republic



EDITORIAL

Hybrid Artificial Intelligence Systems

Emilio Corchado[*], *Manuel Graña*[†], *Václav Snášel*[‡], *Michal Wozniak*[§]

The idea of hybrid intelligent systems is becoming more and more popular due to its capability in handling various real-world complex problems, involving imprecision, uncertainty, vagueness and high-dimensionality. Hybrid intelligent systems provide us with the opportunity to use both our knowledge and raw data to solve problems in a more interesting and promising way. This multidisciplinary research field is continually explored and expanded by the artificial intelligence research community.

The objective of series of international conferences on Hybrid Artificial Intelligence Systems (HAIS) is to provide an interesting opportunity to present and discuss the latest theoretical advances and real-world applications. The 5th International Conference on Hybrid Artificial Intelligence Systems was successfully organized in San Sebastián, Spain, in June of 2010 by the Computational Intelligence Group (GIC) of the University of the Basque Country. More than 140 participants from 12 countries participated in the conference. All submitted papers were reviewed carefully by at least two independent referees, and on the basis of their suggestions, Program Committee of HAIS selected 133 papers for oral presentation. Additionally, six plenary lectures were given by world-recognized scientists, such as Prof. Éloi Bossé, Prof. Mihai Datcu, Prof. Ali-Akbar Ghorbani, Prof. James Llinas, Prof. Marios Polycarpou, and Prof. Gerhard X Ritter.

This special issue of the prestigious journal *Neural Network World* consists of twelve extended papers selected carefully by the HAIS Program Committee Chairs. The articles focus on various hybrid computational intelligence approaches and their applications. Let us give short outlines of the works presented in this issue.

In their paper "A Hybridized Neuro-Genetic Solution for Controlling Industrial R^3 Workspace", Eloy Irigoyen *et al.* deal with the trajectory generation by multi-objective genetic algorithm technique and the reference tracking by a neural control scheme with an enhanced training algorithm.

[*]Emilio Corchado
University of Salamanca, Salamanca, Spain, E-mail: escorchado@usal.es
[†]Manuel Graña
University of the Basque Country, San Sebastián, Spain, E-mail: manuel.grana@ehu.es
[‡]Václav Snášel
VSB Technical University Ostrava, Ostrava, Czech Republic, E-mail: vaclav.snasel@vsb.cz
[§]Michal Wozniak
Wroclaw University of Technology, Wroclaw, Poland, E-mail: michal.wozniak@pwr.wroc.pl

In their article "Assessing the Evolution of Learning Capabilities and Disorders with a Graphical Exploratory Analysis of Surveys Containing Missing and Conflicting Answers", Luciano Sánchez *et al.* propose a novel extension to imprecise data of graphical exploratory statistics, where each element is represented by a shape in a map, modeling the uncertainty of the variables. These maps are used for measuring the evolution of the learning capabilities acquired by a group of students during a course, and also for comparing the development of behavior of possibly dyslexic children.

In their work "Base Classifiers in Boosting-Based Classification of Sequential Structures", Przemyslaw Kazienko and Tomasz Kajdanowicz study the usage of the proper base classifier in a new approach to sequence labeling problem based on the boosting concept.

In their paper "Combination of One-class Classifiers for Multiclass Problems by Fuzzy Logic", Tomasz Wilk and Michal Wozniak propose a new method of one-class classifier combination based on the neuro-fuzzy approach which allows to restore a multiclass recognition problem.

In their article "DASBE: Decision-Aided Semi-Blind Equalization for MIMO Systems with Linear Precoding", José A. García-Naya *et al.* combine unsupervised and supervised learning algorithms to avoid the periodical transmission of unnecessary pilots in digital communication systems with linear precoding, which implies a considerable spectral efficiency improvement in comparison with traditional channel estimation methods.

In their work "Detection of Heat Flux Failures in Building Using a Soft Computing Diagnostic System", Javier Sedano *et al.* discuss a novel Soft Computing Diagnostic System for the detection of heat flux failures in buildings.

In their paper "Evaluating the Performance of Evolutionary Extreme Learning Machines by a Combination of Sensitivity and Accuracy Measures", Javier Sánchez-Monedero *et al.* demonstrate an efficient alternative to the current Pareto based algorithms used when dealing with simultaneous optimization of accuracy and sensitivity objectives.

In their article "Learning Hose Transport Control with Q-learning", Borja Fernandez-Gauna *et al.* demonstrate an innovative solution to the construction of a controller for Multi-Component Linked Robotic Systems (MCLRS) based on reinforcement learning. The approach is demonstrated on a simplified system which exemplifies the basic paradigm of MCLRS moving a hose on a limited environment.

In their work "Combining Classifiers Using Trained Fuser – Analytical and Experimental Results", Michal Wozniak and Marcin Zmyslony discuss several methods of classifier fusion and evaluate their qualities on the basis of analytical and experimental researches.

In their paper "Neural Classifiers for Schizophrenia Diagnostics Support on Diffusion Imaging Data", Alexandre Savio *et al.* demonstrate the innovative detection of schizophrenia on the basis of a feature extraction method applied to anatomical and diffusion weighted brain magnetic resonance imaging. The high discriminant nature of these features allows for easy classification with a variety of neural network architectures.

In his work "The New Upper Bound on the Probability of Error in a Binary Tree Classifier with Fuzzy Information", Robert Burduk presents a new estimation of the

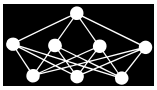
upper bound of error probability for a binary tree classifier with fuzzy observations.

In their work "Ranked Tag Recommendation Systems Based on Logistic Regression", José Ramón Quevedo *et al.* propose an approach to tag recommendation in social networks based on a learning system. In addition, the proposed method explores several pieces of information which the learner feeds on, and the fact that the fashion matters in the sense that recent posts are more useful and suitable for recommendation of new tags.

We hope that these papers will inspire the researchers to invent new ideas of hybrid systems and develop new practical and efficient computer applications on the basis of the concepts mentioned above.

We would like to thank the editors-in-chief of *Neural Networks World*, Prof. Mirko Novák for supporting this special issue, and all the authors for their contributions and the reviewers who did a wonderful job in completing the reviews within a short period of time. We also thank Prof. Dušan Húsek from Institute of Computer Science, Academy of Sciences of the Czech Republic, for his help during the organization of this issue and all the editorial assistance related to this issue. Finally, we would like to express our thanks to the technical staffs of *Neural Network World*, who helped realize this special issue within a very short period of time.

We would also like to invite you to participate in the next, 6^{th} edition of the International Conference on Hybrid Artificial Intelligent Systems, which will take place in Wroclaw, Poland, from 23^{rd} to 25^{th} May, 2011.



A HYBRIDIZED NEURO-GENETIC SOLUTION FOR CONTROLLING INDUSTRIAL R^3 WORKSPACE

*E. Irigoyen, M. Larrea, J. Valera, V. Gómez, F. Artaza**

Abstract: This work presents a hybridized neuro-genetic control solution for R^3 workspace application. The solution is based on a multi-objective genetic algorithm reference generator and an adaptive predictive neural network strategy. The trajectory calculation between two points in an R3 workspace is a complex optimization problem considering the fact that there are multiple objectives, restrictions and constraint functions which can play an important role in the problem and be in competition. We solve this problem using genetic algorithms, in a multi objective optimization strategy. Subsequently, we enhance a training algorithm in order to achieve the best adaptation of the neural network parameters in the controller which is responsible for generating the control action for a nonlinear system. As an application of the proposed hybridized control scheme, a crane tracking control is presented.

Key words: *Hybrid neuro-genetic solution, optimal trajectory generation, multi-objective genetic algorithm, nonlinear neural control, adaptive predictive control*

Received: 20th September 2010
Revised and accepted: 13th November 2010

1. Introduction

Nowadays, our aggressive market requires more accurate, reliable, productive, and competitive industrial solutions. This involves a monumental effort from researchers and technicians in order to solve complex, real-world problems. One of these problems is the industrial kinematic control (where it is necessary to handle raw materials, semi-finished and finished products), which implies a wide number of goals to reach [1]. In sequential industrial processes, for the transportation, handling and machining of materials and products into different manufacturing

*E. Irigoyen, M. Larrea, J. Valera, V. Gómez, F. Artaza
Department of Systems Engineering and Automatic Control, Computational Intelligence Group, University of the Basque Country (UPV/EHU), ETSI, 48013 Bilbao, Spain, E-mail: {eloy.irigoyen, m.larrea}@ehu.es

workplaces, it is more essential than ever to obtain automated and enhanced solutions based on new technologies such as computational intelligence.

This work presents a hybrid intelligent solution that solves tracking and movement problems in an R^3 workspace. It uses a complex calculation of a precise trajectory. It also solves accuracy and control action issues for precise and safe tracking operations. Our solution uses different computational intelligence techniques for solving these problems. We initially implemented one device for tracing optimal trajectories as the reference to the control system. Later on, we chose a control scheme based on adaptive and predictive control fields. Previous control loop approaches have been studied as presented in [2] where a 2D crane anti-swing problem is solved.

The first part of our work focuses on designing a Multi Objective Genetic Algorithm (MOGA). This MOGA solves a nonlinear and complex problem for calculating R^3 trajectories [3]. This solution takes account of requirements based on the workspace (restricted areas, points of passage, etc.), and constraints on the basis of parameter values (max-min) to preserve the life of actuators and different components. The MOGA technique has been used with success in different works such as [4] and [5].

Furthermore, the tracking operation is made by an Adaptive-Predictive Neural Network (APNN) control system, which includes some intelligent strategies to reach the appropriate target. There exist different APNN control approaches where the performances of different control loops are tested in [6] and [7].

Our system contains two Recurrent Neural Networks (NNARX): The first one provides a nonlinear process model for estimation of the process output and derivatives in time, and the second one is involved in the current action calculation at every sample time.

Next, in Chapter 2, the different elements of the hybridized neuro-genetic system will be presented. In Chapter 3., the components of the multi objective genetic algorithm reference generator will be laid out in detail. Then, the neural network adaptive predictive control strategy that was selected as well as the specific NN training algorithms designed will be explained. A case of study with a crane system will be introduced in Chapter 5. Finally, the conclusions obtained and some ideas for future work will be commented.

2. Hybridized Neuro-Genetic Strategy

This work deals with the hybridization of different computational intelligence techniques for solving non-trivial real tracking problems. The genetic algorithms performed well in the optimal solution calculation within multi-objective problems. In this approach, we have designed a MOGA Reference Generator (MOGA-RG) in order to obtain a trajectory within an R^3 workspace. The MOGA-RG takes account of several objectives and different constraints of movement and workspace, which creates a more complex problem, the control of nonlinear systems.

To develop an appropriate control solution, the neural network paradigm has been implemented. Different neural network topologies were designed to perform the identification of the nonlinear system and to generate a nonlinear controller. An adaptive predictive control strategy was selected for this work as a result of

certain needs referring to the control strategy and the use of the NNs as controllers and identifiers. This strategy was employed in several different works like [8], [6].

The scheme used (Fig. 1) has the following four basic blocks: MOGA-RG, neural network identifier, neural network based controller and the nonlinear system to be controlled. All these elements have their respective training algorithms. The identifier can provide an online identified model, which means the scheme has the capability to learn the system dynamics simultaneously to the nonlinear system evolution.

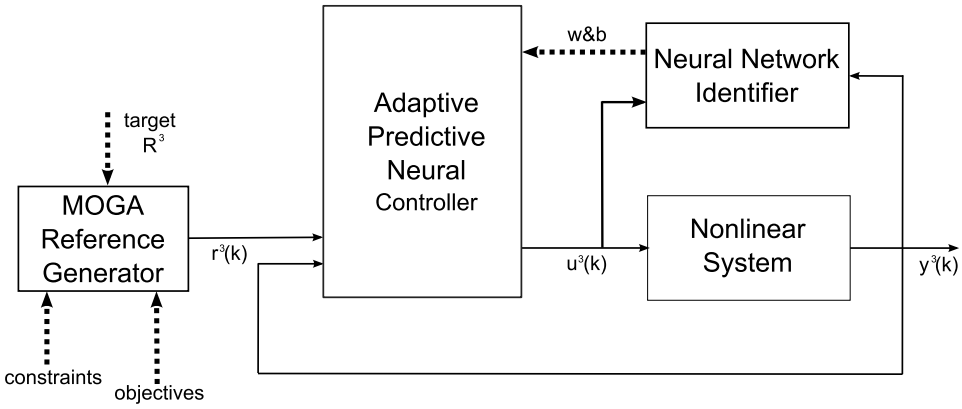


Fig. 1 *Control scheme.*

The block Adaptive Predictive Neural Network Controller (APNNC) is responsible for generating the control action for the nonlinear system calculated in a predefined prediction horizon (Fig. 2) whereas the block MOGA-RG calculates a path to be tracked by the nonlinear system.

The APNNC performs a simulation of the entire loop and employs a replica of the nonlinear system provided by the NN identifier in order to do so. This replica provides not only the nonlinear system output estimation ($\hat{y}$) but also the estimation of the identified system derivatives ($\frac{\partial \hat{y}_{k+1}}{\partial u_k}, \frac{\partial \hat{y}_{k+1}}{\partial y_k}$,..). Those estimations are integrated in the training algorithm to be presented in Section 4.1. Once the training algorithm finalizes its work, the NN controller weight and bias are tuned to generate a control action that will be the output of the block. In Fig. 3, the different stages in the training process are presented.

One of the advantages that the adaptive predictive control has is the capability to change the controller behavior. This is positive when the nonlinear system to be controlled suffers a modification (e.g. deterioration, wear, use of slightly different parts, etc.) and the nonlinear system model changes to a new operating regime, causing the controller change too.

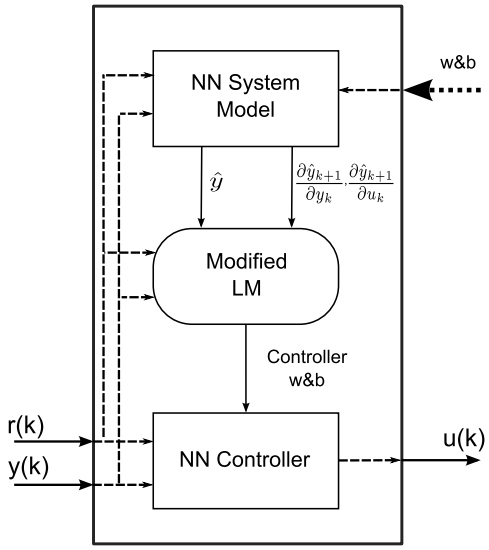


Fig. 2 *Control scheme.*

3. MOGA Reference Generator

The trajectory calculation between two points in an R^3 workspace is a complex optimization problem considering the fact that there are multiple objectives, restrictions and constraint functions which can play an important role in the problem. The following are some important aspects that have to be considered for an appropriate trajectory reference calculation: minimization of the time employed to travel from the initial to the final point, minimization of the traveled distance between these two points avoiding obstacles and restricted areas, minimum oscillation according to previous acceleration reference calculations, and minimization of mechanical elements wear in movement transition. Consequently, the problem formulation is not trivial especially when we take account of the fact that some objectives are not differentiable, so gradient or higher derivatives information is not available when searching for an optimal solution. This kind of problem can be solved using the Genetic Algorithm (GA) [9], in a multi objective optimization strategy, as previously introduced in Valera et al. [3].

Thereby, a possible trajectory reference $r(t)$ between two points in an R^3 workspace is given by Equation 1.

$$r(t) = [x(t), y(t), z(t)] \tag{1}$$

In industrial processes the R^3 workspace usually has some restricted workspaces, as shown in equation 2.

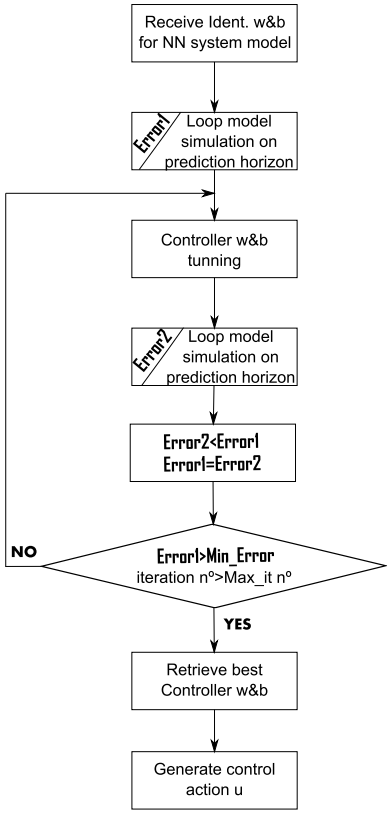


Fig. 3 *Flowchart of adapting parameters and predicting errors.*

$$\begin{aligned} 0 \leq x(t) \leq X_{lim} \\ 0 \leq y(t) \leq Y_{lim} \\ 0 \leq z(t) \leq Z_{lim} \end{aligned} \tag{2}$$

Furthermore, this optimization problem has two main objectives to reach: to minimize the $r(t)$ length or distance traveled, and to minimize the required path time to travel from one point to the other. In addition, the trajectory has to satisfy the following constraints and restriction functions:

- Electromechanical component related constraints [10]: Speed $v(k)$ and acceleration $a(k)$ on each axis or movement must not exceed the thresholds determined by the device manufacturers.
- Mechanical transmission elements and the useful life of the system: The acceleration or torque gradient $j(k)$ of each movement must not exceed a certain value to avoid so-called "impact torques" in the mechanical transmission el-

ements, which cause jerky movements and vibrations, reducing useful life of the elements.

- Constraints related to avoiding obstacles in the workspace: Any point of this trajectory cannot be included in the space defined by the constrained limited surface: $z = f_1(x; y)$, $y = f_2(x)$, and $z_p = f_3(x; y)$.

In our work, a Multi Objective Reference Generator based on Genetic Algorithms (MOGA) has been developed in order to satisfy all the objectives and constraints presented above. Fig. 4 schematically represents the different components that perform the R^3 optimal trajectory within the MOGA reference generator.

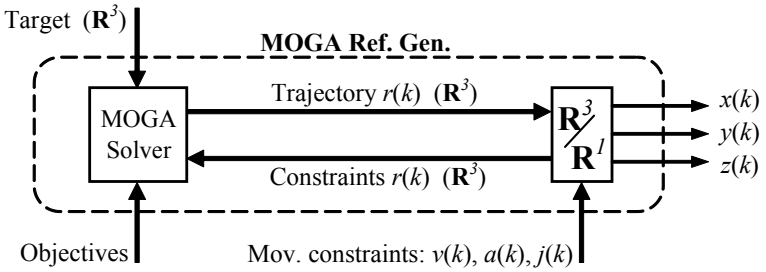


Fig. 4 *MOGA reference generator.*

The MOGA core is the solver that generates values of the optimal trajectory $r(k)$, in each sample time. For this calculation, the solver takes account of the constraints related to the working restricted areas and the movement constraints $[v(k), a(k), j(k)]$, and tries to minimize the travelled time and the trajectory length as objectives.

In order to have a smooth trajectory, bounded acceleration reference and bounded acceleration gradient [11], we divided the positioning time into six intervals taking the speed reference shown in Valera et al. [3] into account.

To find three smooth position references ($x(k)$, $y(k)$, and $z(k)$), we used a nonlinear search method based on the Multi Objective Genetic Algorithm (MOGA) presented before, resulting in an R^3 combined trajectory (R^3 workspace) that simultaneously minimizes the distance traveled, time used, and final position error. The formulated objectives for MOGA execution can also be found in Valera et al. [3].

The trajectory generation is a non-trivial problem because some objectives are in competition. It has therefore been necessary to select the optimal solution by using the Pareto set optimal solutions technique. As seen in Fig. 5, the MOGA calculates a set of non-inferior solutions that we represented in the Pareto front. By analyzing these solutions, we are able to find a solution that optimizes the R^3 movement depending on the actual working point and the objectives priorities previously defined. In Fig. 5, the time required for the trajectory (objective 1) is represented on the x axis, the error of travelling near the point $[xp, yp, zp]$ (objective 2) is represented on z axis, and the total distance (objective 3) is represented

on y axis. In future works this selection will be solved by computational intelligence techniques, as a fuzzy system recording the actions of an experienced operator.

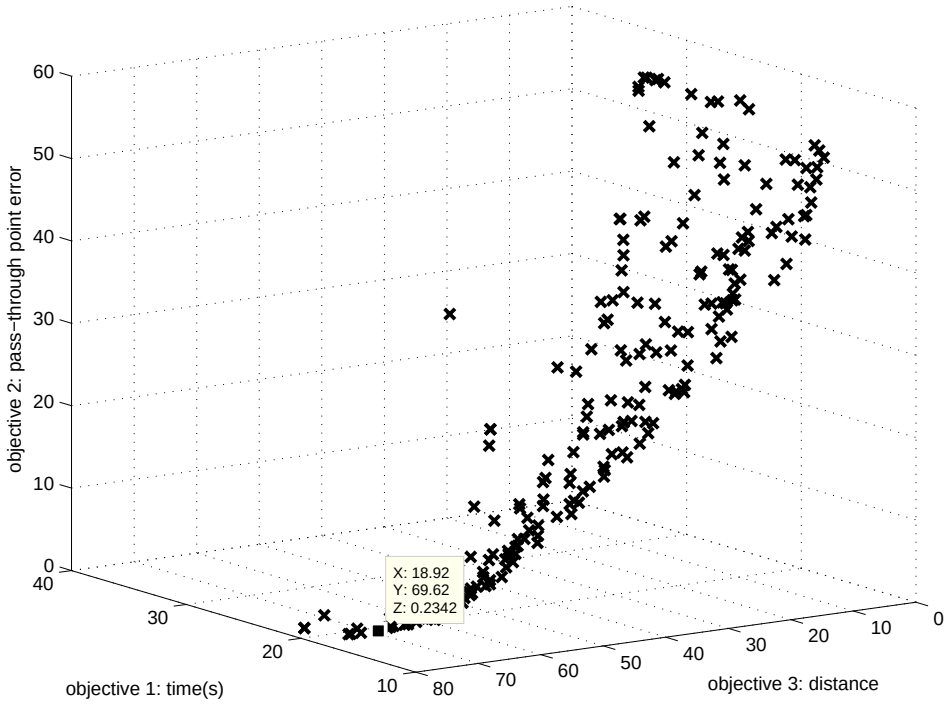


Fig. 5 *Set of non inferior solutions. Pareto frontier.*

4. Neural Network Adaptive Predictive Control

In this section the one dimensional adaptive predictive control will be introduced. Using this strategy, the two NNs employed are MultiLayer Perceptrons (MLP). The MLP are known as universal approximators because of their capacity to approximate any function of interest (both linear and nonlinear) as well as its derivatives [12]. The latter one is of great importance in the implementation of the identifier since the derivatives that it provides will be integrated into the training algorithm. The topology of the NN controller and the NN identifier are correspondingly presented in Fig. 6 and Fig. 7.

The NN controller (Fig. 6) is a NN AutoRegressive with eXogenous input (NNARX) that gives output feedback (control action).

The NN identifier (Fig. 7) obtains the nonlinear system model based on the system input/output relation. Once the model is obtained, it can be used to emulate the nonlinear system behavior and to extract its derivatives. Both the NN controller and the NN identifier can be trained online or offline.

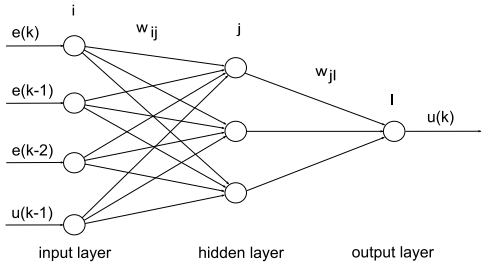


Fig. 6 *NN controller.*

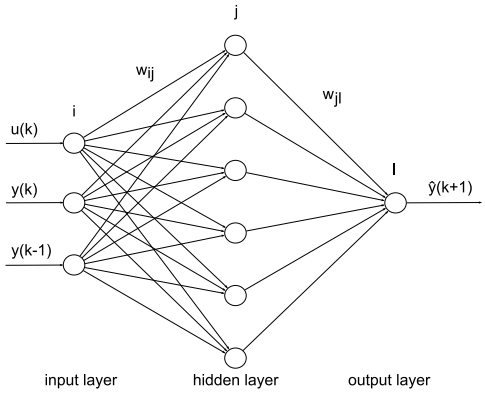


Fig. 7 *NN identifier.*

4.1 Neural network training

The NN controller is trained in the "adaptive predictive neural controller" block that can be seen in Fig. 1. As previously mentioned, inside this block a simulation of the control loop is performed. This simulation creates the possibility to simulate the control loop evolution for a prediction horizon, and to simulate it for different control actions. The NN controller needs to know, or estimate, the error produced on its output in order to be trained. As the desired control action (u) is unknown, the error produced during the output of the NN controller is also unknown. The only known error is the one produced on the output of the nonlinear system ($y(k) - r(k)$ in Fig. 1), which can be related to the NN controller weight and bias through the NN system model. This way, the equation 3 [13] can be used to train the NN controller in a K prediction horizon.

$$\sum_{k=1}^{K} \frac{\partial E_k}{\partial w_{lij}} = \sum_{k=1}^{K} \sum_{k'=1}^{k} \sum_{k''=0}^{k'-1} \frac{\partial E_k}{\partial y_{k'}} \cdot \frac{\partial y_{k'}}{\partial u_{k''}} \cdot \frac{\partial u_{k''}}{\partial w_{lij}}. \tag{3}$$

Equation 3 is made up of three terms. The first one relates the error committed in the control loop output with the nonlinear system output. The second one

relates the nonlinear system output to the control action. Finally, the third term relates the control action to the NN controller weights and biases. The first and the third terms are known terms. The first is the one that depends on the error function used, and the third is the one that can be calculated by backpropagation. The second term represents the model of the nonlinear system to be controlled. A general representation of a nonlinear system can be expressed by using the following Equation 4.

$$y(k') = M[y(k'-1), ..., y(k'-n), u(k'-1), ..., u(k'-m)], \tag{4}$$

where n is the nonlinear system order that must satisfy $m \leq n$. Deriving $y(k')$ from $u(k'')$, the unknown term $(\frac{\partial y_{k'}}{\partial u_{k''}})$ can be obtained. This term can in turn be broken down in the following Equation 5 [13].

$$\frac{\partial^+ y_{k'}}{\partial u_{k''}} = \sum_{i=1}^{n} \frac{\partial y_{k'}}{\partial y_{k'-i}} \cdot \frac{\partial^+ y_{k'-i}}{\partial u_{k''}} + \sum_{j=1}^{m} \frac{\partial y_{k'}}{\partial u_{k'-j}} \cdot \frac{\partial^+ u_{k'-j}}{\partial u_{k''}}. \tag{5}$$

The previous work [14], [13] has shown that the reduction of the computational times can be achieved by neglecting some of these terms ($(\frac{\partial u_{k'-j}}{\partial u_{k''}} = 0$ when $k'-j \neq k'')$. By neglecting these terms, the second term of Equation 5 results in the following Equation 6.

$$\frac{\partial^+ y_{k'}}{\partial u_{k''}} = \sum_{i=1}^{n} \frac{\partial y_{k'}}{\partial y_{k'-i}} \cdot \frac{\partial^+ y_{k'-i}}{\partial u_{k''}} + \frac{\partial y_{k'}}{\partial u_{k''}}. \tag{6}$$

Now the three terms of Equation 5 can be found. These three terms are known on the basis of "NN system model" input/output relations. The "universal approximator" property has been applied in [15] to obtain the derivatives of the identified system using the Equations 7, 8, 9 to do so, being the NN represented in Fig. 7.

$$\frac{\partial \hat{y}(k+1)}{\partial u(k)} = \sum_{j=1}^{n} w_{1j} o_j (1-o_j) w_{j1} \tag{7}$$

$$\frac{\partial \hat{y}(k+1)}{\partial y(k)} = \sum_{j=1}^{n} w_{2j} o_j (1-o_j) w_{j1} \tag{8}$$

$$\frac{\partial \hat{y}(k+1)}{\partial y(k-1)} = \sum_{j=1}^{n} w_{3j} o_j (1-o_j) w_{j1}, \tag{9}$$

where w_{1j} represents the weight that links input 1 with the neuron j of the hidden layer, w_{j1} represents the weight that links the output of the neuron j of the hidden layer to the neuron of the output layer, o_j represents the output of the neuron j of the hidden layer, and the n of the summation represents the number of neurons in the hidden layer.

4.2 NN controller training algorithm modification

The LM algorithm calculates the updated term for the weights and biases on the basis of the equation ΔW in [16]. The modification proposed, which includes the dynamics of the nonlinear system, affects the term on the output layer to be backpropagated (Δ^M presented in [16]).

$$\Delta^M = -\dot{F}^M(\underline{n}^M) \cdot \frac{\partial y_{k'}}{\partial u_{k''}}. \tag{10}$$

Applying this formula and following the development presented in [16], the dynamics of the nonlinear system and the ones of the NN controller are backpropagated. Therefore all the Jacobian terms are calculated so the weight adaptation term (ΔW) can be obtained. Finally we emphasize the different meaning of the term $e'(\underline{w})$ in equation 11 for this work. If the original work represented $e(\underline{w})$ as the error committed in the NN output, this work uses $e'(\underline{w})$ as the error committed in the output of the control loop.

$$\Delta W = \left[J^T(\underline{w}) \cdot J(\underline{w}) + \mu \cdot I\right]^{-1} \cdot J(\underline{w}) \cdot e'(\underline{w}), \tag{11}$$

This Equation is used in the same manner as the traditional LM algorithm in [16]. $J(\underline{w})$ is the Jacobian matrix which is composed of the partial derivatives of the errors in the NN output ($e(\underline{w})$) on the weights ($\underline{w}$)(12).

$$J(\underline{w}) = \begin{pmatrix} \frac{\partial e_1(\underline{w})}{\partial w_1} & \frac{\partial e_1(\underline{w})}{\partial w_2} & \cdots & \frac{\partial e_1(\underline{w})}{\partial w_N} \\ \frac{\partial e_2(\underline{w})}{\partial w_1} & \frac{\partial e_2(\underline{w})}{\partial w_2} & \cdots & \frac{\partial e_2(\underline{w})}{\partial w_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_K(\underline{w})}{\partial w_1} & \frac{\partial e_K(\underline{w})}{\partial w_2} & \cdots & \frac{\partial e_K(\underline{w})}{\partial w_N} \end{pmatrix} \tag{12}$$

To calculate the Jacobian matrix [16], the term ($\frac{\partial^+ y_{k'}}{\partial u_{k''}}$) of equation (6) is backpropagated through the layers of the NN controller.

5. Application to Crane Position Control

The adaptive predictive control strategy is applied in the control of a travelling crane. The load trajectory calculation in the R^3 workspace is a complex optimization problem considering the multiple objectives, restrictions and constraint functions. The nonlinear problem of the swinging angle control is considered as a good exercise for the proposed NN control system. The crane model used consists of a Matlab/Simulink block provided by Inteco company with a real model of the crane (Fig. 8). See [1] for the mathematical model.

The following information pertains to the trajectory of the MOGA: initial pos. $(0, 0, 0)$, final pos. $(30, 80, 10)$ with crossing point $(50, 50, 50)$ cm. The constraints that the MOGA must respect on these 3 axes are; max. acceleration 5 cm/s^2, max. speed 10 cm/s and max. jerk 0.5 cm/s^3. The objectives applied to the MOGA are: passing through the specified crossing point ($error < 0.5\ cm$), minimization of the travel time required and minimization of the distance travelled. The resultant trajectory is shown in Fig. 9.

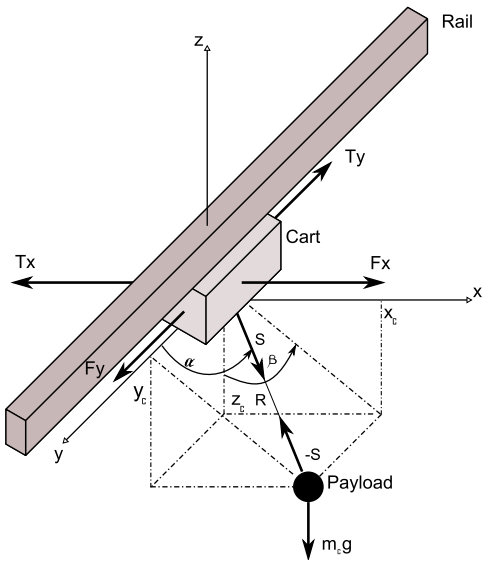


Fig. 8 *Crane model.*

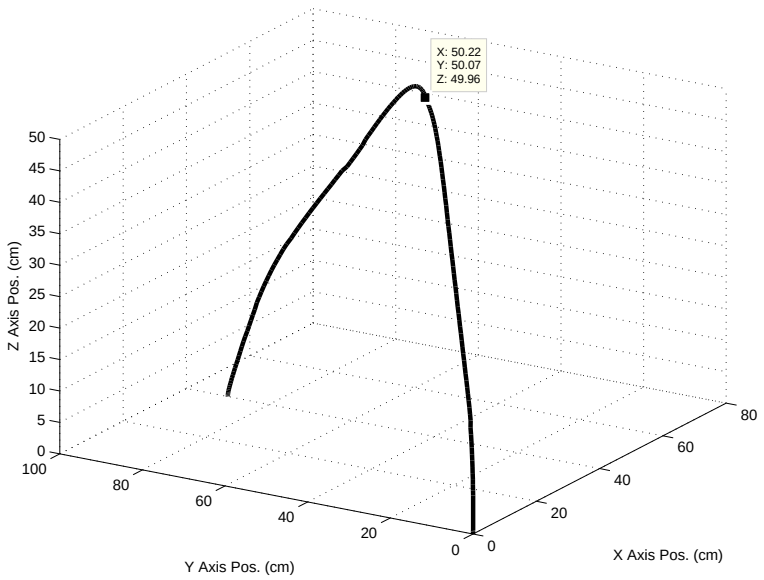


Fig. 9 R^3 *trajectory.*

The x-axis control behavior has been observed in a preliminary test. The main objective of the test has been to control the crane position while minimizing the swing of the load. Offline identification of the crane was performed to extract the

model to be used in the control loop. The identification was carried out applying random entries (both positive and negative steps inside the work range) to the NN identifier. The training has been performed with the following parameters: training vector length = 4001, validation vector length = 1000, number of epochs = 1000, initial weights randomly generated within an interval calculated as in the work [17]. The identification results for the training stage and validation stage are presented in Fig. 10.

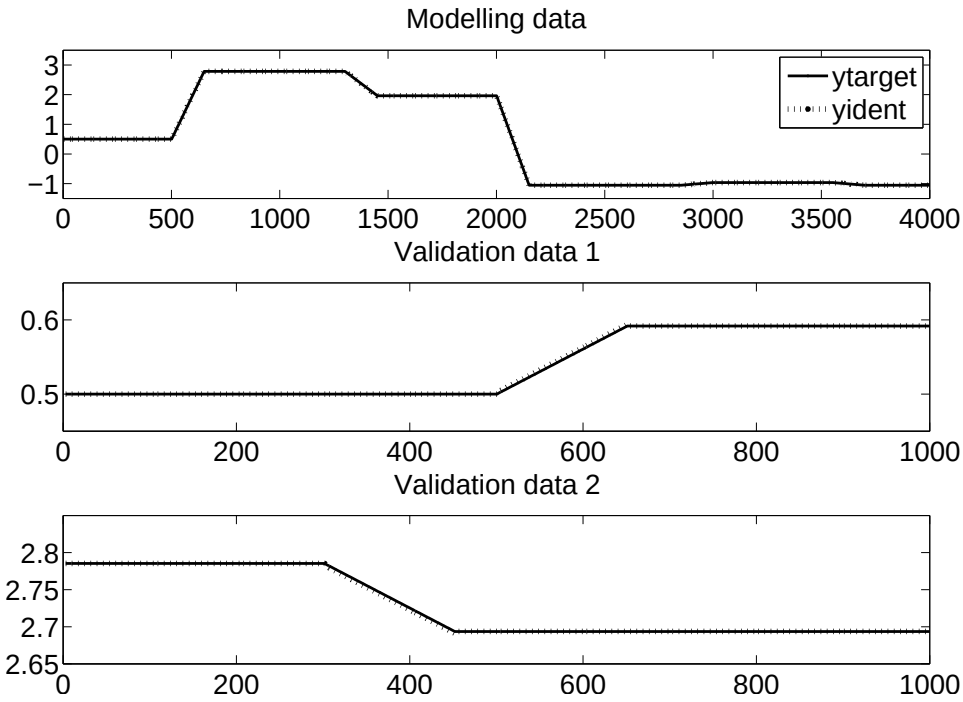


Fig. 10 *Crane identification.*

Fig. 11 shows the control of the x-axis position; the dotted line is the path generated by the MOGA, and the solid line is the tracking performed by the controller. The other lines represent the smooth control action and the low swinging of the load.

6. Conclusions

This work tackles the problem of R^3 multiobjective reference generation and the system control under these circumstances. With an intelligent search algorithm based on MOGA, the solution is stable, robust and it is a fast way to find optimal solutions when real-time requirements are not needed and when the problem involves many objectives.

Moreover, the present paper shows the use of NNs in an adaptive predictive control strategy. The simulation results show a correct online adaptation of the NN

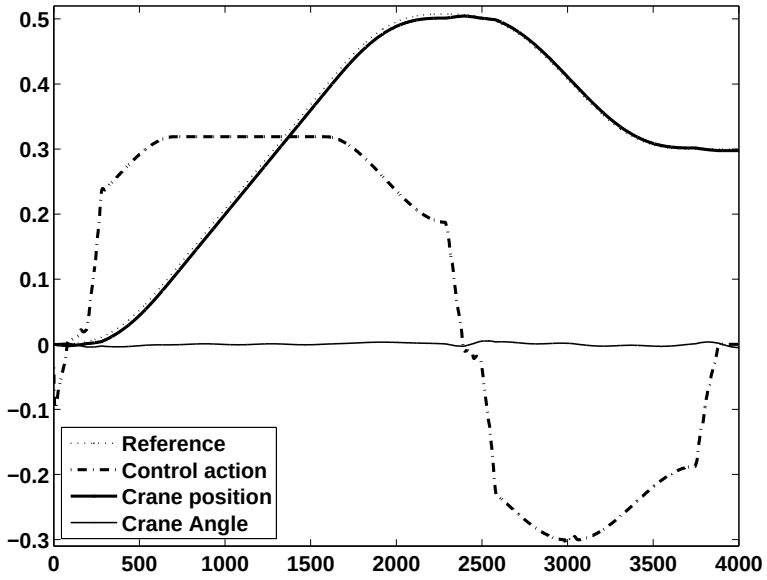


Fig. 11 *Simulation results.*

controller and the validity of the modification made to the LM training algorithm. This modification allows the integration of the nonlinear system dynamics into the training algorithm, thus being able to train the NN controller despite not knowing the nonlinear system. The NN identifier estimates the dynamics of the nonlinear. The use of restrictions to control action has been tested in various works such as [18], where first order training algorithms are used. These restrictions may be of interest when implementing controller training that penalizes abrupt changes in the control action. We will also take hierachical issues [19] into account.

Acknowledgement

This work comes under the ATICTA research project with reference SAIOTEK08 granted by Basque Government, and the BAIP2020 research project granted by CDTI of the Spanish Government, with permission of INGETEAM for the paper publication.

References

[1] Pauluk M., Korytowski A., Turnau A., Szymkat M.: Time optimal control of 3d crane. Proceedings of the 7th Inter. Conference on Methods and Models in Automation and Robotics, 2001, pp. 927–936.

[2] Liu G., Mareels I.: Advantages of smooth trajectory tracking as crane anti-swing schemes. In: IEEE International Conference on Robotics and Biomimetics, (Piscataway, NJ, USA), 2008, pp. 1486–1490.

[3] Valera J., Irigoyen E., Gómez-Garay V., Artaza F.: Application of neuro-genetic techniques in solving industrial crane kinematic control problem, IEEE International Conference On Mechatronics, 2009, pp. 231–237.

[4] Pathak B., Singh H., Srivastava S.: Multi-resource-constrained discrete time-cost tradeoff with moga based hybrid method. In: IEEE Congress on Evolutionary Computation, (Piscataway, NJ, USA), 2007, pp. 4425–4432.

[5] Xing X., Yuan D., Yan J.: A novel moga-based method of flight control law design for a helicopter and its application. In: Proceedings of the SPIE - The International Society for Optical Engineering, vol. **7128**, (USA), 2008, pp. 71282K (6 pp.).

[6] Lu C.-H., Tsai C.-C.: Adaptive predictive control with recurrent neural network for industrial processes: An application to temperature control of a variable-frequency oil-cooling machine, IEEE Transactions on Industrial Electronics, vol. **55**, March 2008, pp. 1366–1375.

[7] Ge S. S., Yang C., Lee T. H.: Adaptive predictive control using neural network for a class of pure-feedback systems in discrete time, IEEE Transactions on Neural Networks, vol. **19**, Sept. 2008, pp. 1599–1614.

[8] Tan K. K., Lee T. H., Huang S. N., Leu F. M.: Adaptive-predictive control of a class of siso nonlinear systems, Dynamics and Control, vol. **11**, Apr. 2001, pp. 151–174.

[9] Eiben A. E., Smith J. E.: Introduction to Evolutionary Computing. Springer Verlag, 2003.

[10] Suh J.-H., Lee J.-W., Lee Y.-J., Lee K.-S.: An automatic travel control of a container crane using neural network predictive pid control technique, International Journal of Precision Engineering and Manufacturing, vol. **7**, no. 1, 2006, pp. 35–41.

[11] Anand V. B.: Computer Graphics and Geometric Modeling for Engineers. New York, NY, USA: John Wiley & Sons, Inc., 1993.

[12] Hornik K., Stinchcombe M., White H.: Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, Neural Networks, vol. **3**, 1990, pp. 551–560.

[13] Irigoyen E., Galván J., Pérez-Ilzarbe M.: Neural networks for constrained optimal control of nonlinear systems, IJCNN, vol. **4**, 2000, pp. 299–304.

[14] Galván J.: Tuning of optimal neural controllers, Proc. Int. Conf. on Engineering of Intelligent Systems, 1998, pp. 213–219.

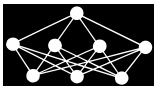
[15] Fujinaka T., Kishida Y., Yoshioka M., Omatu S.: Stabilization of double inverted pendulum with self-tuning neuro-pid, IJCNN, vol. **4**, 2000, pp. 345–348.

[16] Hagan M. T., Menhaj M. B.: Training feedforward networks with the marquardt algorithm, IEEE Transactions on Neural Networks, vol. **5**, Nov. 1994, pp. 989–993.

[17] Irigoyen E., Pinzolas M.: Numerical bounds to assure initial local stability of narx multilayer perceptrons and radial basis functions, Neurocomputing, vol. **72**, no. 1-3, 2008, pp. 539–547.

[18] Irigoyen E., Galván J., Pérez-Ilzarbe M. J.: A neuro predictive controller for constrained nonlinear systems, IASTED International Conference Artificial Intelligence and Applications, 2003.

[19] Grana M., Torrealdea F.: Hierarchically structured systems, European Journal of Operational Research, vol. **25**, no. 1, 1986, pp. 20–26.



ASSESSING THE EVOLUTION OF LEARNING CAPABILITIES AND DISORDERS WITH A GRAPHICAL EXPLORATORY ANALYSIS OF SURVEYS CONTAINING MISSING AND CONFLICTING ANSWERS

Luciano Sánchez[*], *Inés Couso*[†], *José Otero*[‡], *Ana Palacios*[§]

Abstract: The analysis of the evolution of learning with graphical maps is based on the placement of the individuals in positions that are computed on the basis of their answers to certain tests. These techniques are useful for detecting similarities between the knowledge profiles of the subjects and can also be used for assessing the acquisition of capabilities after a course. In this paper, we propose to extend some graphical exploratory analysis techniques to the case where there are missing or conflicting answers in the tests. We will also consider that either a missing or unknown answer, or a set of conflictive answers to a survey, is aptly represented by an interval or a fuzzy set. This representation causes that each individual in the map is no longer a point but a figure whose shape and size determine the coherence of the answers and whose position with respect to its neighbors determines the similarities and differences between the individuals.

Key words: *Knowledge surveys, graphical exploratory analysis, multidimensional scaling, fuzzy fitness-based genetic algorithms*

Received: 20th September 2010
Revised and accepted: 13th November 2010

1. Introduction

Graphical exploratory analysis is a simple but useful strategy for analyzing the latent data in educational questionnaires [19]. This technique consists in processing the answers to certain tests for obtaining a numerical profile of the knowledge

[*]Luciano Sánchez
Computer Science Department, Oviedo University, Spain, E-mail: luciano@uniovi.es
[†]Inés Couso
Statistics Department, Oviedo University, Spain, E-mail: couso@uniovi.es
[‡]José Otero
Computer Science Department, Oviedo University, Spain, E-mail: jotero@uniovi.es
[§]Ana Palacios
Computer Science Department, Oviedo University, Spain, E-mail: palaciosana@uniovi.es

of the subjects under study, and in projecting this data in a map, where each individual will be placed depending on these profiles. This allows the examiner to identify groups with similar background problems, segment heterogeneous groups and perceive the evolution of the learning skills or the abilities acquired during the course.

There exist tools that can generate views of the aforementioned data for easily drawing conclusions and making predictions about the effectivity of a course [14, 18], however the generalization of these techniques to sets of items that can be incomplete or imprecise has not, to the best of our knowledge, been addressed yet. This fact limits the usefulness of this technique in two frequent problems: (a) that the individual leaves unanswered questions (blank items), and (b) the dispersion of the values of the items associated to the same latent variable is too high, thus the average value of the items is no longer a good estimator. According to our approach, a missing or unknown answer in the survey will be represented by an interval or a fuzzy set. For instance, if an item is a number between 0 and 10, an unanswered question will be associated with the interval [0,10]. We will not try to make up a coherent answer for blank items [8], but we will carry the imprecision in all the calculations. In turn, multi-item values will also be represented by intervals or fuzzy sets. For instance, let $(6, 1, 5)$ be the values of three items. With our methodology, instead of replacing this triplet by its mean, the value "4", we could say that the answer is an unknown number in the range $[1, 6]$ (the minimum and the maximum of the answers) or else a fuzzy set, understood as a nested family of intervals at different confidence levels [5].

Using intervals or fuzzy sets for representing unknown values causes that each individual in the map is no longer a point but a figure whose shape and size determine the coherence and completeness of the answers and whose relative position determines the similarities between it and the other individuals. In this paper, we will explain how this map can be generated with the help of interval (or fuzzy) valued fitness function-driven genetic algorithms. Observe that certain modern approaches, like Independent Component Analysis (ICA) and Self Organized Maps (SOM), have fuzzy extensions [2, 7] that might also seem appropriate for this problem, however the algorithms we are aware of are not designed for using fuzzy data but for improving the robustness when working with crisp data, and thus are intended for solving problems fundamentally different than this. Other nonlinear extensions of PCA, like Curvilinear Component Analysis (CCA) [11], have not been extended to the fuzzy case yet. Indeed, all of these advanced nonlinear techniques are closely related to a technique widely used in psychology: Multi-Dimensional Scaling (MDS) [10]. This last technique has been recently generalized to fuzzy data [6]. The algorithm in this last reference shares a common background with our own extension, and we will compare both in the following sections.

The structure of the rest of the paper is as follows: in Section 2, we describe the representation of the information contained in those questionnaires (educational knowledge surveys and tests for diagnosing dyslexia) that will be used in this study. In Section 3, we introduce the concept of Evolutionary Graphical Exploratory Analysis for vague data and discuss its relation with the mentioned questionnaires. In Section 4, we show the outcome of this method in real-world cases. We provide some concluding remarks in Section 5.

2. Representation Issues

Sensible measures for the evolution of students' capabilities are important in order to choose adequate teaching methods. At the beginning of a course, the prerequisite skills of a sometimes heterogeneous group of students must match the instructional approach. After the course ends, the evaluation of the learning outcomes should also consider the differences between the initial preparation of the students, so the impact of the teaching methodology for each kind of student can be precisely assessed.

From another point of view, evaluating the learning outcomes with respect to the prerequisite skills is arguably akin to the problem of measuring the evolution of certain learning disorders. For example, let us consider the diagnosis of dyslexia in children. This problem is detected with non-writing based tests, measuring capabilities, such as verbal comprehension, logic reasoning and sensory-motor skills [15]. As we will explain later in this paper, it is not easy to detect dyslexia in early childhood, as the natural differences between the skills of the children mask the symptoms. However, if the tests are done yearly, then each child passes through different development stages, and interesting information can be obtained when the changes between two consecutive tests are analyzed. Then again, a sensible measure of the evolution of the learning capabilities is needed.

2.1 Questionnaires

In both the preceding cases, the information is acquired by means of questionnaires. Generally speaking, a questionnaire is intended to measure a number of latent or hidden variables, whose value is indirectly determined by averaging the answers to many different questions related to the observable variables, or items. In the following, we will use the term *multi-item value* to refer to the set of items containing all the information conveyed by the questionnaire about the value of a latent variable. We will also assume that the latent variables measure the capability of an individual for solving certain kind of problems, and the items are the answers to the questions comprising the test [17].

Two different kinds of questionnaires will be used for assessing students' capacities and detecting learning disorders, respectively. On the one hand, educational knowledge surveys are intended to measure the capability of the student for understanding and solving those problems related to the learning outcomes of a course. These surveys comprise short questions related to specific aspects of these outcomes, and are designed by the teacher. The students can answer by writing a single line, or choosing between several alternatives in a printed or web-based questionnaire. On the other hand, those tests used for diagnosing a learning disorder are based on different variables, related to the acquisition of language skills, attention deficit, hyperactivity, and other indicators. These last tests are standardized (see Tab. I) and do not comprise questions but consist of graphical exercises involving shapes, colors and lists of names or numbers.

In either case, the variables of interest are latent, and sets of items or *constructs* must be produced to measure each domain of meaning. In this section, we revise

Category	Test	Description
Verbal comprehension	BAPAE	Vocabulary
	BADIG	Verbal orders
	BOEHM	Basic concepts
Logic reasoning	RAVEN	Color
	BADIG	Visual memory
Sensory-motor skills	BENDER	Visual-motor coordination
	BADIG	Perception of shapes
	BAPAE	Spatial relations, Shapes, Orientation
	STAMBACK	Auditive perception, Rhythm
	HARRIS/HPL	Laterality, Pronunciation
	GOODENOUGHT	Spatial orientation, Body scheme
Reading-Writing	TALE	Analysis of reading and writing

Tab. I *Categories of the tests currently applied in Spanish schools for detecting dyslexia in children between 5 and 8 years. The names of the tests are standardized in Spain, see [15] for the bibliographic references.*

the properties of the questionnaires that are used in this study, and define a common representation for both that can be combined with the Evolutionary Graphical Exploratory Analysis described in Section 3.

2.2 Educational knowledge surveys

These surveys can be used for assessing the quality of learning, and they are also meaningful from a didactical point of view, as they allow students to perceive the whole content of the course. Teachers can use these surveys for deciding the best starting level for the lectures [13], specially in Master or pre-doctoral lectures, where the profiles of the students attending the same course are different. Recently this has also been applied to teacher education and certification [20]. When the survey is done at the end of the course, the effectivity of the teaching methodology along with the attitude and dedication of the students is measured. There is certain consensus in the literature about the weak relationship between methodology/dedication and scoring [4]. Because of this, a survey (different than an exam, designed to score the students) is needed.

The design of the constructs involved in a knowledge survey is often guided by the Bloom taxonomy [1, 3]. Other researchers propose taxonomies that classify learning phases [9] that could be useful to design questions that reveal where the student is in the learning curve or assess the critical thinking levels in a given area of the subject. Lastly, with regard to the measurement scales, the constructs in this particular work have been measured by several five-point Likert scales, and therefore the data comprises multi-item values. Each multi-item value is converted into an interval or a fuzzy set by means of the procedure explained later in this section.

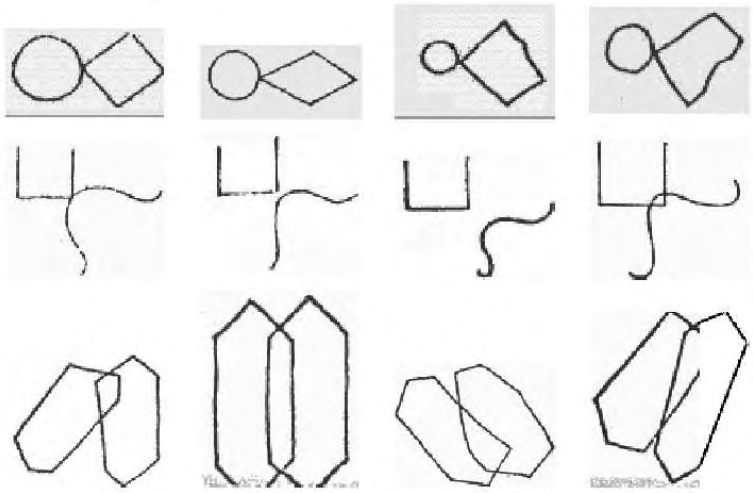


Fig. 1 *Example of some of Bender's tests for detecting dyslexia. Upper part: The angles of the shape in the right are qualified by a list of adjectives that can contain the words "right", "incoherent", "acceptable", "regular" and "extra". Middle and lower part: The relative position between the figures can be "right and separated", "right and touching", "intersecting", etc.*

2.3 Tests for diagnosing dyslexia

All the tests that have been used in this research are currently being used in Spanish schools for detecting dyslexia (see Tab. I). In Fig. 1, we have reproduced one of the graphical exercises involved in the analysis, that is copying some geometric drawings. A psychologist or specialist dyslexia teacher scores these exercises. In this particular case, this expert has to decide whether the angles, relative position and other geometrical properties have been accurately copied or not, choosing between a given set of adjectives such as "right", "incoherent", "acceptable", "regular" or "extra". Other exercises have numerical scores, and generally speaking the data consists of multi-item values, as in the preceding case. However, in this work we have also allowed the expert to express indifference between different responses by means of intervals, as in "lower than 3" or "between acceptable and regular", thus each item may also be an interval. There are 13 categories of tests, that expand to a total of 413 numerical, categorical and interval-valued variables. By aggregating the answers to each one of these categories, we will represent each individual by means of a vector of 13 multi-item variables.

2.4 Fuzzy representation of multi-item values

For estimation of the values of the latent variables, multi-item variables are aggregated. Following previous works [17], we have decided that converting the aggregate into a number is not always convenient, because relevant information is lost. Therefore, a set-valued aggregation operator is used instead. In this work, we

will represent each multi-item value by a fuzzy set (or, as a limit case, an interval) such that its α-cuts are confidence intervals with degree $1-\alpha$ of the mean value of the latent variable. This procedure converts a questionnaire into a vector of fuzzy values (or intervals), one for each variable of interest, and this transformation loses less information than aggregating the items with central tendency measures. The numerical algorithm is described in [17], and it is illustrated in the example that follows, condensed from that reference.

Example 1 *Let us suppose that a latent variable x_0 is described by the following set of items:*

$$X = (2, 1, 3, 3, 2, 2, 4). \tag{1}$$

We will assume that X is a simple random sample of a population whose mean is the unknown value x_0. Let $\widetilde{A}$ be the membership function of the fuzzy set that describes our knowledge about x_0. Then, the family of its cuts $\{\widetilde{A}^\alpha\}$ is a nested family of confidence intervals such that $P(x_0 \in A^\alpha) = P(A^\alpha) \geq 1-\alpha$. The membership function can be built from the quantiles of the bootstrap distribution of the sample mean, as shown in Fig. 2.

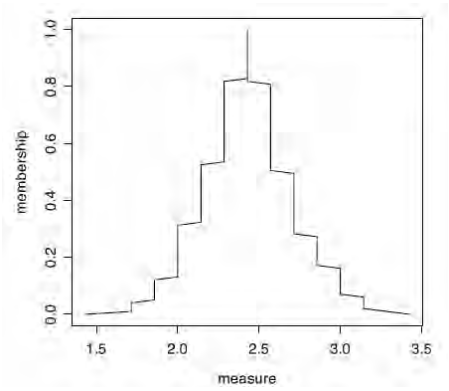


Fig. 2 *Bootstrap-based fuzzy representation of the multi-item value in Example 1.*

Observe also that this construction can easily accommodate interval-valued items, using interval arithmetic for computing the sample mean and the lower and upper bounds of the quantiles of the bootstrap distribution. Finally, it is remarked that, as a particular case, a missing item can be represented by an interval spanning the whole domain of the variable, thus this representation is intrinsically able to handle missing data.

3. Evolutionary Graphical Exploratory Analysis

There are many different techniques for performing graphical exploratory analysis of data, as mentioned in the introduction: Sammon maps, Principal Component Analysis (PCA), Multidimensional Scaling (MDS), self-organized maps (SOM), etc.

[6]. These methods project the instances as points in a low dimensional Euclidean space so that their proximity reflects the similarity of their variables. However, we have mentioned that the surveys can be incomplete or contain conflicting answers. Summing up, an incomplete survey is taken as the set of all surveys with any valid value in place of the missing answer. A multi-valued item can also be understood as a set, as we shown in the preceding section. The most immediate consequence of this representation is that the projection of an instance is no longer a point, but a shape whose size will be larger the more incomplete or imprecise the information about the individual is.

In this section, we will describe first the theoretical basis of our generalization of the MDS algorithm to fuzzy data, paying special attention to the differences between our algorithm and its closest precedent in the literature [6]. Our definition of the stress function and the freedom given to the shape of the projections prevent the use of classical optimization techniques, as done in the mentioned reference, thus we make use of a nonstandard Genetic Algorithm, described first in [16]. The main properties of this algorithm are also described in this section.

3.1 Fuzzy MDS

This extension from a map of points to a map of shapes has already been done for some of the techniques mentioned before. For instance, Fuzzy MDS, as described in [6], extends MDS to the case where the distance matrix comprises intervals or fuzzy numbers, as happens in our problem. Crisp MDS consists in finding a low-dimensional cloud of points that minimizes a stress function. That function measures the difference between the matrix of distances among the data and the matrix of distances among the elements of this last cloud. The interval (or fuzzy) extension of this algorithm defines an interval (fuzzy) valued stress function that bounds the difference between the imprecisely known matrix of distances between the objects and the interval (fuzzy) valued distance matrix between a set of shapes in the low-dimensional projection.

Let us assume for the time being that the distance between two surveys is an interval (the extension to the fuzzy case is straightforward, since it suffices to apply the following to each cut of the fuzzy distance). For two imprecisely measured multivariate values $x_i = [x_{i1}^-, x_{i1}^+] \times \ldots \times [x_{if}^-, x_{if}^+]$ and $x_j = [x_{j1}^-, x_{j1}^+] \times \ldots \times [x_{jf}^-, x_{jf}^+]$, with f features each, the set of distances between their possible values is the interval

$$D_{ij} = \left\{ \sqrt{\textstyle\sum_{k=1}^{f} (x_{ik} - x_{jk})^2} \mid x_{ik} \in [x_{ik}^-, x_{ik}^+], x_{jk} \in [x_{jk}^-, x_{jk}^+], 1 \leq k \leq f \right\}. \quad (2)$$

Some authors have used a distance similar to this before [6], however they further assumed that the shape of projection of an imprecise case is always a circle. We have found that, in our problem, this is a too restrictive hypothesis. Instead, we propose to approximate the shape of the projections by a polygon (see Fig. 3, left part) whose radii R_{ij}^+ and R_{ij}^- are not free variables, but depend on the distances between the cases.

For a multivariate sample of imprecise data $(x_1, \ldots, x_N)$, let $\overline{x}_i$ be the crisp centerpoint of the imprecise value x_i (the center of gravity, if an interval, or the modal point, if fuzzy), and let $((z_{11}, \ldots, z_{1r}), \ldots, (z_{N1}, \ldots, z_{Nr}))$ be a crisp projection, with dimension r, of that set. We propose that the radii R_{ij}^+ and R_{ij}^- depend

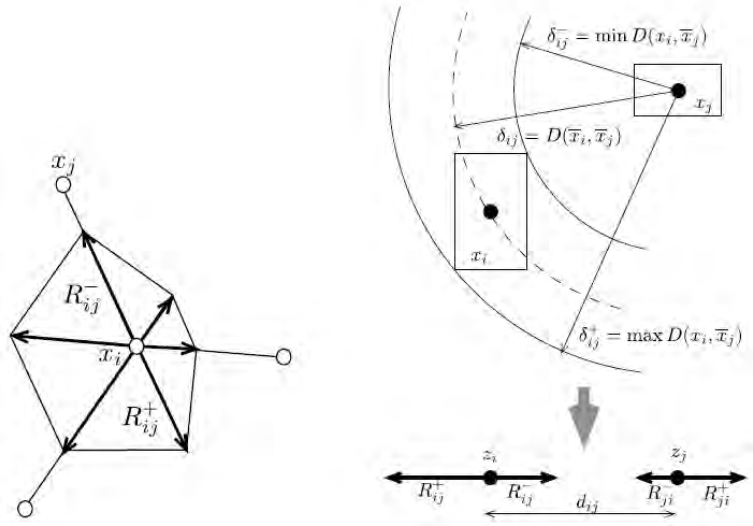


Fig. 3 *Left part: The projected data are polygons defined by the distances R_{ij} in the directions that pairwise join the examples. Right part: The distance between the projections of x_i and x_j is between $d_{ij} - R_{ij}^- - R_{ji}^-$ and $d_{ij} + R_{ij}^+ - R_{ji}^+$.*

on the distance between x_i and $\overline{x}_j$ (see the right part of Fig. 3 for a graphical explanation) as follows

$$R_{ij}^+ = d_{ij}\left(\frac{\delta_{ij}^+}{\delta_{ij}} - 1\right) \quad R_{ij}^- = d_{ij}\left(\frac{\delta_{ij}}{\delta_{ij}^-} - 1\right), \tag{3}$$

where $d_{ij} = \sqrt{\sum_{k=1}^{r}(z_{ik} - z_{jk})^2}$, $\delta_{ij} = \{D(\overline{x}_i, \overline{x}_j)\}$, $\delta_{ij}^+ = \max\{D(x_i, \overline{x}_j)\}$, and $\delta_{ij}^- = \min\{D(x_i, \overline{x}_j)\}$. We also propose that the value of the stress function our map has to minimize is

$$\sum_{i=1}^{N}\sum_{j=i+1}^{N} d_H(D_{ij}, [d_{ij} - R_{ij}^- - R_{ji}^-, d_{ij} + R_{ij}^+ + R_{ji}]^+)^2, \tag{4}$$

where d_H is the Haussdorff distance between intervals.

3.2 Characteristic points

To gain insight into the actual values of the spatial coordinates of the elements displayed in the map, we propose to add several prototypal, fictitious sets of items (we will call them "characteristic points") corresponding to a test without mistakes, a test which is completely wrong, one section well answered but the remaining ones wrong, etc. In the final map, these points will be approximately placed in a circle enclosing the projections of the individuals. With the help of these points, the map can be used for evaluating the capacities of a student or diagnosing a disorder by comparing it with its closest characteristic point.

3.3 Evolutionary algorithm

An evolutionary algorithm is used for searching the map optimizing the stress function (4). In previous works, we have shown that interval and fuzzy fitness functions can be optimized by certain extensions of multiobjective genetic algorithms. In this paper, we have used the extended NGSA-II defined in [16], whose main components are summarized in the following paragraphs.

- **Representation:** Since the shape of each element in the map can be computed given the centerpoints of both the sets and the elements of the map, as described in Section 3.1, each map can be univocally determined from a set of coordinates in the plane, thus each chromosome consists of the concatenation of so many pairs of numbers as individuals, plus one pair for each characteristic point (i.e. "Everything", "Nothing", "Only Subject X", "Every Subject but X", etc). The chromosome is fixed-length, and real coding is used.

- **Objective Function:** The genetic algorithm must minimize the expression defined in eq. (4). However, observe that this equation does not evaluate to a number, but to an interval or a fuzzy set. Generally speaking, one cannot properly define a total order between interval or fuzzy sets, and, therefore, the concept of "minimum" must be replaced by that of "set of minimal elements", which is closely related to the definition of a Pareto front in multicriteria optimization [21]. There exist, however, many different proposals for precedence operators or rankings between fuzzy sets, some of which could be used to define a total order over the solutions and be combined with a suitable scalar evolutionary search [12]. In this work, however, the precedence operator induces a partial order in the set of solutions, thus the search will produce a set of nondominated maps.

- **Evolutionary Scheme:** A generational approach with the multiobjective NSGA-II replacement strategy is considered. Binary tournament selection based on the crowding distance in the objective function space is used. The precedence operator derives from the Bayesian coherent inference with an imprecise prior, the dominated sorting is based on the product of the lower probabilities of precedence, and the crowding in based on the Hausdorff distance, as described in [16].

- **Genetic Operators:** Arithmetic crossover is used for combining two chains. The mutation operator consists in performing crossover with a randomly generated chain.

4. Results

In this section, we will illustrate, with the help of real-world datasets, how to process different kinds of tests and interpret the resulting maps. First, it will be shown how to segment heterogeneous groups of students and how to study the temporal evolution of the learning, which will be represented by arrows. This is useful for finding groups of students that cannot follow the course timeline or those

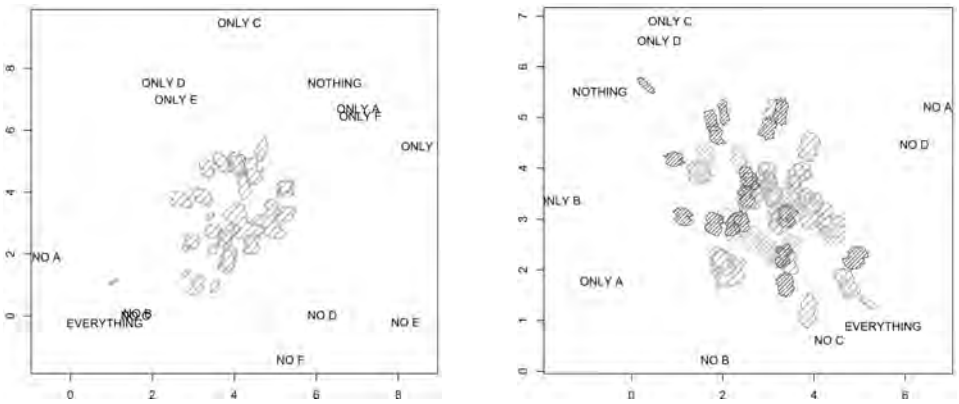


Fig. 4 *Left part: Differences in knowledge of Statistics for students in Ingenieria Telematica. Right part: Differences in knowledge about Computer Science between the students of Ingenieria Tecnica Industrial specialized in Chemistry, Electricity and Mechanics.*

concepts that are learned faster by each group of students. Lastly, we will apply the same techniques to a group of preschoolers with learning disorders and use the techniques developed in this research for analyzing their evolution.

4.1 Knowledge Surveys I: Variation of individual capacities in the same group and between groups

In the left part of Fig. 4, a diagram for 30 students of subject "Statistics" in *Ingenieria Telematica* at Oviedo University, is shown. The data was acquired at the beginning of the course 2009-2010. This survey is related to students' prerrequisite skills in Algebra (A), Logic (B), Electronics (C), Numerical Analysis (D), Probability (E) and Physics (F). The positions of the characteristic points have been marked with labels. Those points are of the type "A" (all the questions about the subject "A" are correct, the others are erroneous) "NO A" (all the questions except "A" ones are correct, the opposite situation), etc.

In the right part of Fig. 4, we have plotted together the results of three different groups, who attend lectures by the same teacher. Each intensification has been coded with a distinctive pattern. This teacher has evaluated, as before, the initial knowledge of the students in subjects that are a prerequisite. From the graphic in that figure, the most relevant fact is that the students of the intensification coded in the less dense pattern *(Ingenieria Industrial)* consider themselves better prepared than those coded in the dark, finer pattern *(Ingeniería Tecnica Industrial Electrica)*, with the other group in an intermediate position, closer to the first one *(Ingeniería Tecnica Industrial Quimica)*. All the students of all the groups have a neutral orientation to math subjects, and some students in the second group think that their background is adequate only in subjects C (Operating Systems) and D (Internet).

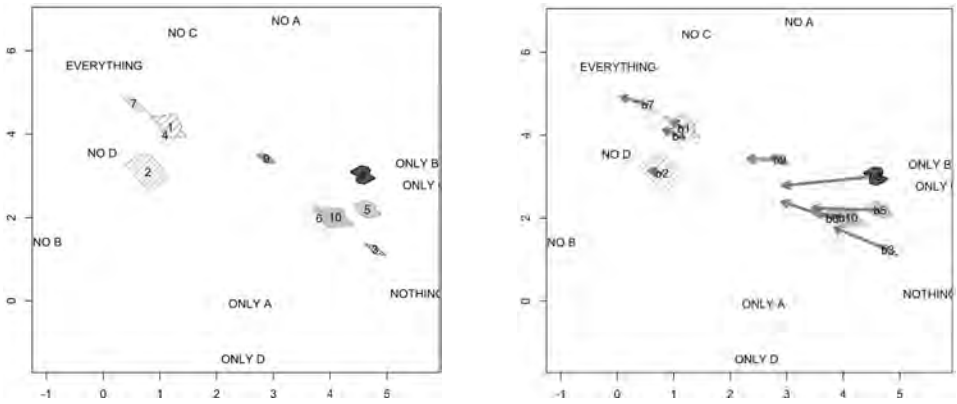


Fig. 5 *Evolution of the learning of pre-doctoral students. Left part: Initial survey. Right part: The displacement has been shown by arrows.*

4.2 Knowledge Surveys II: Evolution of learning capabilities

Ten pre-doctoral students in Computer Science, Physics and Mathematics attending a research master were analyzed. The background of these students is heterogeneous. In the survey, the students were asked 36 questions about the variables "Control Algorithms" (A), "Statistical Data Analysis" (B), "Numerical Algorithms" (C) and "Lineal Models" (D). The left part of Fig. 5 shows that there is a large dispersion between the initial knowledges. Since the course had strong theoretic foundations, students from technical degrees like Computer Science evaluated themselves with the lowest scores (shapes in the right part of each figure).

The same survey, repeated at the end of the course, shows that all the students moved to the left, closer to characteristic point "EVERYTHING". Additionally, the displacement has been larger for the students in the group at the right. This displacement can be seen clearly in the right part of the same figure, where the shapes obtained from the final survey were replaced by arrows that begin in the initial position and end in the final center. The length of the arrows is related with the progress of the student during the course, showing that those students that scored the highest marks in the course (those who also considered themselves best prepared at the beginning of the course) did not make a good use of the course, which, on the contrary, was able to improve the capabilities of students in technical degrees.

4.3 Diagnosis of dyslexia

For this last experiment we have collected a sample of 65 infants between 5 and 6 years old, in urban schools of Asturias (Spain). Afterwards, the same children were examined by a psychologist, who assigned each one of them a class: normal child, dyslexic, slight dyslexia, and attention disorder. In some cases, the child was too young for a definite diagnostic and the expert assigned two classes to them (for example, "might be dyslexia or an attention disorder"). We selected twelve

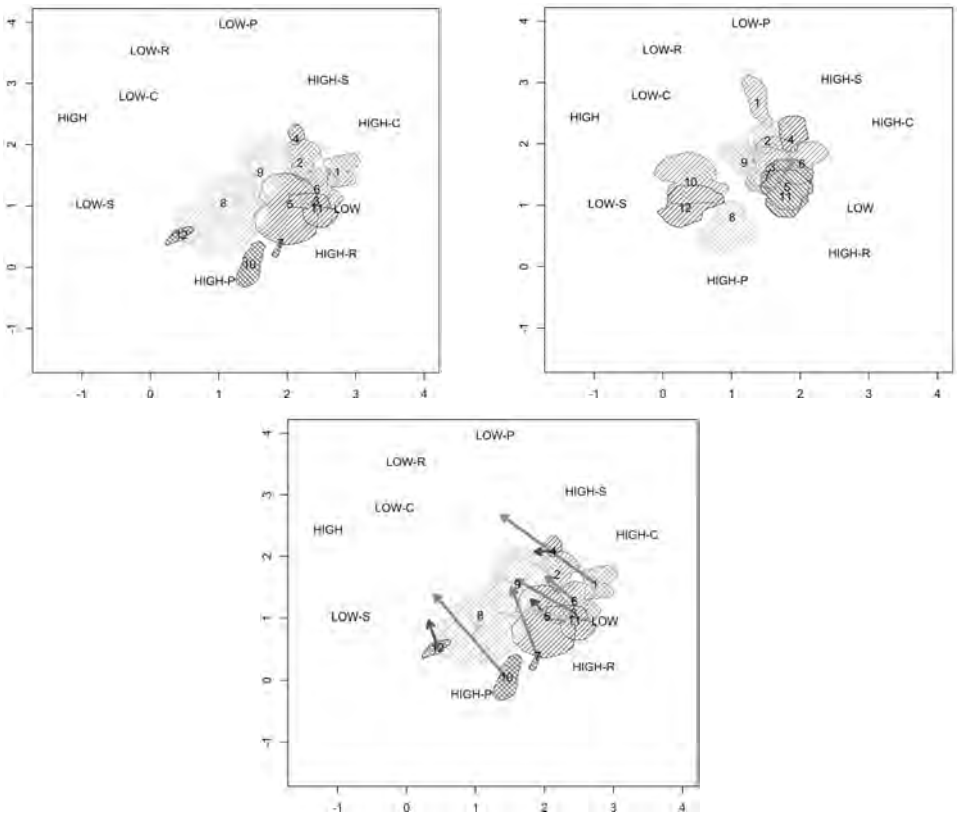


Fig. 6 *Evolution of dyslexia. Upper, left part: 4-5 years. Upper, right: 5-6 years. Lower part: The displacement has been shown by arrows.*

children with potential problems and repeated the tests one year later. We have included the characteristic points of four latent variables: Reasoning (R), Visual-Motor Coordination (C), Shape Perception (P) and Spatial Orientation (S).

In the upper-left part of Fig. 6, the initial map of the children is shown. Children suffering dislexia (individuals 1, 2 and 6) are concentrated on the right part of the map (low scores in all of the latent variables, as indicated by the axis joining the characteristic points "LOW" and "HIGH") and also tend to be in the upper part of the map (lower values in Shape Perception, measured by Bender's tests). The size of some shapes reflects that there is a moderate amount of missing values, however the map shows that the values of the missing items are not too relevant in this stage of the diagnosis, since the intersection of shapes with different classes is low (except for individual number 5, which incidentally had his diagnosis revised one year later). Observe also that the three individuals labeled "dyslexia + attention disorder" (7, 10 and 12) are clearly positioned nearer to the area of attention disorder, and this may indicate that the expert could have used this map for gaining insight in her diagnosis.

The upper-right part of the same figure illustrates the map of these children, one year later. As expected, the skills of all individuals have been enhanced, and all of them are nearer to the characteristic point "HIGH", which is a positive result. This is clearly seen in the lower part of the figure, where both maps are superimposed and the shapes corresponding to the latest test have been removed and replaced by an arrow joining the centers of the initial and final shapes that are the final diagnosis of the expert in this second test. Observe that children 10 and 12 apparently have evolved to the same area of the map, but the expert has labeled the individual number 10 as "dyslexic". In this case, there is a significant overlap between the shapes of these two individuals, and the information given by the tests is too incomplete for being reliable. This last sanity check would have not been possible without the extra information given by the size and shape of the projection that are provided by this method.

5. Conclusions

In this work, we have extended the Multidimensional Scaling to imprecise data, and exploited the new capabilities of the algorithm for producing a method able to process incomplete or non-consistent tests measuring learning capabilities and learning disorders. The map of a group of individuals comprises several shapes whose volumes measure the degree to which a survey has missing data and whose relative positions depend on the similarities between individuals. We have shown with the help of real-world data that these maps can help in detecting heterogeneous groups and measuring the capabilities of the student after the course, and can also be used during the diagnosis of certain learning disorders, being able to condense large amounts of data in a simple graph that permits gaining insight in the evolution of a group of individuals, even when the available data is incomplete or imprecise.

Acknowledgements

This work was funded by the Spanish Ministry of Education, under the grant TIN2008-06681-C06-04.

References

[1] Anderson L. W., Krathwohl D. R., et al. (Eds.): A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Allyn & Bacon. Boston, MA. Pearson. 2001.

[2] Bezdek J. C., Tsao E. C. K., Pal N. R.: Fuzzy Kohonen clustering networks. In: Proc. IEEE Int. Conf. on Fuzzy Systems, 1992, pp. 1035-1043.

[3] Bloom B. S.: Taxonomy of Educational Objectives. The Classification of Educational Goals. Handbook I. – Cognitive Domain, New York, NY: David McKay, 1956.

[4] Cohen P. A.: Student ratings of instruction and student achievement: A meta-analysis of multisection validity studies. Review of Educational Research, **51**, 3, 1981, pp. 281-309.

[5] Couso I., Sánchez L.: Higher order models for fuzzy random variables. Fuzzy Sets and Systems, 159, 2008, pp. 237-258.

[6] Hebert P. A., Masson M. H., Denoeux T.: Fuzzy multidimensional scaling. Computational Statistics and Data Analysis, 51, 2006, pp. 335-359.

[7] Honda K., Ichihashi H.: Fuzzy local independent component analysis with external criteria and its application to knowledge discovery in databases. International Journal of Approximate Reasoning, **42**, 3, 2006, pp. 159-173.

[8] Kim W., Choi B., Hong E-K., Kim S-K.: A Taxonomy of Dirty Data. Data Mining and Knowledge Discovery, 7, 2003, pp. 81-99.

[9] King P. M., Kitchener K. S.: Reflective Judgment: Theory and Research on the Development of Epistemic Assumptions Through Adulthood, Educational Psychologist, **39**, 1, 2004, pp. 5-18.

[10] Kruskal J. B.: Nonmetric multidimensional scaling: a numerical method. Psychometrika, 29, 1964, pp. 115-129.

[11] Lee J. A., Lendasse A., Verleysen M.: Curvilinear Distance Analysis versus Isomap. European Symposium on Artificial Neural Networks ESANN'2002, 2002, pp. 185-192.

[12] Limbourg P.: Multi-objective optimization of problems with epistemic uncertainty. In: EMO 2005: pp. 413-427.

[13] Nuhfer E.: The place of formative evaluations in assessment and ways to reap their benefits. Journal of Geoscience Education, **44**, 4, 1996, pp. 385-394.

[14] Mazza R., Milani C.: Exploring usage analysis in learning systems: Gaining insights from visualisations. In: Workshop on usage analysis in learning systems at 12th International Conference on Artificial Intelligence in Education, New York, USA 1-6 2005.

[15] Palacios A., Sánchez L., Couso I.: Diagnosis of dyslexia with low quality data with genetic fuzzy systems. International Journal on Approximate Reasoning, 51, 2010, pp. 993-1009.

[16] Sanchez L., Couso I., Casillas J.: Modeling vague data with genetic fuzzy systems under a combination of crisp and imprecise criteria. MCDM 2007. Honolulu, Hawaii, USA, 2007.

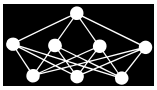
[17] Sánchez L., Couso I., Casillas J.: Genetic Learning of Fuzzy Rules based on Low Quality Data. Fuzzy Sets and Systems, **160**, 17, 2009, pp. 2524-2552.

[18] Shen R., Yang F., Han P.: Data analysis center based on e-learning platform. In: Workshop The Internet Challenge: Technology and Applications, Berlin, Germany 19-28, 2002.

[19] Wirth K., Perkins D.: Knowledge Surveys: The ultimate course design and assessment tool for faculty and students. Proceedings: Innovations in the Scholarship of Teaching and Learning Conference, St. Olaf College/Carleton College, 2005.

[20] Zeki Saka A.: Hitting two birds with a stone: Assessment of an effective approach in science teaching and improving professional skills of student teachers. Social and Behavioral Sciences, **1**, 1, 2009, pp. 1533-1544.

[21] Zitzler E., Thiele L., Bader J.: On Set-Based Multiobjective Optimization. IEEE Transactions on Evolutionary Computation, **14**, 1, 2009, pp. 58-79.



BASE CLASSIFIERS IN BOOSTING-BASED CLASSIFICATION OF SEQUENTIAL STRUCTURES

*Przemyslaw Kazienko, Tomasz Kajdanowicz**

Abstract: Boosting as a very successful classification algorithm represents a great generalization ability with appropriate ensemble diversity. It can be easily applied in the two-class classification problem. However, sequential structure prediction, in which the output is an ordered list of the labeled classes, needs to be realized by an adjusted and extended version. For that purpose the AdaBoostSeq algorithm has been introduced. It performs the multi-class classification with respect to the sequential structure of the classification target. The profile of the AdaBoostSeq algorithm is analyzed in the paper, especially its classification accuracy, using various base classifiers applied to diverse experimental datasets with comparison to other state-of-the-art methods.

Key words: *AdaBoostSeq, boosting, sequence labeling, ensemble methods, multiple classifier system, classification*

Received: 20th September 2010
Revised and accepted: 13th November 2010

1. Introduction

Sequence labeling may be understood as a multi-class classification problem, whose classified target is of a sequential nature. Uniqueness of this classification task lays in the fact that the target labels on some sequence positions may stay in the interaction with one another. In consequence, the knowledge about relations between them may be utilized in the sequence labeling.

Traditional approaches dealing with multi-class classification are realized by reducing the multi-class classification problem to multiple two-class problems. Unfortunately, its application to the sequence labeling problem may result in loss of the information contained in the structure of the classified structure.

For that reason, a new AdaBoostSeq algorithm has been introduced. It naturally extends the original AdaBoost algorithm to the sequence labeling case without

*Przemyslaw Kazienko, Tomasz Kajdanowicz
Wroclaw University of Technology Wybrzeze Wyspianskiego 27, Wroclaw, Poland, E-mail: {kazienko,tomasz.kajdanowicz}@pwr.wroc.pl

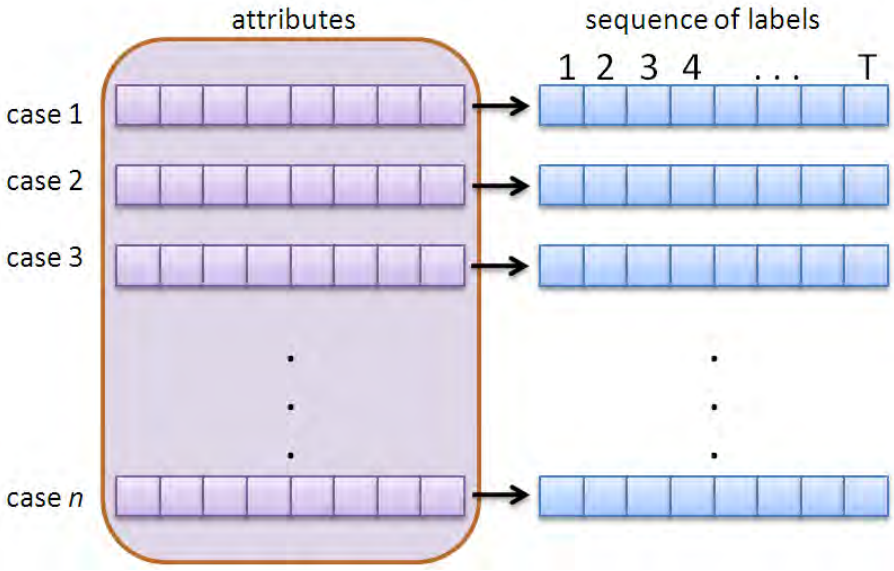


Fig. 1 *Ilustration of sequence labeling problem, where each case (data instance) has assigned sequence of labels.*

reducing it to several two-class problems. Similarly to the AdaBoost applied to the two-class classification, the AdaBoostSeq algorithm also combines some weak base classifiers and it only requires the performance of each weak classifier be better than random guessing.

The concept of the AdaBoostSeq algorithm was introduced in [10]. In this paper, a profile of the AdaBoostSeq algorithm is studied, in particular, the dependence of AdaBoostSeq accuracy on the base classifier selection process. The experiments shown in Section 6. revealed the algorithm's high competitive advantage compared to the best currently available multi-class classification methods.

2. Related Work

The traditional classification assumes each instance (case) belongs to exactly one of a finite set of possible classes. Supposing that a given set of training samples (instances, cases) $(x_1, y_1), \ldots, (x_N, y_N)$, where $x_i \in X^p$ and $y_i \in C$ is a class from finite set of classes, the goal is to find a classification rule $f(x_i)$ from the training data, so that when a new input x_i is given, we can assign to it a class label y_i from the class domain C.

The sequence labeling classification problem considered in this paper allows instances to belong to several classes simultaneously. Classification, in such case, may be addressed by multi-class classification, where classes assigned to the single instance may be represented by a sequence, graph, tree, etc. In general, the goal of mapping the input to the structured output is to assign class values to all elements from the output structure.

The sequence labeling problem, which was studied in many relevant researches [3, 15, 16, 20, 24], refers, among others, to sequence labeling, parsing, collective classification, bipartite matching, and similar.

According to the proposal in [5], it is assumed that classification assigning a sequence to the instance x_i is a type of structured prediction problem being a cost-sensitive classification problem, where classification results y_i have the structure of a vector; y_i may be a sequence of T values (a T-length sequence): y_i=(y_i^1, y_i^2, ..., y_i^T), $\forall$ (μ=1,2,...,T) $y_i^\mu \in C$. Note that the vector notation appears to be useful not only for sequence labeling problems.

Additionally, the algorithms realizing sequence labeling can also make use of the extended idea for feature input space. It means that while computing a given item value y_i^μ, $1 < \mu \leq T$, from the T-length sequence y_i=(y_i^1, y_i^2, ..., y_i^T), the algorithms may utilize the input data both from the original input $x_i \in X$ and from the partially produced output $y_i^{P_\mu}$, where $y_i^{P_\mu}$ is a part of the final y_i obtained so far, e.g. $y_i^{P_\mu}$=(y_i^1, y_i^2, ..., $y_i^{\mu-1}$) [12, 14]. This composition of x_i and $y_i^{P_\mu}$, i.e. $(x_i, y_i^{P_\mu})$ remains an input vector in Euclidean space, but in opposite to the single x_i it also depends on the output $y_i^{P_\mu}$ achieved so far. This concept makes use of the typical nature of the sequential data, in which a given item value y_i^μ may depend somehow on the values of the previous items in the sequence, namely $y_i^1, y_i^2, ..., y_i^{\mu-1}$.

Overall, structured prediction is a research problem that emerges in many application domains, among others in protein function classification [27], semantic classification of images [2] or text categorization [19].

Generally, structured prediction methods can be categorized into two different groups [25]: problem transformation methods, and algorithm adaptation methods. Whereas the former group of methods is independent of algorithms and concerns the transformation of multi-class classification task into one or more single-class classification, the latter adapts existing learning algorithms in order to directly handle multi-class data. This paper focuses on the second group of methods.

As the nature of structured prediction problems is complex, the majority of proposed algorithms is based on the well-known binary classification adapted in a specific way [17]. The most natural adaptation is structured perceptron [3] that has minimal requirements on output space shape and is easy to implement. However, it provides somewhat limited generalization accuracy. An example adaptation of the popular backpropagation algorithm is BPMLL [27], where a new error function takes multiple target into account.

Another solution are Max-margin Markov Nets that consider the structured prediction problem as a quadratic programming problem [20]. They are very useful, however, their performance is very slow. The next, more flexible approach is an alternative adjustment of logistic regression to the structured outputs called Conditional Random Fields [15]. It provides probabilistic outputs and good generalization, but again, it is relatively slow. Another method, similar to Max-margin Markov Net technique, is Support Vector Machine for Interdependent and Structured Outputs (SVM^{STRUCT}) [24], which applies variety of loss functions.

Yet other known algorithms from a lazy learning group realizing the structured prediction task are MLkNN and BRkNN [28]. Both of them extend the popular k Nearest Neighbors (kNN) lazy learning algorithm using a Bayesian approach and the maximum a posteriori principle to assign the label set based on prior

and posterior probabilities for the frequency of each label within the k nearest neighbors. An alternate, based on meta learning approach, are Hierarchical Multi-label Classifier (HMC), HOMER [22] that construct a hierarchy of multi-label classifiers and RAkEL [26], an ensemble of classifiers trained with the applications of different small random subset of the set of labels.

The main motivation in designing a new ensemble method for sequence labeling is to utilize the powerful machine learning concept like boosting and apply it not independently to individual sequence items but force the method to make use of the additional knowledge of the structure being predicted [10].

3. Boosting for Sequence Labeling

Based on the most popular boosting algorithm AdaBoost [7, 21], the modification to the cost function as well as the new structure of sequential increments has been introduced to the algorithm.

It is assumed that there is a binary sequence classification problem with $y_i^\mu \in \{-1, 1\}$, for $i = 1, 2, \ldots, N$ and $\mu = 1, 2, \ldots, T$, where N is the number of instances (observations, cases), T is the length of the sequence. The general goal is to construct T optimally designed linear combinations of K base classifiers of the form:

$$\forall \mu = 1, 2, \ldots, T \quad F^\mu(x) = \sum_{k=1}^{K} \alpha_k^\mu \Phi(x, \Theta_k^\mu) \tag{1}$$

where: $F^\mu(x)$ is the combined final meta classifier for the μ-th sequence item; $\Phi(x, \Theta_k^\mu)$ represents the k-th base classifier, performing according to its Θ_k^μ parameter and returning a binary class label for each instance (case) x; α_k^μ is the weight associated to the k-th classifier.

Values of the unknown parameters (α_k^μ and Θ_k^μ) result from minimization of prediction error for each μth sequence element for all K classifiers. As the direct optimization of these both parameters is highly complex, a stage-wise suboptimal method is performed in M steps [21]. By definition of partial sums and based on recursion properties, the value of $F^\mu(x)$ at step m, i.e. $F_m^\mu(x)$, may be calculated using the value of $F_{m-1}^\mu(x)$ that has already been optimized in the previous step $m-1$, see [11] for details. Therefore, the problem at step m is to compute:

$$(\alpha_m^\mu, \Theta_m^\mu) = \arg\min_{\alpha,\Theta} J(\alpha^\mu, \Theta^\mu), \tag{2}$$

where the sequence-loss balancing cost function J is defined as:

$$J(\alpha^\mu, \Theta^\mu) = \sum_{i=1}^{N} \exp(-y_i^\mu(\xi F_{m-1}^\mu(x_i) + (1-\xi) y_i^\mu \hat{R}_m^\mu(x_i) + \alpha^\mu \Phi(x_i, \Theta^\mu))), \tag{3}$$

where: $\hat{R}_m^\mu(x_i)$ is an impact function denoting the influence on prediction according to the quality of the preceding sequence labels predictions; ξ is a parameter that

allows controlling the influence of impact function in weights composition, $\xi \in \langle 0, 1 \rangle$.

$\hat{R}^{\mu}_{m}(x_i)$ is applied in computation for the current sequence position, as follows:

$$\hat{R}^{\mu}_{m}(x_i) = \sum_{j=1}^{m-1} \alpha^{\mu}_{j} R^{\mu}(x_i) \tag{4}$$

$$R^{\mu}(x_i) = \frac{\sum_{l=1}^{\mu} y^{l}_{i} \frac{F^{l}(x_i)}{\sum_{k=1}^{K} \alpha^{l}_{k}}}{\mu}, \tag{5}$$

where: $R^{\mu}(x_i)$ is the auxiliary function that denotes the average coincidence between prediction result $F^{l}(x_i)$ and the actual value y^{l}_{i} weighted with the weights α^{l}_{k} associated to the k-th base classifiers for all sequence items achieved so far (from 1 to μ) with respect to the value of μ.

The impact function $\hat{R}^{\mu}_{m}(x_i)$, introduced in Eq. 4 and 5, measures the correctness of prediction for all preceding labels $l = 1, \ldots, \mu$ in the sequence. This function is utilized in the cost function and it provides a smaller error deviation for the whole sequence. The greater compliance between prediction and the real value is, the higher the function value is. Due to the binary nature of the base classifier, minimization of Θ^{μ} (from Eq. 3) is equivalent to:

$$\Theta^{\mu} = \arg\min_{\Theta^{\mu}} \left\{ \sum_{i=1}^{N} w^{\mu}_{i(m)} I(1 - y^{\mu}_{i} \Phi(x_i, \Theta^{\mu})) \right\}, \tag{6}$$

where

$$I(x) = \begin{cases} 0, if\ x = 0 \\ 1, if\ x > 0 \end{cases} \tag{7}$$

and $w^{\mu}_{i(m)}$ denotes the weight associated to instance x_i in the m-th step, see [11] for its detailed derivation.

The presented transformation based on basic algebraic operations lead to the algorithm AdaBoostSeq for sequence prediction and may be found in [10].

4. Base Classifiers for the AdaBoostSeq

A crucial component of boosting scheme is a construction of good base classifiers providing core classification. These base classifiers are weighted while undertaking the final classification decision. However, a base classifier that is too weak cannot guarantee high performance on composite generalization. In the context of binary classification, that is, in fact, the most basic operation performed by the AdaboostSeq, it is required the weighted empirical error of each base classifier be smaller than $\frac{1}{2} - \frac{1}{2}\gamma$, where γ is a parameter quantifying the deviation of the base classifier performance. Thus, considering a base classifier Φ obtained from learning on the dataset with all instances (x_i, y^{μ}_{i}), for $i = 1, 2, \ldots, N$ and for $\mu = 1, 2, \ldots, T$ (all sequence items), we expect the empirical error $\epsilon(\Phi(x, \Theta^{\mu}))$ for all cases x in the learning set to be as follows:

$$\epsilon(\Phi(x,\Theta^{\mu})) = \sum_{i=1}^{N} w_i^{\mu} I(y_i^{\mu} \neq \Phi(x_i,\Theta^{\mu})) \leq \frac{1}{2} - \frac{1}{2}\gamma, \tag{8}$$

where w_i^{μ} is the weight associated with instance x_i at μ sequence item, $\gamma > 0$, $I(true) = 1$ and $I(false) = 0$.

For the real world data, while utilizing very simple base classifiers, it may be a very difficult or even impossible task to find such γ, for which Eq. 8 holds. For example, let us consider the two-dimension *xor* problem with $N = 4$ and data instances: $x_1 = (1,1)$, $x_2 = (1,-1)$, $x_3 = (-1,1)$, $x_4 = (-1,-1)$. Clearly, it will not be solved by an axis-parallel half-space, such that ϵ will be smaller than $\frac{1}{2}$ for uniform weighting over data instances.

On the contrary, an excessively complex base classifier may lead to overfitting and can cause drop of the total performance. Again, for the real world data, where a higher noise level usually appears, the usage of complex base classifiers may result in exaggerated weighting for instances belonging to the noise.

As the appropriate choice of the base classifier plays a key role in successful application, it is desirable to empirically examine basic properties of the AdaBoostSeq in terms of base classifier selection.

5. Experiments

The main objective of the performed experiments was to discover the profile of AdaBoostSeq in terms of its accuracy dependency on the type of the base classifier used in classification. The AdaBoostSeq algorithm was examined according to hamming loss, classification accuracy and computation time for five distinct base classifiers (Decision Stump, C4.5, Naive Bayes, Logistic Regression and Support Vector Machine – SVM) together with the other state-of-the-art representative algorithms in the structured prediction domain, namely BPMLL, MLkNN, BRkNN, HMC, HOMER and RAkEL. The parameters of the base classifiers utilized in the experiments are shown in Tab. I.

Algorithm	Settings
Decision Stump	entropy based
Decision Tree C4.5	conf.factor=0.25; pruned;
Naive Bayes	kernel estimator
Logistic Regression	max Its.=-1; ridge=10^{-8}
Support Vector Machine (SVM)	polynomial kernel; exp=1; size=250007

Tab. I *Settings of the base classifiers utilized in the experiments.*

As the nature of structured prediction differs from the standard approaches, it requires different evaluation measures than those used in the traditional single-label classification. Some standard evaluation measures of multi-class classifiers from the previous work have been used in the experiments. The utilized measures are calculated based on the differences of the actual and the predicted sets of labels

over examples. The first employed measure, proposed in [19], is Hamming Loss HL, which is defined as:

$$HL = \frac{1}{N}\sum_{i=1}^{N}\frac{Y_i \triangle F(x_i)}{|Y_i|}, \tag{9}$$

where: N is the number of examples, Y_i denotes actual (real) labels in the sequence, $F(x_i)$ is a sequence of labels predicted by the classifier, and $\triangle$ stands for the symmetric difference of two sets, which is the set-theoretic equivalent of the exclusive disjunction in Boolean logic.

The second evaluation measure utilized in the experiments is Classification Accuracy CA [9], defined as:

$$CA = \frac{1}{N}\sum_{i=1}^{N} I(Y_i = F(x_i)), \tag{10}$$

where: $N, Y_i, F(x_i)$ have the same meaning as in Eq. 9, $I(true)=1$ and $I(false)=0$.

This is a very strict evaluation measure as it requires the predicted sequence of labels to be an exact match of the true set of labels.

The performance of the analyzed methods was evaluated using 10-fold cross-validation and the evaluation measures from Eq. 9 and Eq. 10. These two metrics are widely-used in literature and are indicative for the performance of multi-label classification methods. Additionally, the computation time has been monitored.

The experiments were carried out on three datasets from three diverse application domains: semantic scene analysis, bioinformatics and music categorization. The image dataset *scene* [2] semantically indexes still scenes. The biological dataset *yeast* [6] is concerned with protein function classification. The music dataset *emotions* [23] contains data about songs categorized into one or more classes of emotions.

	Dataset	Examples	Attributes	Sequence length
1	scene	2407	294	6
2	yeast	2417	203	14
3	emotions	593	72	6

Tab. II *Datasets used in the experiments.*

Basic statistics of utilized datasets, such as the number of examples, the number of numeric and discrete attributes and the length of label sequence are presented in Tab. II. In each of examined datasets, the binary classification problem was addressed.

6. Results

The sequence labeling for three diverse datasets was examined in the experiment. First, different base classifiers within the AdaBoostSeq were evaluated in compar-

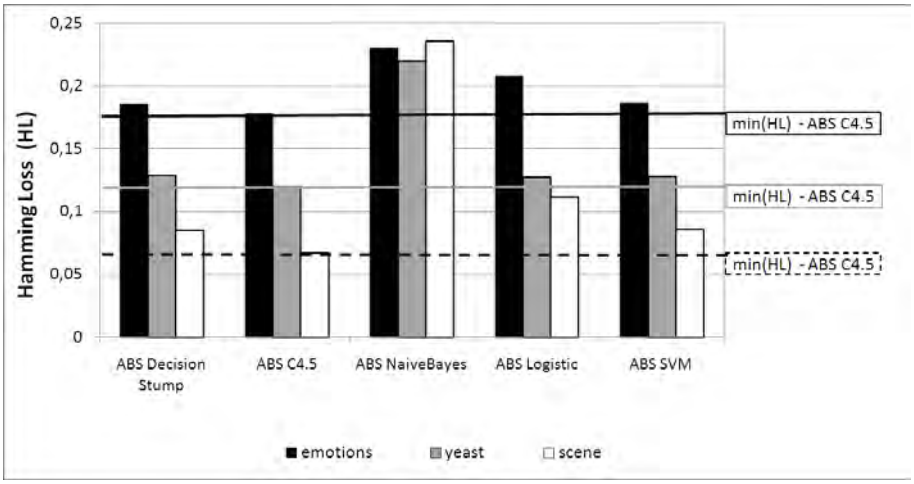


Fig. 2 *Hamming Loss measure for examined base classifiers in the AdaBoostSeq algorithm on scene, yeast and emotions datasets.*

ison with one another, see Section 6.1. Next, the results of AdaBoostSeq were confronted against some other methods, see Section 6.2.

6.1 Evaluation of base classifiers

The AdaBoostSeq algorithm (abbreviated in figures to ABS) was applied separately to three datasets *scene*, *yeast*, and *emotions* using five diverse base classifiers: Decision Stump, Decision Tree C4.5, Naive Bayes, Logistic Regression, and Support Vector Machine. The best value of Hamming Loss HL, Eq. 9, was achieved for Decision Tree C4.5 as the base classifier, Fig. 2. Moreover, it referred all three datasets.

The results obtained by the AdaBoostSeq for HL measure for the most simple base classifiers, i.e. Decision Stump, Naive Bayes as well as with the most complex ones (SVM) are from 5% to 80% worse than for C4.5 base classifier. The worst base classifier was Naive Bayes.

Similar results were obtained for the accuracy measure CA, Eq. 10, see Fig. 3. Decision trees C4.5 performed better from 4% to 29% compared to the other base classifiers. The Naive Bayes classifier was the worse for *emotions*, *yeast* and *scene* datasets (the average value of accuracy CA at the level of only 24%, 11% and 16%, respectively).

The above results confirm, the general boosting property of underfitting for simple classifiers (Decision Stump, Naive Bayes) and overfitting for complex ones (SVM), as mentioned in Section 4.

Concluding, it appears that the balanced base classifiers like decision trees provide the best results for the AdaBoostSeq method.

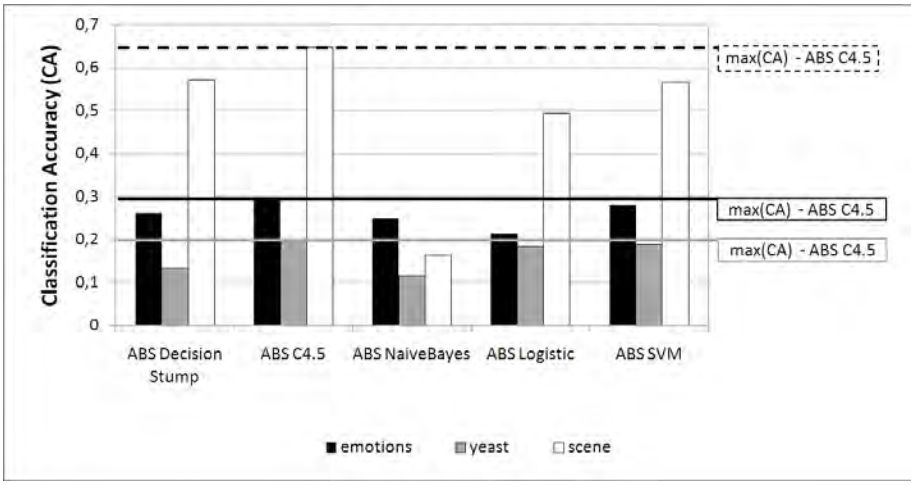


Fig. 3 *Classification Accuracy measure for the examined base classifiers in the AdaBoosSeq algorithm on scene, yeast and emotions datasets.*

6.2 Comparison of AdaBoostSeq with other methods

The AdaBoostSeq algorithm was also compared to some other methods, in particular to BPMLL, MLkNN, BRkNN, HMC, HOMER, and RAkEL.

As regards Hamming Loss HL, Eq. 9, the best AdaBoostSeq (ABS) with C4.5 base classifier performed better by 1% than the second best MLkNN, and by 45% better than the worst BPMLL for the *scene* dataset. In the dataset *yeast*, ABS C4.5 provided 34% better results compared to the second best MLkNN, and by 43% better than the worst RAkEL. In the last dataset *emotions*, the AdaBoostSeq resulted in a 13% better prediction than BPMLL, and was by 29% better than the worst MLkNN, see Fig. 4.

It is worth mentioning that while the other algorithms examined on all three datasets are not resistant to the profile of data (e.g. MLkNN provides a fair prediction only for some of them), the AdaBoostSeq with C4.5 base classifier appears to be rather resistant. It is a sign of stability of the AdaBoostSeq method when using a balanced base classifier.

While considering the classification accuracy CA, Eq. 10, again the AdaBoostSeq with C4.5 base classifier provided the best performance in comparison to all other algorithms, see Fig. 5.

The experimental results confirm that the nature of the sequence-loss cost function, Eq. 3, accompanied by an appropriate, balanced base classifier utilized within the AdaBoostSeq algorithm, promotes minimization of the error on the individual sequence item rather than minimization of the error for the whole sequence at once. As a result, we receive competitive outcome.

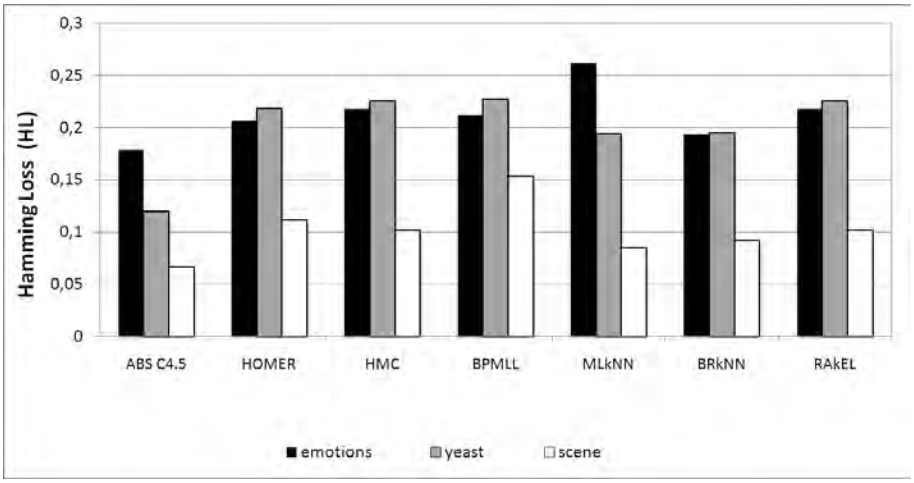


Fig. 4 *Hamming Loss measure for the examined algorithms on scene, yeast and emotions datasets.*

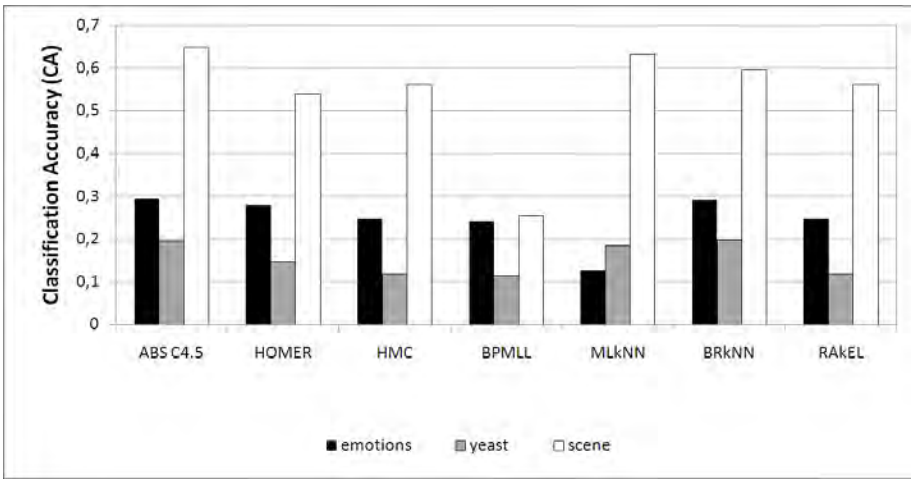


Fig. 5 *Classification Accuracy measure for the examined algorithms on scene, yeast and emotions datasets.*

6.3 Time efficiency

Unfortunately, the AdaBoostSeq algorithm with any of the tested base classifiers requires much more computational effort in comparison to other algorithms, see Fig. 6. Furthermore, decision trees C4.5 and logistic regression taken as base classifiers appear to be the most demanding in terms of computation time. Among AdaBoostSeq approaches, the method using the simple Naive Bayes base classifier was the fastest. Moreover, its execution time was smaller than that of some other classification methods like HOMER, HMC, and RakEL.

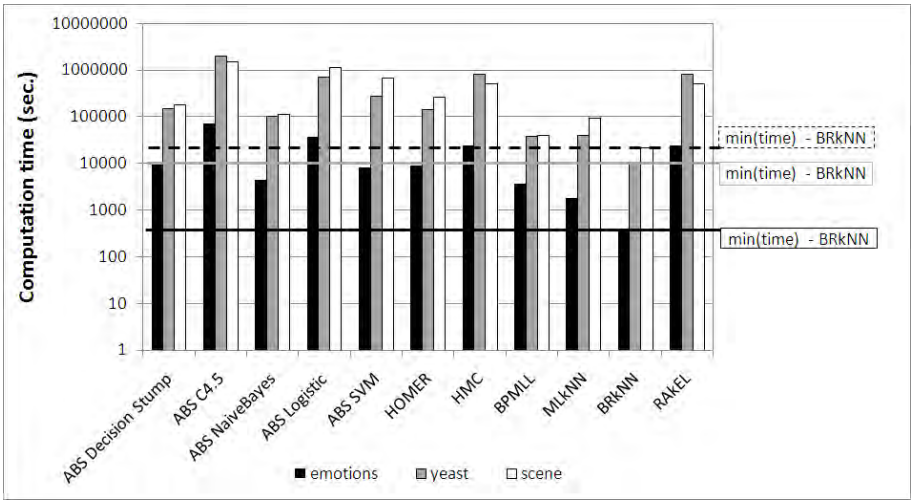


Fig. 6 *Execution time in seconds for the examined algorithms on scene, yeast and emotions datasets (ABS – AdaBoostSeq).*

Overall, we can state that the best balanced classifier – decision tree C4.5 is simultaneously the most time consuming.

7. Conclusions

The usage of the proper base classifier in a new approach to sequence labeling – the AdaBoostSeq algorithm, which is based on a boosting concept, was analyzed in the paper.

The experimental studies were carried out on several known benchmark datasets using diverse base classifiers. Additionally, the AdaBoostSeq algorithm was compared to other state-of-the-art algorithms in the structured prediction domain.

The results have shown that the AdaBoostSeq is a valuable and useful alternative approach that may provide the best and most accurate classification results if an appropriate, balanced base classifier is used. Both simple and complex base classifiers performed worse compared to the best balanced decision tree C4.5 all the experiments conducted.

The AdaBoostSeq method may be more or less resistant to under- and overfiting when utilizing diverse base classifiers. Overall, the balanced base classifiers enable the AdaBoostSeq to provide the most accurate outcomes, which are better than those of other recently known methods.

A significant disadvantage of the best classifier C4.5 was its long processing time.

Acknowledgement

This work was supported by The Polish Ministry of Science and Higher Education, under the development project 2009-11, and research project 2010-13.

References

[1] Altun Y., Hofmann T., Johnson M.: Discriminative Learning for Label Sequences via Boosting, Advances in Neural Information Processing Systems, 15, 2009, pp. 1001-1008.

[2] Boutell M., Luo J., Shen X., Brown C.: Learning multi-label scene classification, Pattern Recognition, 37, 2004, pp. 1757-1771.

[3] Collins M.: Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In: Conference on Empirical Methods in Natural Language Processing 2002, vol. **10**, 2002, pp. 1-8.

[4] Daume H.: Practical Structured Learning Techniques for Natural Language Processing, Ph.D. thesis, University of Southern California, Los Angeles, CA, USA, 2006.

[5] Daume H., Langford J., Marcu D.: Search-based structured prediction, Machine Learning, 75, 2009, pp. 297-325.

[6] Elisseeff A.,Weston J.: A kernel method for multi-labelled classification, Advances in Neural Information Processing Systems, 14, 2001.

[7] Freund Y., Schapire R.: A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences, 55, 1997, pp. 119-139.

[8] Friedman J., Hastie T., Tibshirani R.: Additive logistic regression: a statistical view of boosting, The Annals of Statistics, **28**, 2, 2000, pp. 337-407.

[9] Ghamrawi N., McCallum A.: Collective multi-label classification. In: Proceedings of the 3005 ACM Conference on Information and Knowledge Management 2005, 2005, pp. 195-200.

[10] Kajdanowicz T., Kazienko P., Kraszewski J.: Boosting Algorithm with Sequence-Loss Cost Function for Structured Prediction. In: 5th International Conference on Hybrid Artificial Intelligence Systems HAIS 2010, Lecture Notes in Artificial Intelligence LNAI 6076, Springer, 2010, pp. 573-580.

[11] Kajdanowicz T., Kazienko P.: Boosting-based Sequence Prediction, New Generation Computing, vol. **29**, no. 3, in press, 2011.

[12] Kajdanowicz T., Kazienko P.: Incremental Prediction for Sequential Data. In: The 2nd Asian Conference on Intelligent Information and Database Systems 2010, Lecture Notes in Artificial Intelligence LNAI 5991, Springer, 2010, pp. 359-367.

[13] Kajdanowicz T., Kazienko P.: Hybrid Repayment Prediction for Debt Portfolio. In: The 1st International Conference on Computational Collective Intelligence – Semantic Web, Social Networks and Multiagent Systems 2009, Lecture Notes in Artificial Intelligence LNAI 5796, Springer, 2009, pp. 850-857.

[14] Kajdanowicz T., Kazienko P.: Prediction of Sequential Values for Debt Recovery. In: The 14th Iberoamerican Congress on Pattern Recognition CIARP 2009, Lecture Notes in Computer Science LNCS 5856, Springer, 2009, pp. 337-344.

[15] Lafferty J., McCallum A., Pereira F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning ICML 2001, Morgan Kaufmann, 2001, pp. 282-289.

[16] McCallum A., Freitag D., Pereira F.: Maximum entropy Markov models for information extraction and segmentation. In: Proceedings of the 17th International Conference on Machine Learning ICML 2000, Morgan Kaufmann, 2000, pp. 591-598.

[17] Nguyen N., Guo Y.: Comparisons of Sequence Labeling Algorithms and Extensions. In: Proceedings of the 24th International Conference on Machine Learning ICML 2007, Morgan Kaufmann, 2007, pp. 681-688.

[18] Punyakanok V., Roth D.: The use of classifiers in sequential inference, Advances in Neural Information Processing Systems, 13, 2001, pp. 995-1001.

[19] Schapire R. E., Singer Y.: Boostexter: a boosting-based system for text categorization, Machine Learning, 39, 2000, pp. 135-168.

[20] Taskar B., Guestrin C., Koller D.: Max-margin Markov networks, Advances in Neural Information Processing Systems, 16, 2004, pp. 25-32.

[21] Theodoris S., Koutroumbas K.: Pattern Recognition, Elsevier, 2009.

[22] Tsoumakas G., Katakis I., Vlahavas I.: Effective and Efficient multi-label Classification in Domains with Large Number of Labels. In: ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08), 2008.

[23] Trohidis K., Tsoumakas G., Kalliris G., Vlahavas I.: Multi-label Classification of Music into Emotions. In: Proceedings of the 9th International Conference on Music Information Retrieval ISMIR 2008, 2008, pp. 325-330.

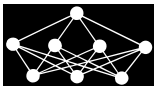
[24] Tsochantaridis I., Hofmann T., Thorsten J., Altun Y.: Large margin methods for structured and interdependent output variables, Journal of Machine Learning Research, 6, 2005, pp. 1453-1484.

[25] Tsoumakas G., Katakis I.: Multi-label classification: An overview, International Journal of Data Warehousing and Mining, 3, 2007, pp. 1-13.

[26] Tsoumakas G., Vlahavas I.: Random k-Labelsets: An Ensemble Method for multi-label Classification. In: Proceedings of the 18th European Conference on Machine Learning ECML 2007, Lecture Notes in Computer Science 4701, Springer, 2007, pp. 406-417.

[27] Zhang M. L., Zhou Z. H.: Multi-label neural networks with applications to functional genomics and text categorization, IEEE Transactions on Knowledge and Data Engineering, 18, 2006, pp. 1338-1351.

[28] Zhang M. L., Zhou Z. H.: Ml-knn: A lazy learning approach to multi-label learning, Pattern Recognition, 40, 2007, pp. 2038-2048.



COMBINATION OF ONE-CLASS CLASSIFIERS FOR MULTICLASS PROBLEMS BY FUZZY LOGIC

*Tomasz Wilk, Michał Woźniak**

Abstract: Combining classifiers, so-called Multiple Classifier Systems (MCSs), gained a lot of interest has recent years. Researchers, developed a large variety of methods in order to exploit strengths of individual classifiers. In this paper, we address the problem of how to implement a multi-class classifier by an ensemble of one-class classifiers. To improve performance of a compound classifier, different individual classifiers (which may, e.g., differ in complexity, type, training algorithm or other) can be combined and that could increase its both performance, and robustness. The model of one-class classifiers can only recognize one of the classes, therefore, it is quite difficult to produce MCSs on the basis of one-class classifiers. Thus, we introduce a new scheme for decision-making in MCSs through a fuzzy inference system. Specifically, we address two important open problems in the context: model selection and combiner training. Classifiers' outputs as supports for given classes are combined by means of a fuzzy engine. Thus, we are interested in such individual classifiers which can return support for given classes. There are no other restrictions on the used classifiers. The proposed model has been evaluated by computer experiments on several benchmark datasets in the Matlab environment. Their results prove that fuzzy combination of binary classifiers may be a valuable classifier itself. Additionally, there are indicated both some application areas of the models, and new research frontiers to be examined.

Key words: *Multiple classifier systems, pattern recognition, fuzzy logic, one-class classifiers*

Received: 20th September 2010
Revised and accepted: 13th November 2010

1. Introduction

There is much current research into developing even more efficient and accurate recognition algorithms. Multiple classifier systems (MCSs), known as combining

*Michał Woźniak
Department of Systems and Computer Networks, Wroclaw University of Technology, E-mail: michal.wozniak@pwr.wroc.pl

classifiers, are currently the focus of intense research. In this approach, the main effort is concentrated on combining knowledge of the set of individual classifiers. The main motivations of using MCSs are as follows:

- It could avoid selection of the worst classifier by e.g. averaging the individual ones [20].
- There is ample evidence that combination of classifiers can result in a classifier that outperforms the best individual.
- Many machine learning algorithms use heuristic search algorithms which do not guarantee that optimal solution is found. Exhaustive search, i.e. testing the whole space of possible solutions for most decision problems, is impossible; therefore, the combining approach which starts the machine learning algorithm from different points is an attractive proposition.
- Combined classifier could be used in distributed environment, especially in the case that database is partitioned for privacy reasons.
- According to the "no free-lunch theorem" there is not a single solution which could solve all problems, but classifiers have different domains of competence [20].

There is a number of important issues while building the aforementioned systems. The problem of designing compound classifiers consists of three main areas:

- topology,
- classifier ensemble design,
- fuser design.

As a topology parallel one is the most popular because it has a good methodological background.

Another important issue while building MSCs is how to select classifiers in a way making the quality of ensemble better than quality of individual classifier. Let us notice that combining similar classifiers could not contribute much to the system being constructed, apart from increasing the computational complexity. That is why it is important to select members of a committee with possibly different components. One of current research is trying to answer the question how the diversity could be measured. Proposed methods exploit several types of diversity measures which, for example, can be used to minimize the possibility of coincidental failure by different classifiers in the ensemble [18].

A strategy for generating the ensemble members must seek to improve the ensemble's diversity. We could use varying components of the MCS to enforce classifier diversity:

- using different input data, e.g. we could use different partitions of data set or generate various data sets by data splitting, cross-validated committee, bagging, boosting [23], because we hope that classifiers trained on different inputs are complementary;

- using classifiers with different outputs, i.e. each individual classifier could be trained to solve subset of M class problem (e.g. binary classifier – one class against remaining ones strategy) and fusion method should recover the whole set of M classes. The well-known technique is Error-Correcting Output Codes (ECOC) [6];

- using classifiers with the same input and output, but trained on the basis of different models or model's versions.

Another important issue is the choice of a collective decision-making method. There are many different voting methods like majority voting [35] and more advanced types based on weighting the importance of decisions coming from particular committee members [23, 33]. Treating the process of weight selection as a separate learning process is an alternative method [15, 16].

The second group of collective decision-making methods bases on supports given by individual classifiers for each given classes, the main form of which are the posterior probability estimators, associated with probabilistic models of a given pattern recognition task [1, 5, 17]. One also has to mention many other works that describe analytical properties and experimental results, like [9, 12, 32]. The aggregating methods, which do not require a learning procedure, use simple operators, like average, maximum, minimum, or product, but they are typically subject to very restrictive conditions [8], which severely limits their practical use. Therefore, the design of new fusion classification models are currently the focus of intense research.

The purpose of our contribution is to present a theoretical model of fuzzy combination method of one-class classifiers. Its implementation is elaborated and validated. We prove efficiency of the proposed scheme through a series of tests and discuss next steps for the future research and improvement areas.

2. Related works

Fuzzy set theory can be successfully used when dealing with uncertainty in decision-making. Thus, fuzzy sets gained attention on many research frontiers such as information technology, production support, decision-making, pattern recognition, diagnostics, data analysis, etc. [4, 11, 22, 23, 24, 34, 37].

Kuncheva *et al.* [22] report that the Fuzzy Integral (FI) gives excellent results as a classifier combiner. Its main idea is to measure the competency of the collections of classifiers, instead of measuring competency of only single classifiers. The measure of strength is defined as a Fuzzy Measure. Lee *et al.* [19] show that usage of FI allows us to use relative weight of each of individual classifiers. One must also mention Decision Templates (DT) with fuzzy measure as a similarity measure [22]. During a classification process, every DT is compared with decision profile of an input object. Kuncheva *et al.* [22] proof that DT supported by the fuzzy logic give good results.

The other fuzzy combination method involves usage of neuro-fuzzy systems. Fuzzy systems implemented as adaptive neural networks (ANNs) are fuzzy systems, which use neural networks support in their properties determination process (for

both fuzzy sets and rules). Therefore, neuro-fuzzy systems harness the power of the two paradigms: fuzzy logic and ANNs, by combining learning ability of neural networks with the strength of the tuned and approximated human logic to process uncertain information [27]. Good example of such a system is the Adaptive Neuro-Fuzzy Inference System (ANFIS) which shows good results in modeling nonlinear functions. ANFIS learns the membership function parameters from a data set with features corresponding to a given problem [11].

Güler *et al.* [11] use model based on ANFIS to classify EEG signals. ANFIS based classifiers were trained on different sets of features and finally combined. Zeng *et al.* [37] reports successful implementation of ANFIS with genetic algorithms support for toughening of materials.

A multi-class classification problem can be decomposed in the finite quantity of two-class classification problems [37]. Thus, connecting binary classifiers should aim to solve multi-class problem by dividing it into dichotomies. In literature there are several examples of construction of the multi-class classifier by combining the outputs of two-class classifiers [7, 14, 29].

Usually the combination is made via a simple nearest-neighbor rule, which finds the class that is closest in some sense to the outputs of the binary classifiers. The most common variations of binary classifier combinations are: one-against-one and one-against-all [7]. The latter allows one to create neat and intuitive multi-class classifier. In this model, at least one binary classifier corresponds to each class. The hypothesis that the given features vector belongs to the selected class is tested against it belonging to one of the other classes. Such an approach has a flow in a case of conflicting answers from classifiers which is not quite straightforward. One-against-all method is usually implemented as so called Winner Takes All (WTA). Each classifier is trained on instances of different class which becomes first class, all the other classes correspond to the second one. Final result is achieved by the maximum rule on the values of support for every class. Dieterich and Bakiri [6] propose a combination model, which in case of binary classifier ensembles appeared to be a good extension of approaches mentioned above. Each sequence of bits produced by a set of binary classifiers is associated with codewords during learning. The ECOC method selects a class with the smallest Hamming distance to its codeword. Passerini *et al.* [26] used successfully this scheme for support vector machines.

On the other hand, the combination of one class classifiers still awaits proper attention [10]. One-class classification problem, also called data description, is a special case of binary classification [28]. Their main goal is to detect anomaly or a state other than the one for the target class [30]. It is assumed that only information of one of the classes, the target class, is available. The task is to define a boundary around the target class, such that it accepts as much of the target objects as possible, while it minimizes the chance of accepting outlier objects.

3. Fuzzy Combiner

3.1 Model of proposed combiner

In the modeling task, we concentrate on both the parameter, and the structure identification. Therefore, it can be treated as a system identification procedure.

First, we must focus on such global system parameters as the following: membership functions, linear coefficients, fuzzy engine type etc. Afterwards, we search for an optimal number of rules, a feature selection scheme and a proper partition of the feature space.

The aim of the pattern recognition task is to classify a given object described by its features $x = [x^{(1)}, ..., x^{(d)}]^T \in R^d$ to one of the predefined categories $j \in \boldsymbol{M} = \{1, ..., M\}$, which leads to the following definition of classifier Ψ

$$\psi : X \to \boldsymbol{M}. \tag{1}$$

To use the fuzzy decision-making (FDM) for combination of classifiers, the following fuzzy combiner model is proposed (Fig. 1).

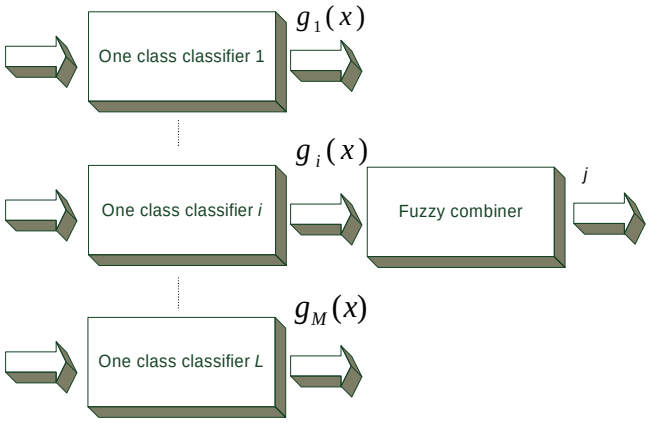


Fig. 1 *A general scheme of the proposed fuzzy ensemble where* $M \geq 3$.

Let us denote a dataset described as $DS = \{(x_1, j_1), ..., (x_n, j_n)\}$, where n defines the size of a dataset. Each k-th element of DS is described by a feature vector values x_k and its correct classification j_k.

In the presented model, the quantity of one-class classifiers will be also equal to M. Let $g_i(x)$ denote the support value for the statement that the current input data belongs to the i-th class. In the case of the proposed combiner, normalized values or $g_i(x)$ in the form of probability estimates are not required. Set of discriminant functions, $\boldsymbol{G}$, is defined by $G = \{G_1, ..., G_M\}$, where G_i denotes the i-th one-class classifier in the ensemble, $i = 1, 2, ..., M$. One seeks classifiers that minimize the number of misclassified samples. According to the proposed approach, the label of each data is determined from the aggregation of the experts in the form of one-class classifiers. In contrast to FI and DT, $g_i(x)$ couldn't be a fuzzy support value. It is an input value which still needs to be fuzzified.

In the proposed model, both one-class classifiers and one-against-all binary classifiers could be used. Binary classifiers are transformed to the one-class classifiers through the following equation:

$$g(x) = g_t(x) - g_o(x). \tag{2}$$

In such case, the classifier support for a given class is a difference between a support for the target $g_t(x)$ and the outlier class $g_o(x)$. The individual classifiers are assumed to be trained, and this issue will not be investigated further.

One-class classifiers provide us uncertain data. The proposed solution operates on the abstract level in order to extract more data, which is not possible when operating on classifiers' outputs in the form of the class labels or binary values. Therefore, values returned by classifiers are interpreted using fuzzy logic.

Decision problems could be divided into two groups:

- processing of partial information
- processing of uncertain information.

In the second case, fuzzy logic can be used successfully [36].

In a generic form we can describe our fuzzy combiner as a classifier with M discrimination functions. We will define their set as $Y = \{y_1, ..., y_M\}$.

It is a common practice that a Fuzzy Decision Making System (FDMS) is usually comprised of four main components: fuzzification block, reasoning block, knowledge base and finally defuzzification block (Fig. 2).

Explanation of the given blocks is following [27]:

- Fuzzification block is responsible for the assigning of the membership values to the fuzzy sets for a given input vector.
- Reasoning block combines fuzzy rules with membership values of corresponding fuzzy sets and provides reasoning similar to human.
- Defuzzification block is responsible for translating fuzzy output variables into actual output value.

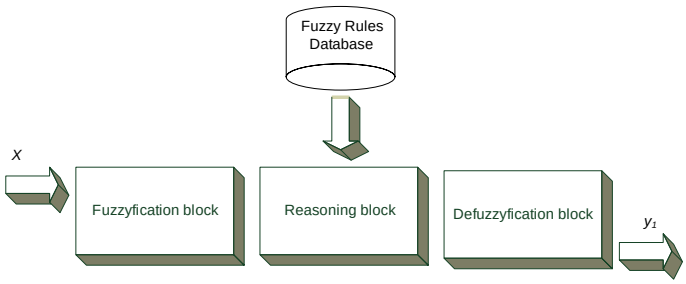


Fig. 2 *Fuzzy engine general scheme.*

According to definition [27], in some non-empty space fuzzy set A is denoted as

$$A(g_i(x)) = \{(g_i(x), \mu_A(g_i(x))); \mu_A(g_i(x))) \in < 0, 1 >\}, \tag{3}$$

where μ_A is a membership function of the fuzzy set A. Fuzzification in proposed scheme can be described as a mapping of $g_i(x)$ to a fuzzy set A.

The most common membership functions are trapezoid

$$f(x) := max(1; min(0; 0.5 + \sigma(x - \alpha)) \tag{4}$$

or sigmoid function

$$f(x) := \frac{1}{1 + e^{\sigma(x-\alpha)}}, \tag{5}$$

where σ and α are parameters which require optimization.

Männle [25] points out that the shape of the membership function has a minor influence on the FDMS performance. On the other hand, Kuncheva *et al.* [22] argue that classifiers usually output extreme values, that is, values with very high or very low level of support for the hypothesis. Therefore, despite the shape of the membership functions used, their parameters (center, width) must be adjusted to each type of one-class classifiers for a given multi-class problem separately.

Theoretically, a higher number of fuzzy sets should improve quality, but it would be at the cost of the performance of the proposed model. It could also decrease generalization properties of the fuzzy combiner and provide additional parameters of the model, for which larger training dataset would be required.

The assumption is made that the memory to sustain fuzzy rules is of sufficient size. We propose space partitioning in order to find membership function parameters and fuzzy rules candidates, therefore the number of the fuzzy rules will not be a parameter of the proposed system.

In reasoning block conclusions are made basing on decision rules (linguistic model), where the k-th rule looks as follows

$$\begin{aligned} &IF(g_1(x)\ IS\ A_1^k\ AND/OR\ ...\ AND/OR\ g_M(x)\ IS\ A_M^k) \\ &\qquad THEN\ (y_1\ IS\ D_1^k\ AND\ ...\ AND\ y_M\ IS\ D_M^k), \end{aligned} \tag{6}$$

where k denotes number of a fuzzy rule, D_1^k denotes fuzzy set for output variable y_1, used in k-th rule.

Above rule is an example of the Mandani type rule [27]. It has an implementation drawback in the form of the defuzzification block in fuzzy engine and also high calculation complexity. These problems do not exist in Yasukawa and Sugeno type rules.

In Yasukawa's fuzzy rules system response is calculated using constant $c^{(k)} \in R$:

$$\begin{aligned} &IF(g_1(x)\ IS\ A_1^k\ AND/OR\ ...\ AND/OR\ g_M(x)\ IS\ A_M^k) \\ &\qquad THEN\ (y_k = c^{(k)}). \end{aligned} \tag{7}$$

Sugeno's k-th rule on the right side of the equation has linear function in the form:

$$\begin{aligned} &IF(g_1(x)\ IS\ A_1^k\ AND/OR\ ...\ AND/OR\ g_M(x)\ IS\ A_M^k) \\ &\qquad THEN\ (y_k = c_0^{(k)} + c_1^{(k)} G_1 + ... + c_M^{(k)} G_M), \end{aligned} \tag{8}$$

where each $c_0^{(k)}, ..., c_M^{(k)} \in R$.

It was shown [25] that Sugeno's fuzzy rules give better results than Yasukawa's ones, therefore, proposed model uses Sugeno's rules. Parameters of these rules, as well as the rules themselves, fuzzy sets parameters, will be adjusted to each given problem separately. It can be achieved through expert knowledge or by an automatic technique, like one with support of ANNs [27].

Each output of fuzzy combiner will be equal to Sugeno's fuzzy engine output and can be denoted as:

$$y_i(G) = \frac{\sum_{k=1}^{S}(w_k(G) * f_k(G))}{\sum_{k=1}^{S} w_k(G)}, \tag{9}$$

where S is a number of rules, $w_k(G)$ is a fuzzy relation on the k-th rule and $f_k(G)$ is a linear function output.

Class supports are calculated using (9). Fuzzy combiner points to a class for which the support value is the highest.

3.2 Illustrative example

Below we try to give more insight into why the proposed fuzzy combiner is expected to work differently from other widely used combiners.

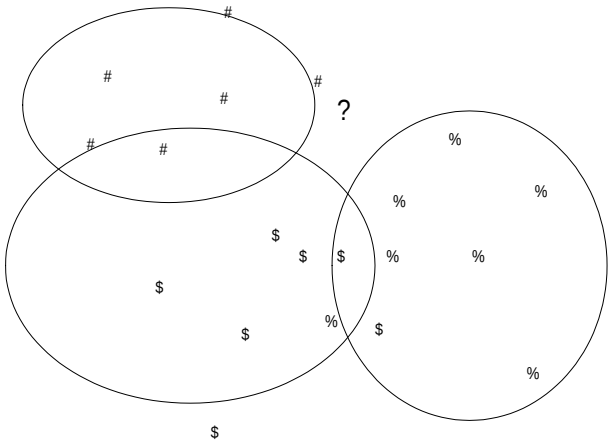


Fig. 3 *Attempt to classify object "?".*

One-class classifiers give us a potential of adjusting single class supports. On the combiner level, we can connect such classifiers using repeating errors or some sort of intuitive connections between class distributions. In our case, it is not even important if the support value will be in the form of a distance to some reference group of objects, probability estimate or other. In the proposed model, support values for classes are being interpreted by fuzzy engines through fuzzification process.

A whole proposed classification process can be compared to team skate racing. In such a race, the result of slowest team member is taken in consideration. In theory, each competitor tries to do his/her best. The team coach is responsible for the tactic, that is, for choosing when the team is to slow down or speed up, what should be the right tempo for the team. Similar operation can be expected from the fuzzy combiner. Like the coach, it knows how team effort can be adjusted to optimize the final result. Knowing where one class classifier is usually wrong, we can adjust the supports to be higher in some cases, lower in the others. Therefore, an attempt to classify an object (Fig. 3) can gain from fuzzy reasoning (Fig. 4).

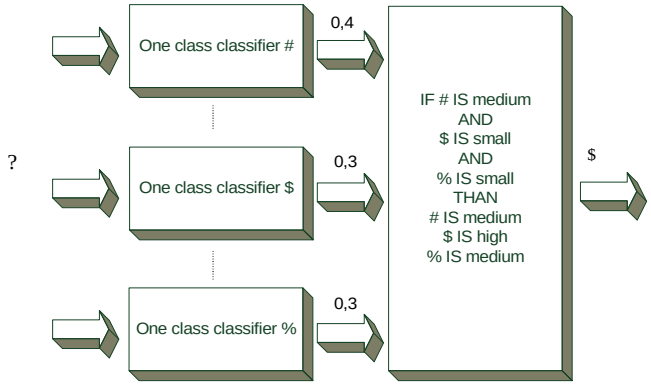


Fig. 4 *Example of pattern recognition using one fuzzy rule.*

We achieve a model which can be described as a set of generic templates. Such solution can change decision boundaries, and – perhaps – improve the overall classification quality. That makes the proposed fuzzy combiner similar to DT. Proposed fuzzy combiner is also a class-indifferent as it also treat the classifiers outputs as a context-free set of features. Kuncheva argues [23] that all class-conscious combiners are idempotent by design, that is, if an ensemble consists of L copies of classifier Ψ, the ensemble decision will be no different from the decision of Ψ. As in case of the DT, the proposed combiner will not be necessarily identical to D.

4. Implementation Model of the Proposed Fuzzy Combiner

ANFIS was chosen as the implementation model because of its strong background in both control and classification tasks. It is based on a fuzzy Sugeno model which is optimized via the ANNs training. The initial membership functions and rules for the fuzzy inference system can – but do not need to – be designed by employing human expertise about the target system to be modeled. Afterwards, ANFIS adjusts fuzzy rules and membership functions to improve description of the given system behavior [37]. The only drawback is that ANFIS is a fuzzy inference system with one output only. Therefore, proposed fuzzy combiner must be implemented as a set of 'one-against-all' ANFIS blocks, which is depicted in (Fig. 5).

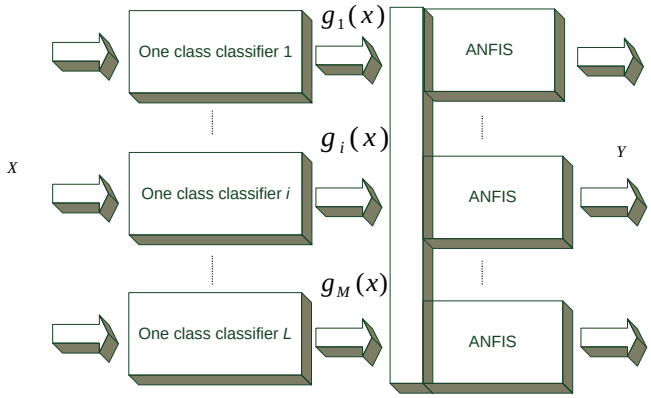


Fig. 5 *Implementation model.*

A fuzzy decision-making system is designed and implemented using Matlab, PRTools and DD tools software for combining the decisions of experts systems that have made our MCS. PRTools [13] is a Matlab based toolbox for pattern recognition. DD Tools [31] is a framework for one-class classification.

5. Learning algorithm

The general scheme of the learning algorithm is presented in Fig. 6. In order to improve the training efficiency and eliminate the possible trapping due to local minima, a hybrid learning algorithm is applied to tune the parameters of the membership functions. It is a combination of the gradient descent methodology and the least-squares estimate. During the forward pass, the node outputs advance until the output membership function layer where the consequent parameters are identified by the least squares estimate. The backward pass uses the back propagation

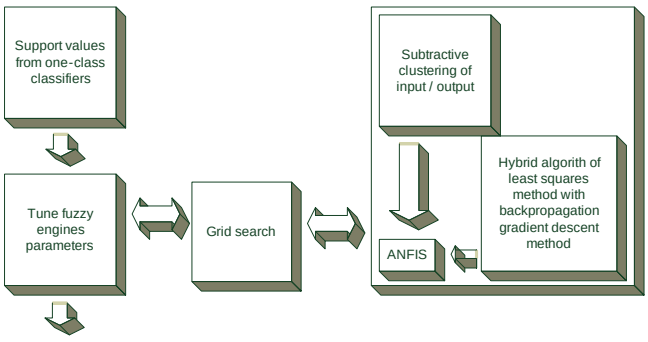


Fig. 6 *Fuzzy combiner training.*

gradient descent approach to update the premise parameters, based on the error signals that propagate backward. More detailed description of ANFIS can be found in [37]. Factorial design (grid search) [3] is chosen to find values for the learning algorithm parameters. The selected factors are revealed in Tab. I.

Factor	Description	Values
Radii	Subtractive clustering	0.2 **0.3** 0.4 0.5
The squash factor	Subtractive clustering	**1** 1.5 2
The accept ratio	Subtractive clustering	0.3 0.4 **0.5**
The reject ratio	Subtractive clustering	**0.1** 0.2 0.3
Epochs number	ANFIS epochs number	60 75 **90** 105
Initial step size	ANFIS initial step size for backpropagation gradient descent method	0.005 **0.01** 0.05 0.1

Tab. I *Factors used in grid search. The most often used values in experiments are bolded.*

In order to find values for selected training parameters a given dataset is divided into two sets: one for training and for combiner performance evaluation. The first one is twice bigger than the second one. Classification error is used as a grid search criterion. The parameters are listed underneath.

1. Radii (subtractive clustering) – represent a cluster radius in which cluster center will be searched.
2. The squash factor (subtractive clustering) – multiplied by radius, is used to discourage selection of other cluster centers near actual one.
3. The accept ratio (subtractive clustering) – it is a fraction of the potential of the first center, above which another point will be accepted as an another cluster center.
4. The reject ratio (subtractive clustering) – it is a fraction of the potential of the first center, below which another point will be rejected as another cluster center.
5. Training epoch number (ANFIS).
6. Initial step size.

6. Experimental investigation

6.1 Set-up of experiments

The experiments have been carried out using five benchmark databases described in Tab. II.

Classification errors were obtained using 5x2 Fold Cross-Validation. Commonly used T-test has a flow in the form of quite significant results variability and that

no.	dataset	classes	features	objects	percentage of the most represented class	origin
1	Cone Torus	3	2	800	50	generated
2	Iris	3	4	150	Probabilities of all classes are equal	UCI
3	Glass identification	6	9	214	35,51	UCI
4	Image segmentation	7	19	2310	Probabilities of all classes are equal	UCI
5	Gaussian distributed classes *)	8	2	600	14	generated

***)** Objects were randomly generated according to Gaussian distributions using PRTools toolbox.

Tab. II *Datasets used for testing.*

is why the Alpaydin's 5x2 Fold Cross-Validation Paired F Test [2] was used for hypothesis testing.

Tax *et al.* [29] proposed few approaches for binary classifiers combination. Two schemes of voting were compared, one including "one-against-one" classifiers (vote 1-1), as also "one-against-all" classifiers (vote 1-r). In two other methods (prob 1-1 and prob 1-r) outputs of binary classifiers were mapped to a posterior probability estimate. Afterwards, the object was assigned to the class with the largest output. In the method vote 1-1, as also in vote 1-r, random assignment of rejected objects reduced performance. Because of this fact, prob 1-r gave the best results from all the combination methods used by Tax [29], and we decided to use it as a reference combiner. One can easily notice that in case of one binary classifier per class with the output in the form of posterior probability estimate, this combination method is equal to 'one-against-all' ECOC. Though not possessing good error correcting capabilities, they are the most used schemes due to their simplicity. Yet, in tests [21] they appeared to be a challenging opponent to other combination methods of binary classifiers.

Classification error is denoted by:

$$f = \frac{f_p + f_n}{N}, \tag{10}$$

where N is objects quantity, f_p, f_n are false positive and false negative response.

Individual classifiers implementation available in PRTools and DD Tools were used. No attempt was made to additionally tune the individual classifiers. Linear

One-class classifiers per class	Types of one-class classifiers	Description
1	Linear perceptron	Standard PRTools training
1	Quadratic Bayes Normal Classifier	Standard PRTools training
1	Support Vector Machine (SVM)	Standard PRTools training, polynomial kernel
1	Simple Gaussian Target Distribution	Standard DD Tools training
1	The Auto-encoder Neural Network	Standard DD Tools training
1	The Support Vector Data Description (SVDD)	Standard DD Tools training, RBF kernel

Tab. III *Variations of proposed algorithm.*

Perceptron, Quadratic Bayes Normal classifier and Support Vector Machine were changed into one-class classifiers using (2). Simple Gaussian Target Distribution, the Auto-encoder Neural Network and the Support Vector Data Description were used as examples of standard one-class classifiers. In case of the SVDD, sigma parameter must have been altered for Image segmentation dataset (value changed from 5 to 50). The problem was that these classifiers for this dataset accepted all objects as a target class, returning quasi constant values.

6.2 Experimental results

Through tests we attempted to defend the following hypotheses:

- In the real problems fuzzy combination can improve response of the binary classifier ensembles.
- Proposed fuzzy combination method can be used successfully both with binary classifiers and one-class classifiers.
- Fuzzy combination of one-class classifiers can be a valuable multiclass classifier itself.

Tests results on described datasets are summarized in Tab. IV.

Results are presented in the following way: the number of the dataset (defined in Tab. II) followed by fuzzy combiner (FC) and prob 1-r (1-r) combination errors. The base classifier is presented at the top of each table. Both fuzzy combiner and prob 1-r combination method have used exactly the same instances of individual classifiers. Multiclass classifier (MC) is an individual classifier trained in PRtools for a multi-class task directly. F statistic paired test (FSPT1) was performed between fuzzy combiner and prob 1-r combination method. F statistic paired test 2 (FSPT2) was performed between proposed fuzzy combiner and multiclass classifier.

	Linear Perceptron					Auto-encoder neural network		
DS	FC	1-r	MC	FSPT1	FSPT2	FC	1-r	FSPT
1	3,4	8,1	8,8	0,88	1,57	8,8	13,2	2,21
2	13,8	38,1	38,1	4,88	8,34	17,4	22,6	3,70
3	39,6	47,5	54,3	1,85	2,93	44,6	61,3	1,94
4	3,6	14,8	13,7	40,32	5,66	8,74	29,4	16,22
5	13,3	67,5	66,0	37,50	77,80	16,3	24,4	8,41
	Quadratic classifier					Simple Gaussian Target distribution		
1	11,7	6,5	3,6	1,38	2,18	3,1	3,6	0,68
2	14,9	25,1	19,3	13,08	3,99	13,4	24,9	15,25
3	48,7	51,9	85,4	1,37	10,65	42,5	44,9	0,70
4	15,7	31,5	14,6	70,31	0,58	4,5	16,3	15,77
5	13,6	28,2	14,0	29,01	1,16	16,3	21,8	5,159
	The support vector machine					The support vector data description		
1	4,0	24,1	4,0	38,54	0,67	7,5	6,8	2,33
2	15,2	28,4	26,9	282,76	244,86	14,3	35,3	125,41
3	43,1	52,2	41,0	2,60	1,66	36,8	59,1	6,74
4	5,1	19,5	7,9	84,50	18,05	10,5	24,5	12,61
5	16,8	61,2	46,9	258,81	226,25	15,4	39,6	21,38

Tab. IV *Classification errors (in %) from 5x2 Fold Cross Validation Test.*

In the second part of the table, standard one-class classifier errors are presented in the same way. There is no default implementation of multi-class classifier using one-class classifiers. Finally, Tab. V summarizes experiments we carried out.

F Test – hypothesis rejected	F Test – hypothesis defended	FC error is smaller	FC error is bigger	FC gave the best result	Number of tests
19	11	28	2	26	30

Tab. V *Tests sum-up.*

We can state that according to 5x2 Fold Cross Validation Paired F-Test in most cases we can reject hypothesis that compared classifiers (proposed fuzzy combination model, prob 1-r) are equal. In no case, where fuzzy combination gave higher error, has the F-Test allowed us to make statement that the combination methods are considerably different (significance level $F_\alpha = 474$). Only in some cases the performance of the proposed fuzzy combiner is worse than prob 1-r or a standard multiclass classifier.

In most cases, the F-test sustained the hypothesis that on the Iris data set combination methods do not differ significantly. This seems to be caused by a relatively

good performance of all classification methods for the mentioned dataset, causing the field for correction to be relatively small. Similar results were achieved in relation to the Glass identification. It is somewhat surprising because difference in classification error of combination schemes exceeds 16% for the Auto-encoder Neural Network used as a base classifier. The performances on the Glass identification dataset are very poor, the best was the proposed fuzzy combiner with the support vector data description as an individual classifier. For the experiments regarding datasets – Cone Torus, Image segmentation as well as Gaussian distributed classes – in most of the examined cases the F-test rejected the hypothesis that combination schemes do not differ significantly. Therefore, the proposed fuzzy combiner appeared especially successful for the above listed datasets.

Proceeding from the F-Test results, we can state that we managed to achieve significant improvement for both the Support Vector Machine, and the Support Vector Data Description. For these classifiers, in four cases from among the five, the F-Test allowed to reject the hypothesis that the combination methods do not differ noticeably. The worst result was noticed for the Auto-encoder Neural Network for which the F-test was rejected only in two cases from among the five. On the other hand, results obtained from the proposed fuzzy combiner had smaller error than prob 1-r combination of this base expert.

Incorrectly classified objects bring us some additional information, showing that the one-class classifiers combination for the multi-class tasks reduces control of the allowed error type I. Furthermore, one cannot expect the one-class classifier to have a good performance as a two-class one because training samples from two classes provide more information to define the decision boundary. However, the use of one-class classifiers is fully justified by the results presented above, as the proposed fuzzy combiner yields consistently lower error rates for the major part of the datasets. In conclusion, the performed tests support our hypotheses stated in the beginning of the chapter.

7. Conclusions

The paper has presented a new method of one-class classifier combination based on the neuro-fuzzy approach which allows to restore a multiclass recognition problem. The proposed idea for implementing a fuzzy multi-class classifier was found to perform quite satisfactorily on some benchmark datasets. The obtained results indicate that the fuzzy combination can improve response of the binary classifier ensembles in the real problems. It is worthwhile to point out that satisfactory results can be achieved even in the standard case of one one-class classifier on class in a multi-class problem. The proposed fuzzy combiner provides a clear improvement of the overall results proving that one-class classifiers combination leads to a good and flexible approach. Among many advantages of the proposed scheme, one must mention a small number of limitations on structure of combined one-class classifiers. Another strong point of the presented solution is the intuitive way to provide nonlinearity by fuzzy combiner of linear classifier, as it was received in the case of the linear perceptron. According to the performed tests, the one-class classifiers based on class distributions can additionally provide modifications of their responses in order to get better approximation of true classes distribution,

or at least to update the intersection points between them.

Authors are confident of the need for further research. As a research frontier we can point out both the combination of different types of one-class classifiers in one ensemble, and the influence of individual classifiers' increase upon the classification results. Additionally, results of the proposed methods of one-class classifier combination should be compared to well-known alternative fusers used by MCSs, such as ECOC or Decision Templates, in the case of classification with reduced number of classes.

Acknowledgement

This work is supported in part by the Polish State Committee for Scientific Research under a grant for the period 2010-2013.

References

[1] Alexandre L. A., Campilho A. C., Kamel M.: Combining Independent and Unbiased Classifiers Using Weighted Average, Proc. of the 15th Internat. Conf. on Pattern Recognition, vol. **2**, 2000, pp. 495-498.

[2] Alpaydin M. E.: Combined 5x2cv F Test for Comparing Supervised Classification Learning Algorithms, Neural Computation, **11**, 8, 1999, pp. 1885-1892.

[3] Alpaydin M. E.: Introduction to Machine Learning. Second Edition, The MIT Press, Cambridge, MA, 2010.

[4] Aarabi A., Fazel-Rezai R., Aghakhani Y.: A fuzzy rule-based system for epileptic seizure detection in intracranial EEG Clinical Neurophysiology, Clinical Neurophisiology, vol. **120**, Issue 9, 2009, pp. 1648-1657.

[5] Biggio B., Fumera G., Roli F.: Bayesian Analysis of Linear Combiners, Lecture Notes in Computer Science, vol. **4472**, 2007, pp. 292-301.

[6] Dietterich T. G., Bakiri G.: Solving multiclass learning problems via error-correcting output codes, Journal of Artificial Intelligence Research, 2, 1995, pp. 263-286.

[7] Duan K., Keerthi S. S., Chu W.: Multi-Category Classification by Soft-Max Combination of Binary Classifiers. 4th International Workshop, **06**, 11-13, 2003, pp. 125-134.

[8] Duin R. P.: The Combining Classifier: to Train or Not to Train?, 16th International Conference on Pattern Recognition, IEEE Computer Society, ISBN 0-7695-1695-X, 2002, pp. 765-770.

[9] Fumera G., Roli F.: A Theoretical and Experimental Analysis of Linear Combiners for Multiple Classifier Systems, IEEE Trans. on PAMI, **27**, 6, 2005, pp. 942-956.

[10] Giacinto G., Perdisci R., Del Rio M., Roli F.: Intrusion detection in computer networks by a modular ensemble of one-class classifiers. Information Fusion, 9, 2009, pp. 69-82.

[11] Güler I., Übeyli E. D.: Adaptive neuro-fuzzy inference system for classification of EEG signals using wavelet coefficients. Journal of Neuroscience Methods. 2005, **2**, 148, pp. 113-121.

[12] Hashem S.: Optimal linear combinations of neural networks, Neural Networks, **10**, 4, 1997, pp. 599-614.

[13] van der Heijden F., Duin R. P. W., de Ridder D., Tax D. M. J.: Classification, parameter estimation and state estimation – an engineering approach using Matlab, John Wiley and Sons, 2004.

[14] Hong J., Min J., Cho U., Cho S.: Fingerprint classification using one-vs-all support vector machines dynamically ordered with naive Bayes classifiers. Pattern Recognition, 41, 2008, pp. 662-671.

[15] Inoue H., Narihisa H.: Optimizing a Multiple Classifier Systems, LNCS, vol. **2417**, 2002, pp. 285-294.

[16] Jain A. K., Duin R. P. W., Mao J.: Statistical Pattern Recognition: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence., **1**, 22, 2000, pp. 4-37.

[17] Kittler J., Alkoot F. M.: Sum versus Vote Fusion in Multiple Classifier Systems, IEEE Trans. on Pattern Analysis and Machine Intelligence, 20, 2003, pp. 226-239.

[18] Krzanowski W., Partrige D.: Software Diversity: Practical Statistics for its Measurement and Exploitation, Department of Computer Science, University of Exeter, 1996.

[19] Lee C., Lin C.: Multiple Compensatory Neural Fuzzy Networks Fusion Using Fuzzy Integral. Journal of Information Science and Engineering, **3**, 23, 2007, pp. 837-851.

[20] Marcialis G. L., Roli F.: Fusion of Face Recognition Algorithms for Video-Based Surveillance Systems. In: Foresti G. L., Regazzoni C., Varshney P. (Eds.), Multisensor Surveillance Systems: The Fusion Perspective, Kluwer Academic Publishers, 2003.

[21] Klautau A., Jevtic N., Orlitsky A.: Combined Binary Classifiers With Application to Speech Recognition, Journal of Machine Learning Research, 4, 2003, pp. 1-15.

[22] Kuncheva L. I., Bezdek J. C., Duin R. P. W.: Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recognition, 2, 2001, pp. 299-314.

[23] Kuncheva L. I.: Combining Pattern Classifiers: Methods and Algorithms. New Jersey: Wiley, 2004.

[24] Kurzyński M., Burduk R.: Two-stage binary classifier with fuzzy-valued loss function. Pattern Analysis & Applications, **9**, 10, 2006, pp. 353-358.

[25] Männle M.: Parameter Optimization for Takagi-Sugeno Fuzzy Models – Lessons Learnt. In: Proceedings of IEEE Systems Man and Cybernetics, Tucson (AZ), USA, October 2001, pp. 111-116.

[26] Passerini A., Pontil M., Frasconi P.: New Results on Error Correcting Output Codes of Kernel Machines. Neural Networks, 15, 2004, pp. 45-54.

[27] Rutkowski L.: Flexible Neuro-Fuzzy Systems. Structures, learning and performance evaluation, Kluwert, 2004.

[28] Tax D. M. J., Duin R. P. W.: Combining One-class Classifiers. In: Proceedings of Multiple Classifier Systems, 2001, pp. 299-308.

[29] Tax D. M. J., Duin R. P. W.: Using two-class classifiers for multiclass classification. Pattern Recognition Group, 2, 2002, pp. 124-127.

[30] Tax D. M. J., Duin R. P. W.: Characterizing one-class datasets. Proceedings of the Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa, 2005, pp. 21-26.

[31] Tax D. M. J.: DDtools, the Data Description Toolbox for Matlab, version 1.7.3, Dec. 2009.

[32] Tumer K., Ghosh J.: Analysis of Decision Boundaries in Linearly Combined Neural Classifiers, Pattern Recognition, 29, 1996, pp. 341-348.

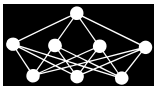
[33] Van Erp M., Vuurpijl L. G., Schomaker L. R. B.: An overview and comparison of voting methods for pattern recognition, Proc. of IWFHR.8, Canada, 2002, pp. 195-200.

[34] Walkowiak T., Wilk T.: Incident Detection and Analysis in Communication and Information Systems by Fuzzy Logic, Dependability of Computer Systems, IEEE Computer Society, 2007, pp. 205-212.

[35] Xu L., Krzyzak A., Suen Ch. Y.: Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition, IEEE Trans. on SMC, no. 3, 1992, pp. 418-435.

[36] Zadeh L. A.: Fuzzy Sets and Applications: Selected Papers by Lotfi A. Zadeh. Information and Control, 8, 1965, pp. 338-358.

[37] Zeng Q., Zhang L., Xu Y., Cheng L., Yan X., Zu J., Dai G.: Designing expert system for in situ Si3N4 toughened based on adaptive neural fuzzy inference system and genetic algorithms. Materials and Design, 30, 2009, pp. 256-259.



DASBE: DECISION-AIDED SEMI-BLIND EQUALIZATION FOR MIMO SYSTEMS WITH LINEAR PRECODING

José A. García-Naya, Adriana Dapena, Paula M. Castro, Daniel Iglesia*

Abstract: *Multiple-Input Multiple-Output* (MIMO) digital communications standards usually acquire *Channel State Information* (CSI) by means of supervised algorithms, which implies loss of performance since pilot symbols do not convey information. We propose obtaining this CSI by using semi-blind techniques, which combine both supervised and unsupervised (blind) methods. The key idea consists in introducing a decision criterion to determine when the channel suffered a significant change. In such a case, transmission of pilot symbols is required. The use of this criterion also allows us to determine the time instants in which CSI has to be sent to the transmitter from the receiver through a low-cost feedback channel.

Key words: *Channel estimation, blind source separation, learning algorithms, hybrid systems, linear precoding*

Received: 20th September 2010
Revised and accepted: 13th November 2010

1. Introduction

The main task when transmitting over channels with multiple antennas at the transmitter and/or at the receiver side is the separation or equalization of the transmitted data. *Linear Transmit Processing* (LTP), also termed *Linear Precoding* (LP), is a powerful method to separate signals in *Multiple-Input Multiple-Output* (MIMO) systems since it reduces computational costs and power consumption at the receiver end. Thus, the equalization task is performed at the transmitter, so the channel is pre-equalized or *precoded* before transmission with the goal of simplifying one side of the link and avoiding filter operations at the receiver. Such an operation prior to transmission is only possible for a centralized transmitter, e.g. the base-station in the downlink of a cellular system. Moreover, in case of a multiuser scenario with non-cooperative receivers, the users cannot cooperatively transform the received signals. Thus, transmit filters are necessary to separate

*Adriana Dapena – Corresponding author
University of A Coruña, Facultad de Informática, Campus de Elviña, 15071 A Coruña, Spain, E-mail: adriana.dapena@udc.es

signals for different users before transmission through a fading channel. Therefore, the advantages of carrying out this pre-equalization of channel effects at the transmitter are clear, compared to the traditional receiver and equalization alternatives. Although *Wiener Filtering* (WF) for precoding has been dealt with by only a few authors [15], unlike other criteria, Wiener linear precoding seems to be attractive transmit optimization that minimizes the *Mean Square Error* (MSE) between the transmitted and received symbols [8, 14, 17, 19].

The design of LP schemes has been widely studied for an ideal case in which *Channel State Information* (CSI) is perfectly known at the transmitter side [8, 14, 17, 19]. However, for transmit processing the availability of instantaneous CSI at the transmitter is the major difficulty. Thus, this work is focused on determining channel changes which will allow us to update the CSI available at the transmitter side by sending appropriate information through a reverse (also called *feedback*) channel. Most recent wireless communications standards include such a feedback channel for sending user link parameters. For example, *Worldwide Interoperability for Microwave Access* (WiMAX) standard uses this channel to send an index for selecting the most adequate code according to channel conditions [11]. However, to the knowledge of the authors, none of the current standards —even those under development— make use of such information to decide whether pilot symbols must be sent or not.

In this work, we propose a novel approach termed *Decision-Aided Semi-Blind Equalization* (DASBE), which allows us to reduce penalizations introduced by the use of pilot symbols and to get efficient utilization of the feedback channel. The main difficulty is to detect channel variations by using a simple decision criterion. The proposed criterion compares the channel matrix estimated using an unsupervised algorithm, to the previous estimate obtained by a supervised one. Pilot symbols are required only when channel variations are significant with respect to a previously selected threshold. A similar scheme has been proposed by the authors in [7, 10], where the transmitter could send two types of frames: frames containing only pilot symbols or frames containing only user symbols. This setup differs from the frame structure used in current standards, in which frames are composed by both pilot and user symbols. For this reason, the decision criterion proposed in DASBE is used to determine the instants where pilot symbols can be eliminated (or reduced) in standards frames.

This work is organized as follows. Section 2 shows our digital communications system. Section 3 reviews some supervised and unsupervised algorithms for channel estimation and source data recovery. Section 4 proposes the DASBE approach. Representative computer simulations are presented in Section 5 and Section 6 states some concluding remarks.

Vectors and matrices are denoted by lower case bold and capital bold letters, respectively. We use E[.], tr(.), $(.)^*$, $(.)^{\mathrm{T}}$, $(.)^{\mathrm{H}}$, det(.), ln(.) and $\|.\|_2$ for expectation, trace of a matrix, complex conjugation, transposition, conjugate transposition, determinant of a matrix, natural logarithm and Euclidean norm, respectively. The i-th element of a vector $\boldsymbol{x}$ is x_i. $h(.)$ is used to denote a scalar function and $h'(.)$ and $h''(.)$ denote its first and second derivatives.

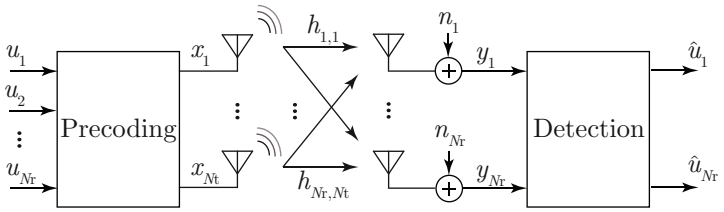


Fig. 1 *System with precoding over flat MIMO channels.*

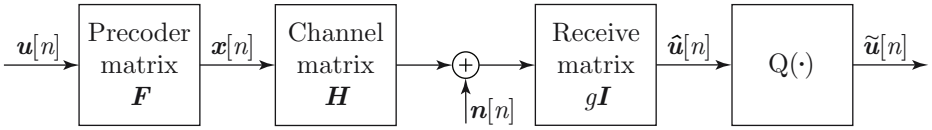


Fig. 2 *MIMO system with linear precoding.*

2. System Model

We consider a MIMO system with N_t transmit antennas and N_r receive antennas, as plotted in Fig. 1. The user symbols are expressed as $\boldsymbol{u}[n] = [u_1[n], \ldots, u_{N_\mathrm{r}}[n]]^\mathrm{T}$ and they are used by the encoder to generate the transmitted signal denoted by $\boldsymbol{x}(n) = [x_1(n), \ldots, x_{N_\mathrm{t}}(n)]^\mathrm{T}$. Suppose that $x_i(n)$ is transmitted from the i-th transmit antenna to the j-th receive antenna through the path $h_{ji}[q]$. Thus, the received signals (observations) presents the form

$$\boldsymbol{y}[n] = \boldsymbol{H}[q]\boldsymbol{x}[n] + \boldsymbol{n}[n], \tag{1}$$

where $n = 0, 1, 2, \ldots$ corresponds to sample index, q denotes time-slot, and $\boldsymbol{n}(n) = [n_1(n), \ldots, n_{N_\mathrm{r}}(n)]^\mathrm{T}$ contains *Additive White Gaussian Noise* (AWGN) with covariance matrix $\boldsymbol{C_n}$. We assume that the sources are transmitted in frames of N_B symbols and that the channel remains constant during several frames, i.e. the index q is unchanged during those frames. It can be demonstrated that this discrete-time model is equivalent to the continuous-time one only if *Inter–Symbol Interference* (ISI) between samples is avoided, i.e. if the Nyquist criterion is satisfied. In that case, we are able to reconstruct the original continuous signal from samples by means of interpolation. Hereafter, we assume this channel model, known as time-varying flat block fading channel.

As it was mentioned above, in order to simplify the requirements at the receiver, the equalization task can be performed at the transmitter so the channel is precoded before transmission, as plotted in Fig. 1. Such an operation —prior to transmission— is only possible when a centralized transmitter is used (e.g. the base-station for the downlink of a cellular system). The goal is to find the optimum transmit and receive filters, $\boldsymbol{F} \in \mathbb{C}^{N_\mathrm{t} \times N_\mathrm{r}}$ and $\boldsymbol{G} = g\boldsymbol{I} \in \mathbb{C}^{N_\mathrm{r} \times N_\mathrm{r}}$, respectively. Note that N_r is the number of scalar data streams. The resulting communications system is shown in Figure 2, in which the data symbols $\boldsymbol{u}[n]$ are passed through the

transmit filter $\boldsymbol{F}$ to form the transmit signal $\boldsymbol{x}[n] = \boldsymbol{F}\boldsymbol{u}[n] \in \mathbb{C}^{N_\mathrm{t}}$. The constraint for the transmit energy must be fulfilled, i.e.

$$E\left[||\boldsymbol{x}[n]||_2^2\right] = \mathrm{tr}\left(\boldsymbol{F}\boldsymbol{C}_{\boldsymbol{u}}\boldsymbol{F}^{\mathrm{H}}\right) \leq E_{\mathrm{tr}},$$

where $\boldsymbol{C}_{\boldsymbol{u}} = \mathrm{E}[\boldsymbol{u}[n]\boldsymbol{u}^{\mathrm{H}}[n]]$ is the correlation between the uncoded symbols and E_{tr} is the total transmitted energy. Thus, the received signal is given by

$$\boldsymbol{y}[n] = \boldsymbol{H}\boldsymbol{F}\boldsymbol{u}[n] + \boldsymbol{n}[n]. \tag{2}$$

After multiplying by the gain g, we get the estimated symbols

$$\hat{\boldsymbol{u}}[n] = g\boldsymbol{y}[n] = g\boldsymbol{H}\boldsymbol{F}\boldsymbol{u}[n] + g\boldsymbol{n}[n] \in \mathbb{C}^{N_\mathrm{r}}. \tag{3}$$

Therefore, the implementation of precoded systems implies very simple receivers since the observations are only multiplied by the gain factor g. Clearly, the restriction about common weights g for all the receivers is not necessary in case of decentralized receivers.

As mentioned before, we consider Wiener linear filtering, whose optimization consists in minimizing the MSE with a transmitted energy constraint, i.e.

$$\{\boldsymbol{F}_{\mathrm{WF}}, g_{\mathrm{WF}}\} = \arg\min_{\{\boldsymbol{F},g\}} \mathrm{E}\left[||\boldsymbol{u}[n] - \hat{\boldsymbol{u}}[n]||_2^2\right] \quad \text{s.t.: } \mathrm{tr}(\boldsymbol{F}\boldsymbol{C}_{\boldsymbol{u}}\boldsymbol{F}^{\mathrm{H}}) \leq E_{\mathrm{tr}}. \tag{4}$$

Note that such a constraint is necessary to avoid the dependence of the resulting transmitted energy on the channel realization. So, the transmitted energy constraint mentioned above might be the maximum value for poor channel realizations and thus the respective precoder solution is not valid. The transmitter may also not use the whole available transmitted energy, and therefore the final quality would not be as good as possible, since it could be improved by using more transmitted energy. In [8, 17], it is shown that the solution for the linear filters designed using that MSE criterion is expressed as

$$\begin{aligned} \boldsymbol{F}_{\mathrm{WF}} &= g_{\mathrm{WF}}^{-1}\left(\boldsymbol{H}^{\mathrm{H}}\boldsymbol{H} + \psi\boldsymbol{I}\right)^{-1}\boldsymbol{H}^{\mathrm{H}}, \\ g_{\mathrm{WF}} &= \sqrt{\frac{\mathrm{tr}\left(\left(\boldsymbol{H}^{\mathrm{H}}\boldsymbol{H} + \psi\boldsymbol{I}\right)^{-2}\boldsymbol{H}^{\mathrm{H}}\boldsymbol{C}_{\boldsymbol{u}}\boldsymbol{H}\right)}{E_{\mathrm{tr}}}}, \end{aligned} \tag{5}$$

where $\psi = \frac{\mathrm{tr}(\boldsymbol{C}_{\boldsymbol{n}})}{E_{\mathrm{tr}}}$.

3. Source Data Recovery Methods

Current digital communications standards define a frame as a sequence of pilot and user symbols. Supervised algorithms use the pilot symbols to estimate the channel (and to recover the transmitted signals), while unsupervised (blind) approaches discard this information [9]. In particular, in order to recover transmitted signals (sources), we will use a linear system whose weight matrix $\boldsymbol{W}[n] \in \mathbb{C}^{N_\mathrm{r} \times N_\mathrm{r}}$ (termed also *recovering matrix*) will be obtained using a supervised or unsupervised algorithm. The outputs of this system are computed using

$$\boldsymbol{z}[n] = \boldsymbol{W}^{\mathrm{H}}[n]\boldsymbol{y}[n]. \tag{6}$$

3.1 Supervised approach

We consider the utilization of a supervised approach to estimate the channel matrix $\boldsymbol{H}$ using the model in Eq. (1), in which $\boldsymbol{y}[n]$ and $\boldsymbol{x}[n]$ represent observations and sources, respectively, as a reference.

An important family of adaptive filtering algorithms arises from the minimization of the MSE between the outputs, $\boldsymbol{z}[n]$, and the sources, $\boldsymbol{x}[n]$ [13, 18]. Mathematically, the cost function is defined as

$$\begin{aligned} J_{\mathrm{MSE}} &= \sum_{i=1}^{N_{\mathrm{B}}} \mathrm{E}\left[|z_i[n] - x_i[n]|^2\right] \\ &= \mathrm{E}\left[\mathrm{tr}\left((\boldsymbol{W}^{\mathrm{H}}[n]\boldsymbol{y}[n] - \boldsymbol{d}[n])(\boldsymbol{W}^{\mathrm{H}}[n]\boldsymbol{y}[n] - \boldsymbol{x}[n])^{\mathrm{H}}\right)\right]. \end{aligned} \tag{7}$$

Then, the recovering matrix is updated using the following gradient algorithm

$$\boldsymbol{W}[n+1] = \boldsymbol{W}[n] - \mu\, \nabla_{\boldsymbol{W}} J_{\mathrm{MSE}}[n], \tag{8}$$

where $\nabla_{\boldsymbol{W}} J_{\mathrm{MSE}}$ is the gradient of J_{MSE} with respect to $\boldsymbol{W}$, i.e.

$$\nabla_{\boldsymbol{W}} J_{\mathrm{MSE}} = \mathrm{E}\left[\boldsymbol{y}[n](\boldsymbol{W}^{\mathrm{H}}[n]\boldsymbol{y}[n] - \boldsymbol{x}[n])^{\mathrm{H}}\right]. \tag{9}$$

The classical stability analysis for gradient–based algorithms consists in finding the point in which the gradient vanishes, and in defining the Hessian matrix whose coefficients are given by the second derivatives of J [4]. In particular, it can be demonstrated that the stationary points of the rule defined by Eq. (8) are

$$\nabla_{\boldsymbol{W}} J_{\mathrm{MSE}} = 0 \Rightarrow \boldsymbol{W} = \boldsymbol{C}_{\boldsymbol{y}}^{-1}\boldsymbol{C}_{\boldsymbol{yx}}, \tag{10}$$

where $\boldsymbol{C}_{\boldsymbol{y}} = \mathrm{E}[\boldsymbol{y}[n]\boldsymbol{y}^{\mathrm{H}}[n]]$ is the autocorrelation of the observations and $\boldsymbol{C}_{\boldsymbol{yx}} = \mathrm{E}[\boldsymbol{y}[n]\boldsymbol{x}^{\mathrm{H}}[n]]$ is the cross-correlation between the observations and the desired signals. In practice, these desired signals are considered as known only in a finite number of instants (pilot symbols) in which the estimation is used to recover the transmitted symbols. For this reason, the performance of this type of algorithms is degraded in the presence of calibration errors.

3.2 Unsupervised approach

The transmission of pilot symbols and the prior knowledge about channel matrices can be avoided by using *Blind Source Separation* (BSS) algorithms [5, 9, 16]. BSS methods simultaneously estimate the mixing matrix and the realizations of the source vector. In particular, we consider the model given by Eq. (2), where $\boldsymbol{y}[n]$ and $\boldsymbol{u}[n]$ represent observations and sources, respectively. The joint matrix $\boldsymbol{HF}$ is the matrix to be estimated.

One of the best known BSS algorithms has been approached by Bell and Sejnowski [3]. The idea proposed by these authors is to obtain the weighted coefficients of an artificial neural network, $\boldsymbol{W}[n]$, in order to maximize the mutual information (MI) between the outputs before the activation function $\boldsymbol{h}(\boldsymbol{z}[n]) =$

$\boldsymbol{h}(\boldsymbol{W}^{\mathrm{H}}[n]\boldsymbol{y}[n])$, where $h(.)$ is the activation function, and $\boldsymbol{y}[n]$ are the inputs. The resulting cost function is given by

$$J_{\mathrm{MI}}(\boldsymbol{W}[n]) = \ln(\det(\boldsymbol{W}^{\mathrm{H}}[n])) + \sum_{i=1}^{N_t} \mathrm{E}[\ln(h_i'(z_i[n]))], \tag{11}$$

where h_i is the i–th element of the vector $\boldsymbol{h}(\boldsymbol{z}[n])$, and $'$ denotes the first derivative. The maximum of this cost function can be obtained using a relative gradient algorithm [1, 2], which gives

$$\begin{aligned} \boldsymbol{W}[n+1] &= \boldsymbol{W}[n] + \mu \boldsymbol{W}[n]\boldsymbol{W}^{\mathrm{H}}[n] \left(\boldsymbol{z}[n]\ \boldsymbol{f}^{\mathrm{H}}(\boldsymbol{y}[n]) - \boldsymbol{W}^{-\mathrm{H}}[n]\right) \\ &= \boldsymbol{W}[n] + \mu \boldsymbol{W}[n] \left(\boldsymbol{z}[n]\boldsymbol{f}^{\mathrm{H}}(\boldsymbol{z}[n]) - \boldsymbol{I}\right) \end{aligned} \tag{12}$$

where $\boldsymbol{f}(\boldsymbol{z}) = [-h''(z_1)/h'(z_1), ..., -h''(z_{N_r})/h'(z_{N_r})]^T$. The expression in Eq. (12) admits an interesting interpretation by means of the use of the non-linear function $f(z) = z^*(|z|^2 - 1)$. In this case, Castedo and Macchi [6] have shown that the Bell and Sejnowski rule can be interpreted as an extension of the *Constant Modulus Algorithm* (CMA) proposed by Godard [12].

4. Decision-Aided Semi-Blind Equalization (DASBE)

Recent digital communications standards include a low-cost feedback channel which can be used to send estimates obtained using a supervised approach. Using this information, the transmitter adapts the precoding matrix $\boldsymbol{F}$ according to existing channel conditions. This approach has several limitations: Firstly, transmission of pilot symbols penalizes throughput, and secondly, as a consequence, overhead of the feedback channel appears in case of CSI and the transmission must be sent from the receiver each time a new frame is acquired. In addition, a large number of pilot symbols is needed to guarantee the convergence of the adaptive algorithm in Eq. (8) or to ensure that the matrix $\boldsymbol{C_y}$ in Eq. (10) is not singular.

In this section, we present a novel DASBE approach, which combines supervised and unsupervised techniques to mitigate the limitations found in classical approaches. By $\boldsymbol{W_u}[n]$ and $\boldsymbol{W_s}[n]$ we denote the respective matrices for the unsupervised and supervised modules.

We consider two frames types: firstly, classical frames formed by pilots and user symbols, and secondly, user frames containing only user symbols. The following procedure is performed at the receiver side each time a classical frame is received:

- First, the supervised algorithm estimates the channel matrix $\boldsymbol{H}$ from pilot symbols and, subsequently, it computes the gain parameter g_{WF} and the precoding matrix $\boldsymbol{F}$ according to Eq. (5).

- The joint matrix $\boldsymbol{HF}$ (denoted by $\hat{\boldsymbol{HF}}$) is computed and the unsupervised algorithm is initialized so that $\boldsymbol{W_u}[n] = \hat{\boldsymbol{HF}}^{-\mathrm{H}}$.

- The channel matrix $\boldsymbol{H}$ is sent to the transmitter through the feedback channel allowing the transmitter to update the precoding matrix $\boldsymbol{F}$ as given by Eq. (5).

On the contrary, when user frames are received, the unsupervised algorithm (see Eq. (12)) is adapted and the decision criterion is evaluated after processing all the frame symbols. An "alarm" is sent to the transmitter through the feedback channel when that decision criterion indicates that a significant channel variation has occurred. The user symbols included in both types of frames are recovered using $\hat{\boldsymbol{u}}[n] = g_{\mathrm{WF}}\boldsymbol{y}[n]$.

An important question is how to design the decision module in order to detect such channel variations. By combining Eqs. (2) and (6), the output $\boldsymbol{z}[n]$ can be rewritten as a linear combination of the sources

$$\boldsymbol{z}[n] = \boldsymbol{\Gamma}[n]\boldsymbol{u}[n], \tag{13}$$

where $\boldsymbol{\Gamma}[n] = \boldsymbol{W_u}^{\mathrm{H}}[n]\boldsymbol{H}\boldsymbol{F}$ represents the overall mixing/separating system. Sources are optimally recovered in case of selecting the matrix $\boldsymbol{W_u}[n]$ such that every output extracts a different single source. This occurs when the matrix $\boldsymbol{\Gamma}[n]$ has the form

$$\boldsymbol{\Gamma}[n] = \boldsymbol{D}\boldsymbol{P}, \tag{14}$$

where $\boldsymbol{D}$ is a diagonal invertible matrix and $\boldsymbol{P}$ is a permutation matrix. An interesting consequence of using a linear precoder is that the permutation ambiguity associated with unsupervised algorithms is avoided because of the initialization $\boldsymbol{W_u}[n] = (\boldsymbol{H}\hat{\boldsymbol{F}})^{-\mathrm{H}}$. This implies that the data sources are recovered in the same order as they were transmitted. Therefore, taking Eq. (14) into account, the optimum separation matrix produces a diagonal matrix $\boldsymbol{\Gamma}[n]$ and thus, the mismatch of $\boldsymbol{\Gamma}[n]$ with respect to a diagonal matrix allows us to measure channel variations.

Although channel matrices are unknown, we can use the estimation $\hat{\boldsymbol{H}}\boldsymbol{F}$ obtained by means of the supervised approach as a reference. Thus, we compute $\boldsymbol{\Gamma}[n] = \boldsymbol{W_u}^{\mathrm{H}}[n]\hat{\boldsymbol{H}}\boldsymbol{F}$ after processing the symbols in a frame. Consequently, that distance with respect to a diagonal matrix is measured using the following "error" criterion:

$$\mathrm{Error}[n] = \sum_{i=1}^{N_{\mathrm{t}}} \sum_{j=1, j\neq i}^{N_{\mathrm{t}}} \left(\frac{|\gamma_{ij}[n]|^2}{|\gamma_{ii}[n]|^2} + \frac{|\gamma_{ji}[n]|^2}{|\gamma_{ii}[n]|^2} \right), \tag{15}$$

where $\gamma_{ii}[n]$ denotes the i–th diagonal element of the matrix $\boldsymbol{\Gamma}[n]$. A possibility for determining when the channel changed significantly is to compare the above error value to a fixed threshold value (denoted by t), i.e. $\mathrm{Error}[n] > t$ would mean that a classical frame (i.e. a frame with pilot and user symbols) is required.

5. Simulation Results

In order to show the performance achieved with the proposed DASBE approach, we present results obtained by several computer simulations performed considering that 10 000 QPSK symbols have been transmitted through a MIMO system in blocks of 200 symbols each one (i.e. 50 frames). The system consists of four transmit

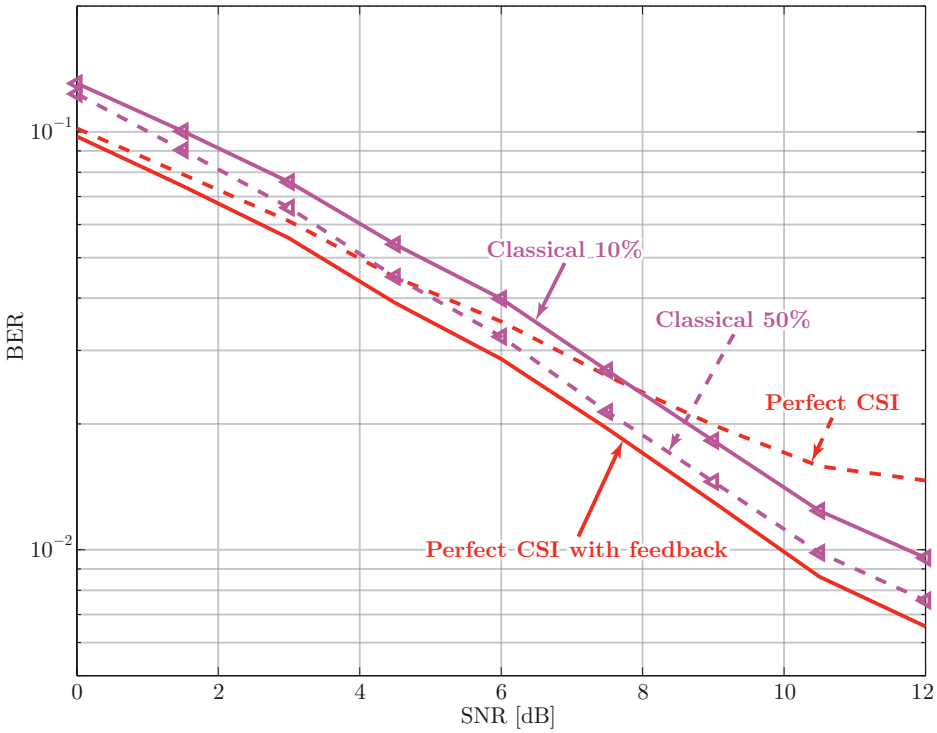


Fig. 3 *BER versus SNR obtained using a classical approach.*

and four receive antennas. The channel matrix changes each 10 frames according to the following model

$$\boldsymbol{H} = (1-\alpha)\boldsymbol{H} + \alpha \boldsymbol{H}_{\text{new}},$$

where $\boldsymbol{H}_{\text{new}}$ is a 4×4 complex matrix randomly generated according to a Gaussian distribution. The rest of parameters used for DASBE has been: threshold of $t = 0.1$ and initial step-size parameter of $\mu = 0.001$ for the unsupervised algorithm. The following results have been obtained by averaging 1 000 independent realizations, varying both channels and transmitted symbols.

Using the classical supervised approach, Fig. 3 shows the performance in terms of *Bit Error Rate* (BER) versus *Signal–to–Noise Ratio* (SNR) for a channel updating parameter $\alpha = 0.05$ and different percentages of pilot symbols per frame. Specifically, we select 10%, which means that 20 symbols per frame are dedicated to pilot symbols, and 50%, which corresponds to 100 pilots per frame. As a performance bounds, the following curves are also plotted in Fig. 3:

- BER curve when both perfect CSI and feedback channel are available between the receiver and the transmitter side (labeled as *Perfect CSI with feedback*).
- BER curve without feedback channel (labeled as *Perfect CSI*). In such a case, the precoding matrix is never updated, which leads to loss in performance with respect to the previous situation with the existing feedback channel.

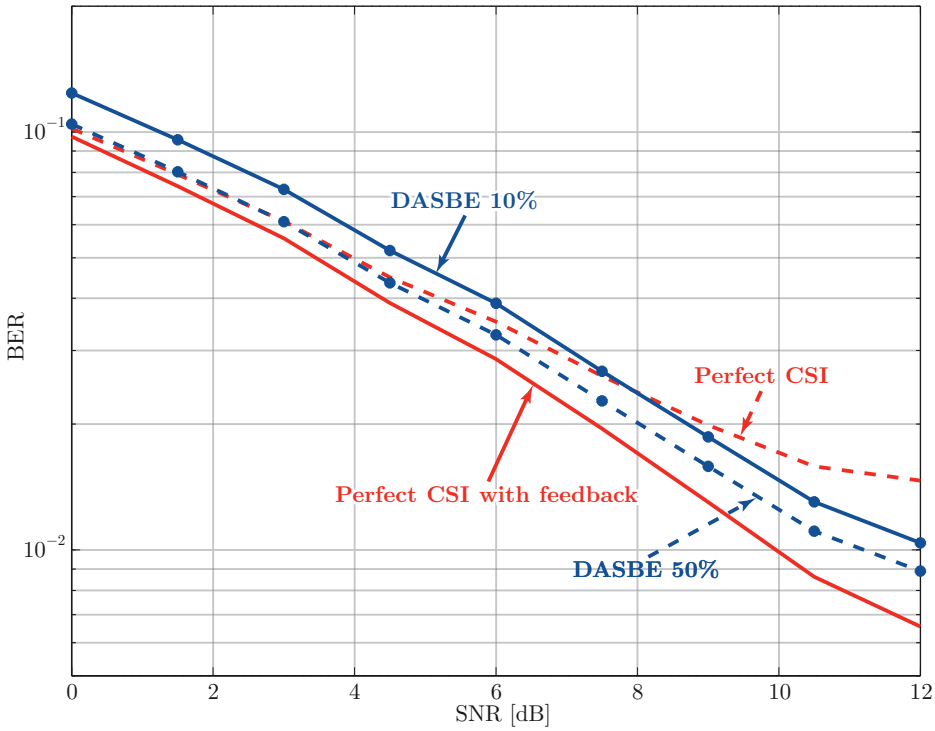


Fig. 4 *BER versus SNR obtained using DASBE.*

Notice that the utilization of the feedback channel produces a considerable improvement in terms of BER and SNR (in fact, for the SNR plotted in this figure, the system without feedback is not able to achieve a BER of 10^{-2}). It is also apparent that the classical approach needs 50% of pilot symbols to obtain a performance close to the Perfect CSI with feedback.

Fig. 4 plots the results obtained with the DASBE approach considering both 10% and 50% percentages of pilot symbols per frame. Notice that, using the DASBE approach, only the classical frames carry pilot symbols, while the user frames exclusively contain data symbols. From Fig. 4 it is apparent that the performance is similar to that offered by the classical approach (see Fig. 3), but with the advantages of reducing the feedback channel overhead (see Fig. 5) as well as the amount of needed pilot symbols (see Fig. 6).

Fig. 5 presents the utilization of the feedback channel depending on the approach used for tracking channel variations. Note that the classical approach transmits through the feedback channel each time a new frame is received, i.e. 50 times in total (independently of the pilot symbols percentage). However, the channel utilization for DASBE depends only on the decision criterion. It can be observed from Fig. 5 that the feedback channel utilization is considerably reduced in case of implementing DASBE.

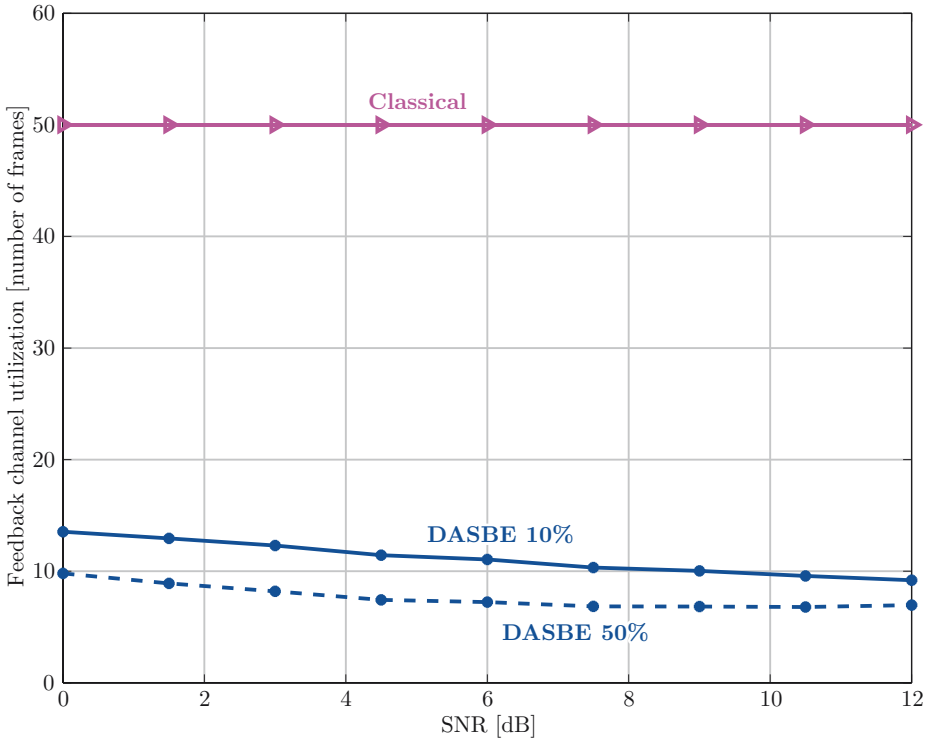


Fig. 5 *Utilization of the feedback channel versus SNR obtained using a classical approach and DASBE.*

Finally, Fig. 6 shows another important advantage of DASBE with respect to the classical approach, which consists in a considerable reduction in the number of needed pilot symbols. This is because pilots are included only when the degradation of the channel estimates is too large (according to the previously fixed threshold). Also note that for the DASBE approach, Fig. 6 plots the mean number of pilot symbols per frame considering the two frame types (i.e. classical and user frames) required to transmit 50 frames.

5.1 Remarkable comments

It is important to note that in case of the supervised estimation in Eq. (10), the matrix $\boldsymbol{C_y}$ may be singular. When this occurs, we decided to consider the previous channel estimate. Moreover, for those frames in which the unsupervised algorithm diverges, we reduced the step-size parameter to $\mu = \mu/10$ and initialized the algorithm to the matrix $\boldsymbol{W_u}[n]$ given by the previous frame.

Moreover, note that the BSS problem assumes that the observations are linear mixtures of the sources. From Eq. (5) it is easy to verify that for LP systems, assuming perfect CSI at the transmitter side, the joint matrix $\boldsymbol{HF}$ is diagonal

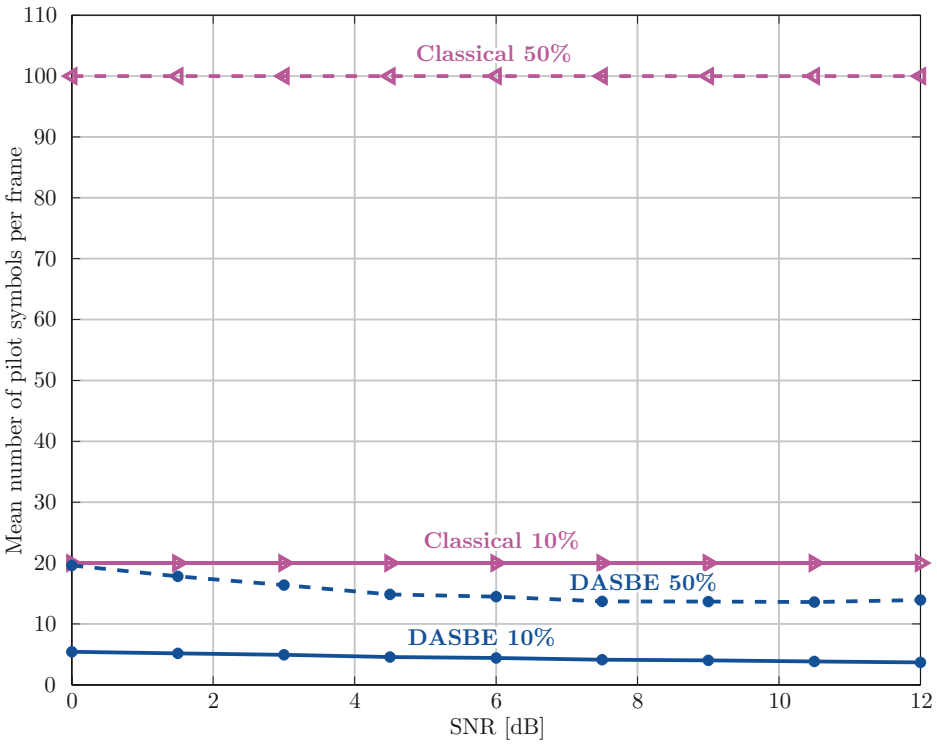


Fig. 6 *Pilot symbols versus SNR obtained using a classical approach and DASBE.*

when ψ is close to zero or, equivalently, when SNR is large. In that case, BSS methods are not justified. However, under realistic transmission scenarios, SNR is usually constrained to the interval $[5\,\mathrm{dB}, 15\,\mathrm{dB}]$ and perfect CSI is not available at the transmitter, which produces a non-diagonal matrix $\boldsymbol{HF}$ that allows us to use BSS algorithms.

6. Conclusions

Given a communications system in which a block flat fading channel is considered, we proposed an intuitive as well as simple method to detect channel variations. This decision criterion is used to develop a novel hybrid approach which combines both supervised and unsupervised algorithms. In case of significant channel variations, our system utilizes a supervised approach to estimate the channel coefficients, which are sent to the transmitter through a low-cost feedback channel. Otherwise, an unsupervised adaptive algorithm is used to track those channel variations. Simulation results have shown that the proposed approach is an attractive solution for wireless systems since it provides an adequate BER with a low overhead caused by transmitted pilot symbols and with reduced feedback channel occupancy.

Acknowledgments

This work has been funded by the Ministerio de Ciencia e Innovación of Spain, and the FEDER funds of the European Union under the grants 09TIC008105PR, TEC2007-68020-C04-01, and CSD2008-00010.

References

[1] Amari S.-I.: Gradient Learning in Structured Parameter Spaces: Adaptive Blind Separation of Signal Sources. In: Proc. WCNN'96, San Diego, 1996, pp, 951–956.

[2] Amari S.-I., Chen T.-P., Cichocki A.: Stability Analysis of Learning Algorithms for Blind Source Separation. Neural Networks, **10**, 8, August 1997, pp. 1345–1351.

[3] Bell A., Sejnowski T.: An Information-Maximization Approach to Blind Separation and Blind Deconvolution. Neural Computation, **7**, 6, November 1995, pp. 1129–1159.

[4] Benveniste A., Metivier M., Priourent P.: Adaptive Algorithms and Stochastic Approximations. Springer-Verlag, New York, 1990.

[5] Cardoso J.-F.: Blind Signal Separation: Statistical Principles. Proceedings of IEEE, **86**, 10, October 1998, pp. 2009–2025.

[6] Castedo L., Macchi O.: Maximizing the Information Transfer for Adaptive Unsupervised Source Separation. In: Proc. SPAWC'97, Paris, France, April 1997, pp. 65–68.

[7] Castro P. M., García-Naya J. A., Iglesia D., Dapena A.: A Novel Hybrid Approach to Improve Performance on Frequency Division Duplex Systems with Linear Precoding. In: Lecture Notes on Computer Science, vol. **II**, 6077, 2010, pp. 248–255.

[8] Choi R. L., Murch R. D.: New Transmit Schemes and Simplified Receiver for MIMO Wireless Communication Systems. IEEE Transactions on Wireless Communications, **2**, 6, November 2003, pp. 1217–1230.

[9] Comon P., Jutten C.: Handbook of Blind Source Separation, Independent Component Analysis and Applications. Academic Press, 2010.

[10] Dapena A., Castro P. M., Labrador J.: Combination of Supervised and Unsupervised Algorithms for Communication Systems with Linear Precoding. In: Proc. of WCCI'2010, July 2010, pp. 1265–1272.

[11] Ergen M.: Mobile Broadband: Including WiMAX and LTE. Springer Verlag, 2009.

[12] Godard D.: Self-recovering Equalization and Carrier Tracking in Two–Dimensional Data Communication Systems. IEEE Transactions on Communications, 28, November 1980, pp. 1867–1875.

[13] Haykin S.: Neural Networks: A Comprehensive Foundation. Macmillan College Publishing Company, New York, 1994.

[14] Joham M.: Optimization of Linear and Nonlinear Transmit Signal Processing. PhD dissertation. Munich University of Technology, 2004.

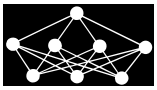
[15] Joham M., Kusume K., Gzara M. H., Utschick W., Nossek J. A.: Transmit Wiener Filter for the Downlink of TDD DS-CDMA Systems. In: Proc. ISSSTA, vol. **1**, September 2002, pp. 9–13.

[16] Jutten C., Herault J.: Blind Separation of Sources, Part I: An Adaptive Algorithm Based on Neuromimetic Architecture. Signal Processing, **24**, July 1991, pp. 1–10.

[17] Karimi H. R., Sandell M., Salz J.: Comparison between Transmitter and Receiver Array Processing to Achieve Interference Nulling and Diversity. In: Proc. PIMRC, vol. **3**, September 1999, pp. 997–1001.

[18] Macchi O.: Adaptive Processing. The Least Mean Squares Approach with Applications in Transmission. John Wiley and Sons, 1995.

[19] Nossek J. A., Joham M., Utschick W.: Transmit Processing in MIMO Wireless Systems. In: Proc. of the 6th IEEE Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication, Shanghai, China, May/June 2004, pp. I–18 – I–23.



DETECTION OF HEAT FLUX FAILURES IN BUILDING USING A SOFT COMPUTING DIAGNOSTIC SYSTEM

Javier Sedano[*], *Emilio Corchado*[†], *Leticia Curiel*[‡], *José Ramón Villar, Enrique de la Cal*[§]

Abstract: The detection of insulation failures in buildings could potentially conserve energy supplies and improve future designs. Improvements to thermal insulation in buildings include the development of models to assess fabric gain -heat flux through exterior walls in the building- and heating processes. Thermal insulation standards are now contractual obligations in new buildings, and the energy efficiency of buildings constructed prior to these regulations has yet to be determined. The main assumption is that it will be based on heat flux and conductivity measurement. Diagnostic systems to detect thermal insulation failures should recognize anomalous situations in a building that relate to insulation, heating and ventilation. This highly relevant issue in the construction sector today is approached through a novel intelligent procedure that can be programmed according to local building and heating system regulations and the specific features of a given climate zone. It is based on the following phases. Firstly, the dynamic thermal performance of different variables is specifically modeled. Secondly, an exploratory projection pursuit method called Cooperative Maximum-Likelihood Hebbian Learning extracts the relevant features. Finally, a supervised neural model and identification techniques constitute the model for the diagnosis of thermal insulation failures in building due to the heat flux through exterior walls, using relevant features of the data set. The reliability of the proposed method is validated with real datasets from several Spanish cities in winter time.

Key words: *Computational intelligence, soft computing, identification systems, artificial neural networks, non-linear systems, energetic efficiency*

[*]Javier Sedano
Departament of Artificial Intelligence and Applied Electronics, Technological Institute of Castilla and Leon, Poligono Industrial de Villalonquejar, C/Lopez Bravo 70, 09001, Burgos, Spain, E-mail: javier.sedano@itcl.es

[†]Emilio Corchado
Departamento de Informática y Automática, Universidad de Salamanca, Plaza de la Merced s/n, 37008, Salamanca, Spain, E-mail: escorchado@usal.es

[‡]Leticia Curiel
Department of Civil Engineering, University of Burgos, EPS Politecnica, Campus Vena, Edificio C. C/ Francisco de Vitoria, s/n, 09001, Burgos, Spain, E-mail: lcuriel@ubu.es, Tel.: +34 947259358.

[§]José Ramón Villar, Enrique de la Cal
Department of Computer Science, University of Oviedo, Campus de Viesques s/n, 33204 Gijón, Spain villarjose@uniovi.es, delacal@uniovi.es, Tel.: +34 985182597

Neural Network World 7/10, 883-898

Received: 20th September 2010
Revised and accepted: 13th November 2010

1. Introduction

The diagnostic system for identification of thermal insulation failures (TIF) could significantly increase building energy efficiency and substantially contribute to reductions in energy consumption and in the carbon footprints of domestic heating systems. Conventional methods can be greatly improved through the application of learning techniques to detect TIF when a building is in operation through a heat flux model - heat flux through exterior walls in a building-.

Assessing thermal insulation in new buildings is a well-known problem that has not as yet been fully resolved [21, 50]. Several different techniques are proposed in the literature. In [23], thermal insulation leaks are found by measuring thermal resistance and infrared (IR) thermography, while in [2], [37] only IR thermography is used to locate thermal insulation failures. The main drawback of using IR thermography is the high cost of equipment, alternatives using different technologies are always of interest.

Nevertheless, predicting the thermal dynamics of a building in operation is a complex task. The dynamic thermal performance of a building has mainly been used to estimate its power requirements. As an example, the difficulties of obtaining a black-box model for a generic building are documented in [47]. Furthermore, [11] cites examples of the errors associated with different kinds of techniques while providing possible solutions. Local building regulations need to be analyzed in the determination of TIF in order to profile the premises and the legal specifications for their physical parameters.

This interdisciplinary research represents a step forward in the development of techniques to improve dynamic thermal efficiency in existing buildings through a diagnostic system -modeling of heat flux- in the building. Although this may at first appear simple, noise due to occupancy and lighting profiles can introduce distortions and complicate detection. A novel three-step soft computing procedure for testing and validating the model -used in the diagnostic system- is proposed: firstly, the dynamic thermal behavior of a specific configuration is calculated using HTB2 software [29]. The outcome of the HTB2 should then be post-processed to obtain a suitable dataset. Subsequently, the dataset is analyzed using an exploratory projection pursuit (EPP) method [9], [16] called Cooperative Maximum-Likelihood Hebbian Learning (CMLHL) [6, 7], to extract the dataset structure and key relationships between the variables. Finally, a dynamic ANN model is trained and validated with them, which is used for fault diagnosis. This diagnosis dynamic model is responsible for estimating the heat flux through the exterior walls in the building, and the results are then compared with the real heat flux. Differences between the estimated and the real measures -above a reference value- are detected, which indicate the TIF.

Soft Computing represents a set of several technologies that aim to solve inexact problems [51]. It investigates, simulates and analyzes very complex issues and phenomena in order to solve real-world problems [40]. Soft Computing has been

successfully applied in feature selection, and plenty of algorithms are reported in the literature [4], [5], Principal Component Analysis (PCA) among others [30]. In this study, an extension of a neural PCA version [17] and other extensions are used to select the most relevant input features in a dataset as well as to study its internal structure.

This paper is organized as follows. Following this introduction, Section 2 describes the problem. Section 3 introduces the unsupervised connectionist techniques for analyzing the datasets in order to extract their relevant internal structures. Section 4 deals with classical identification techniques used in the diagnostic system -modeling system-. Section 5 describes a real case study in detail and the multi-step procedure. Section 6 describes the experiments and results obtained and finally, the conclusions are set out and comments are made on future lines of work.

2. Spanish Regulations and the Problem Description

Several national regulations on buildings and their construction were approved in Spain, 2007. The minimum pre-requisites for energy efficiency with which buildings must comply are given in the European Directive 2002/91/CE [13]. Project specifications, construction conditions and the basic requirements in Spain are specified in the CTE (*Código Técnico de Edificación* [Building Regulations]) [35]. One of the basic requirements is document HE1 that specifies the energy consumption limitation in buildings [35] and its revised updates.

Local regulations will be analyzed to extract the minimum requirements and parameters for heating systems and thermal comfort, and the certification procedure for energy efficiency. In Spain, energy efficiency is calculated as the ratio of combustible consumption needed to satisfy the energy demand of the building. Energy efficiency in the case of buildings constructed before the CTE approval is still an open issue, and the assumption is that it will be based on heat flux and conductivity measurement.

In these conditions, it could be interesting to model the heat flux in order to detect the isolation failures in buildings in operation. It is interesting that such model could distinguish the climate zone to analyze, the specific building geometry and orientation, etc. A novel procedure is proposed for this modeling task. This procedure includes several steps: the thermal dynamics simulation, the feature selection, the heat flux identification using neural networks models, and the detection of failures.

3. Analysis of the Internal Structure of the Dataset

In general, to obtain an efficient diagnostic system it is necessary to model it with a good dataset. Often, the systems are modeled using all the variables collected. This is not a proper way as some of them influence in the dynamic of the system and its inclusion only adds complexity to the model process degrading the effectiveness of the final model. Therefore, in this research, we propose a previous analysis

of the dataset using statistical methods and neural models, as Principal Component Analysis (PCA) [27, 34] and the Cooperative Maximum Likelihood Hebbian Learning model (CMHL) [6, 8], respectively, in order to know if the dataset is informative enough and to extract the most relevant variables in order to model it using only the main variables.

3.1 Component analysis

Principal Component Analysis (PCA) originated in the work by Pearson [34], and independently by Hotelling [27], describing multivariate dataset variations in terms of uncorrelated variables, each of which is a linear combination of the original variables. Its main goal is to derive new variables, in decreasing order of importance, which are linear combinations of the original variables and are uncorrelated with each other.

3.2 A neural implementation of exploratory projection pursuit

The standard statistical method of EPP [9, 16] provides a linear projection of a dataset, but it projects the data onto a set of basic vectors which best reveal the interesting structure in data; interestingness is usually defined in terms of how far the distribution is from the Gaussian distribution [42].

One neural implementation of EPP is Maximum Likelihood Hebbian Learning (MLHL) [9, 18]. It identifies interestingness by maximizing the probability of the residuals under specific probability density functions that are non-Gaussian.

An extended version of this model is the Cooperative Maximum Likelihood Hebbian Learning (CMLHL) [6] model. CMLHL is based on MLHL [9, 18] adding lateral connections [6, 8], which have been derived from the Rectified Gaussian Distribution [42]. The resultant net can find the independent factors of a dataset but does so in a way that captures some type of global ordering in the dataset.

Considering an N-dimensional input vector (x) and an M-dimensional output vector (y), with W_{ij} being the weight (linking input i to output j), then CMLHL can be expressed [8, 18] as:

1. Feed-forward step:
$$y_i = \sum_{j=1}^{N} W_{ij} x_j, \forall i \tag{1}$$
2. Lateral activation passing:
$$y_i(t+1) = [y_i(t) + \tau(b - Ay)]^+ \tag{2}$$
3. Feedback step:
$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i, \forall j \tag{3}$$
4. Weight change:
$$\Delta W_{ij} = \eta . y_i . sign(e_j) |e_j|^{p-1} \tag{4}$$

Where: η is the learning rate, $[\ ]^+$ is necessary to ensure that the y-values remain within the positive quadrant, τ is the "strength" of the lateral connections, b the bias parameter, p a parameter related to the energy function [9, 8, 18] and A the symmetric matrix used to modify the response to the data [6]. The effect of this matrix is based on the relation between the distances separating the output neurons.

4. Diagnostic System Using Identification Algorithms

Among the different methods for the detection and diagnosis of faults are: checking limits or thresholds, physical redundancy, deterministic methods -mathematical models-, methods based on knowledge and so on. Some examples are found in the literature [1, 28, 44, 45, 48].

In this context, System identification [31] is concerned with obtaining a model that best suits a given process behavior [31]. Firstly, several measurements are sampled from the process. The data gathered are then analyzed to obtain a model that estimates the desired process behavior. The model is then used to optimize the process output. Finally, the process is modified in order to enhance its outcome. If more adjustments are needed the cycle is repeated.

The system identification procedure includes the experiment design, the data visualization and analysis, the model learning and testing, and the model validation [31, 32, 33, 38, 39, 46, 49].

The experimental design determines the signals to be measured, the sensors to be used and their placement, the sample rate, and the generation of the datasets. Expertise is required as the experimental design decisions are problem dependent. Moreover, it is not always feasible in real world applications to gather data from the most relevant variables and, in most cases, the data are limited by the locations of the sensors that are installed. In other cases, portable instrumentation can be employed to measure some extra process variables. Nevertheless, the human-expert who designs the experiment always has an a priori theory and knowledge about the relationships between the variables.

When the dataset is gathered, several tasks should be carried out: eliminating missing data and outliers [3, 12, 14, 19, 20] scaling and normalizing the data [43], etc. Whenever data gathering is expensive and little data are available, it is usual to partition the data generating several train and test datasets. Standardized partitioning schemas are the k-fold cross validation and the 5x2 cross validation. This is all included in the data pre-processing and analysis step.

The selection of the model structure, their training and validation represents the core of the system identification. The classic theory includes a vast amount of model structures and training methods [31, 41]. Well-known functions are also used to rank the viability of the models. These functions are used as the criteria in the optimization problem of training the model. In the next subsection, several criteria functions are introduced and the use of Artificial Neural Networks in system identification is outlined.

4.1 The system identification criteria

According to [31], several measures have been proposed in the literature to evaluate the viability of a model:

- The representation percentage of the estimated model in relation to the true system, that is, the numeric value of the normalized mean error. There are several typical estimation models used in the literature, such as the one-step ahead prediction error (FIT1), the ten-step ahead prediction error (FIT10), and the simulation error (FIT). Equations (5) to (10) are used to calculate the FIT1 and FIT indexes. The FIT10 index can be derived in a similar manner as FIT1. In these equations, $u(t)$ is the input, $y(t)$ is the output, $\hat{y}_1(t|m)$ is the one-step ahead prediction, $\hat{y}_\infty(t|m)$ is the simulated output of the model, $\hat{G}(q)$ is the estimated transfer function from $u(t)$ to $y(t)$, $\hat{H}(q)$ is the estimated transfer function from $e(t)$ to $y(t)$ and q is the forward shift operator. The term $e(t)$ represents the white noise signal and it is included in the modeling errors. The term $e(t)$ is associated with a series of random variables of mean null value and variance λ.

$$\hat{y}_1(t|m) = \hat{H}^{-1}(q)\hat{G}(q)u(t) + (1 - \hat{H}^{-1}(q))y(t) \tag{5}$$

$$J_1(m) = \frac{1}{N}\sum_{t=1}^{N} |y(t) - \hat{y}_1(t|m)|^2 \tag{6}$$

$$FIT1(\%) = (1 - \frac{\sqrt{J_1(m)}}{\sqrt{\frac{1}{N}\sum_{t=1}^{N} |y(t)|^2}})100 \tag{7}$$

$$\hat{y}_\infty(t|m) = \hat{G}(q)u(t) \tag{8}$$

$$J_\infty(m) = \frac{1}{N}\sum_{t=1}^{N} |y(t) - \hat{y}(t|m)|^2 \tag{9}$$

$$FIT(\%) = (1 - \frac{\sqrt{J_\infty(m)}}{\sqrt{\frac{1}{N}\sum_{t=1}^{N} |y(t)|^2}})100 \tag{10}$$

- The loss or error function (V): the numeric value of the mean square error (MSE) that is calculated from the estimation dataset by means of Eq. (6).

- The generalization error value: the numeric value of the normalized sum of squared errors (NSSE) that is computed with the validation dataset by means of Eq. (6).

- The average generalization error value: the numeric value of the final prediction error (FPE), which is a criterion that is calculated from the estimation dataset. Eq. (11) is used to calculated the FPE value, where d_M is the dimension of θ -the estimated parametrical vector- and N is the number of samples of the estimation dataset.

$$FPE = \overline{J}_p(m) \approx J_1(m) + \frac{J_1(m)}{1 - (\frac{d_M}{N})}\frac{2d_M}{N} \tag{11}$$

- The graphical representations of true system output and both the one-step ahead prediction $\hat{y}_1(t|m)$, the ten-step ahead prediction $\hat{y}_{10}(t|m)$, and the model simulation $\hat{y}_\infty(t|m)$.

4.2 The ANN in the identification process

The use of ANN in the process of identification requires the selection of several parameters: the number of layers, the number of neurons per layer, and the activation functions. The methods by which the parameters are set up are fully documented in the literature. It was found that ANNs with two layers using non-linear functions in the hidden layer are universal approximators or predictors [10, 26].

The number of neurons per layer is also a relevant design parameter, and it should be analyzed in order to avoid over fitting [22, 24]. Each algorithm introduces some restrictions in the weight matrix. The most widely used training algorithms in system identification are the Lenvenberg-Marquardt method [15], the recursive Gauss-Newton method [31] and the batch and recursive versions of the back-propagation algorithm [25].

When using ANN, the purpose of an identification process is to determine the weight matrix based on the observations Z^t, so as to obtain the relationships between the nodes in the network. The weight matrix is usually referred as w, W or θ.

The supervised learning algorithm is then applied to find the estimator θ, so as to obtain the identification criterion [40]. Several well-known model structures are used when merging system identification with ANN. If the AutoRegressive with eXternal input model (ARX) is used as the regression vector, the model structure is called a Neural Network for ARX model (NNARX). Likewise, the Neural Network for Finite Impulse Response model (NNFIR), the Neural Network for Autoregressive Moving Average with eXternal input model (NNARMAX), and the Neural Network for Output Error model (NNOE), are also extensively used [40]. In the same way, it is possible to use an estimator for the one-step ahead prediction of the output $\hat{y}_1(t|m)$, where the polynomial degree values -n_a, n_b, n_c, n_d, n_f and n_k- are given as parameters.

5. A Multi-Step Method for Modeling Heat Flux in Buildings

The novel three-step Soft computing method is proposed to diagnose insulation failures, for the detection of heat flux through exterior walls in the building incorporates a diagnostic system that integrates different methodologies to obtain a parametric model which performs the diagnosis.

Firstly, the building is parameterized and its dynamic thermal performance in normal operation is obtained by means of simulation. Then, the data gathered are processed using CMLHL as a dimensionality reduction technique to choose the most relevant features in order to determine the heat flux. The second step outcome is a dataset, which is finally used to train and validate the heat flux nonparametric model that was used in the diagnostic system.

Fig. 1 shows the diagnostic system in a global manner. It indicates how training data are acquired from a theoretical model -HTB2-, which incorporates all the dynamic characteristics of thermal system. After data are preprocessed, using feature selection techniques and attributes. A dynamic ANN model is trained and validated with them, which is used for fault diagnosis. The actual data -real dataset- of the building's thermal system will be evaluated in the model that is generated by assessing two indexes: the representation percentage of the estimated output in relation to the true output (FIT1), Eq. (7) and the numeric value of the error, which is the difference between the responses of the heat flux -$y_1(t)$- in the building and the estimated heat flux -$\hat{y}_1(t|m)$- in the diagnostic system.

When the error exceeds a certain value -threshold value- or the FIT1 is less than a reference value, then the diagnostic system determines a failure.

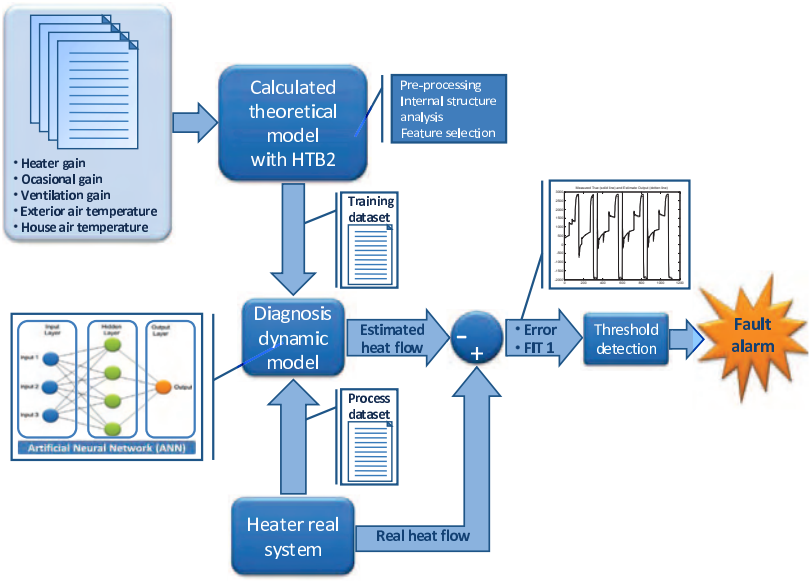


Fig. 1 *The diagnostic system: the data are obtained from a theoretical model -through HTB2-. They are then processed and a better dataset is found. The dataset is used to train the dynamic ANN model. Actual data -from a thermal system in operation- will be evaluated on the model, identifying errors that will determine the failure.*

5.1 Thermal dynamics data gathering by means of simulation

The following variables and datasets should be gathered in order to simulate the thermal behavior of a building: building topology; climate zone according to the specific regulations; building materials that comply with local regulations for the chosen climate zone; meteorological data for the climate zone and the simulated

time period: such as solar radiation, outdoor temperature, wind speed, etc., and realistic profiles for heating, lighting, small power devices, occupancy and ventilation.

In this study, the system is applied in Spain where the regulations establish five winter/summer zones, from E1 (a more severe climate zone) to A3 (a gentler climate zone).

Having defined and/or gathered these datasets, then the chosen simulation tool is applied to obtain the output data. In our case, the simulation software used is HTB2 [29]. The typical values that each variable could take for an E winter climate zone of maximum severity in Spain -i.e. the cities of Leon, Burgos or Soria among others- are shown in Tab. I.

Variable (Units)	**Range of values**	**Transmittance level (W/m^2K)**
Fabric gain -heat flux- (w), $y_1(t)$.	0 to -7,100	-External cavity wall: 0.54 -Double glazing: 2.90
Heater gain (w), $u_1(t)$.	0 to 4,500	-Floor/ceiling: 1.96
Occasional gain small power, occupancy and lighting gain – (w), $u_2(t)$.	0 to 5,500	-Party wall between buildings: 0.96 -Another party's wall: 1.05 -Internal partition: 2.57
Ventilation gain (w), $u_3(t)$.	0 to -5,500	
Exterior air temperature in February (°C), $u_4(t)$.	1 to 7	
Air temperature of the house (°C), $u_5(t)$.	14 to 24	

Tab. I *Typical values of each variable in an E winter climate zone city in Spain.*

5.2 Selection of the relevant features

As detailed in Section 2, PCA (Fig. 2.a) and CMLHL (Fig. 2.b), which were both applied to this real-life problem, are instrumental in identifying the internal structure of the data. In this procedure, the dataset gathered in the previous step is analyzed. The objective is to find the relationships between the input variables with respect to the heat flux. CMLHL (Fig. 2.b) allows to detect the relations of dependence and to choose the most relevant features. The outcome of this step is a new dataset with the features for which a relationship with the heat flux is found.

5.3 System identification applied to model normal building operation

Once the relevant variables and their transformations have been extracted from the thermal dynamics data, then a model to fit the normal building operation should be obtained in order to identify bias in the heat flux through exterior walls in the building. The heating process exhibits nonlinear behavior between output and

inputs, due to which the linear modeling techniques do not behave properly except in the linear behavior zones of the process. Consequently, the heating process has been modeled using soft computing techniques, specifically an ANN.

The different learning methods used in this study were implemented in Matlab© [36]. The experiment followed the identification procedure detailed in Section 4: the model structures were analyzed in order to obtain the models that best suited the dataset. The Akaike Information Criterion (AIC) is used to obtain the best degree of the model and its delay for each model structure. A total of thirty four different combinations of model structures and optimization techniques were considered, such as the Levenberg-Marquardt method and the recursive Gauss-Newton method for the NNARX, NNFIR, NNARMAX and NNOE models [31, 36].

Three different residual analyses based on cross correlation were performed: residual analysis between the residual $\hat{R}^N_\varepsilon(\tau)$, between the residual and the input $\hat{R}^N_{\varepsilon u}(\tau)$ and the non-linear residual correlation $\hat{R}^N_{\varepsilon^2 u^2}(\tau)$.

6. Experimentation and Results

The theoretical model has been generated from realistic situations. The model used in this study was implemented in HTB2 [29] and used to gather the initial dataset. The main output of a HTB2 simulation is the heater gain –the power requirements in the modeled building-, but also the Fabric gain -heat flux- the temperature and other variables in Tab. I.

The realistic materials in the construction, the volumetric measures of each room, the neighbourhood of the rooms, the orientation and geographical earth zone, the solar radiation profile, the environment data, the heating subsystems, the occupancy profile, the temperature-time profile for each heating subsystem, the small power devices and the light ON profiles were considered, among others, to validate the proposal. A building in the E winter zone, in the city of Avila is used as the actual building location. Different sample periods and the length of the simulations have been fixed too.

This initial dataset has been analyzed, then, in order to select the features that best describe the relationships with the heat flux. As may be seen in Fig. 2, PCA (Fig. 2.a) and CMLHL (Fig. 2.b), both methods have identified the occasional gain as the most relevant variable but more structured clusters than in the PCA projections may be noted in the CMLHL projections (Fig. 2.b).

Having analyzed the results obtained with the CMLHL model (Fig. 2.b), it can be concluded that CMLHL has identified four relevant variables and seven clusters ordered by occasional gain. Inside each cluster there are further classifications according to heater gain, ventilation gain and, to a lesser degree, exterior air temperature. Accordingly, it may be said that the heat flux and the dataset have an interesting internal structure. When the dataset is considered sufficiently informative, then the third step of the process begins. This step performs an accurate and efficient optimization of the heating system model to detect the heat flux model in the building, through the application of several conventional modeling systems.

Thus, an ANN was used to monitor the thermal dynamics of the building. The objective was to find the best suite of polynomial model orders [n_a, n_{b1}, n_{b2}, n_{b3}, n_{b4}, n_c, n_d, n_f, n_{k1}, n_{k2}, n_{k3}, n_{k4}]. Using the dataset from the previous stage and

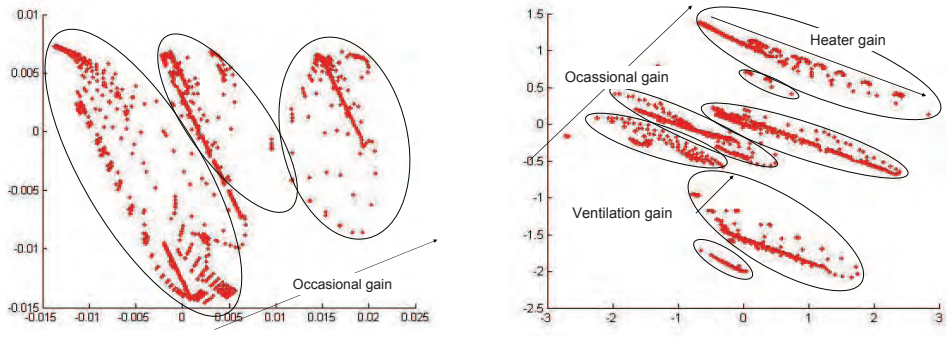


Fig. 2 *PCA projections in the left figure (Fig. 2.a) and CMLHL projection in the right figure (Fig. 2.b) after 20000 iterations using a learning rate of 0.05, 3 output neurons $p = 0.3$ and $\tau = 0.3$.*

the Optimal Brain Surgeon (OBS) [22, 24] network pruning strategy to remove superfluous weights, the best suite model was found from the residual analysis. Tab. II shows the estimation and prediction characteristics and qualities of the chosen ANN, along with their indexes.

Model	**Indexes**
ANN model for the heating process, NNARX regressor, the order of the polynomials of the initial fully connected structure are n_a=4, n_{b1}=4, n_{b2}=5, n_{b3}=1, n_{b4}=4, n_{k1}=2, n_{k2}=2, n_{k3}=2, n_{k4}=2, [4 4 5 1 4 2 2 2 2]. The model was obtained using the regularized criterion. This model was optimized by CMLHL analysis, residual analysis and the pruned network, using OBS. The model structure has 10 hidden hyperbolic tangent units and 1 linear output unit. The network is estimated using the Levenberg-Marquardt method, and the model order is decided on the basis of the best AIC criterion of the ARX model.	FIT1:91.4% V:0.0068 FPE:0.12 NSSE:0.0049

Tab. II *The value of the quality indexes obtained for the proposed model. FIT1, V, NSSE and FPE stand for the graphical representation percentage, the loss function error, the normalized sum of squared error and the final prediction error.*

Fig. 3 shows the time responses of the heat flux -$y_1(t)$- and of the estimated heat flux -$\hat{y}_1(t|m)$- for the NNARX model [40]. The x-axis shows the number of samples used in the estimation and validation of the model and the y-axis represents the normalized output variable range, which is the normalized heat flux of the house. The estimation and validation datasets include 2000 and 1126 samples, respectively, and have a sampling rate of 1 sample/minute. Fig. 4 indicates the final neural network structure chosen for modeling heat flux, both of which are

polynomial model orders. These orders specify the inputs to the ANN -four for a full connected- and the indices of the orders represent each of the thermal system inputs.

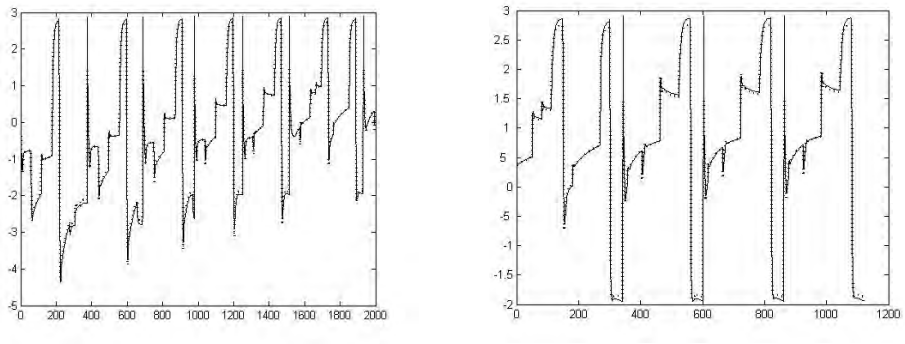


Fig. 3 *Output response of NNARX model: the actual output (solid line) is graphically presented with one-step-ahead prediction (dotted line). In Fig. 3.a (left) the real measure can be compared with the estimated data, while in Fig. 3.b (right) the real measure is compared with the validation data.*

It can be concluded from Fig. 4 that the pruned network of the NNARX model is able to simulate and predict the behavior of the heat flux through exterior walls in a building as a consequence of the heating process: and it is capable of modeling more than 91.4% of the actual measurements. This model does not only present a lower loss function (V) and error values (NSSE and FPE), but also a higher system representation index value (FIT1).

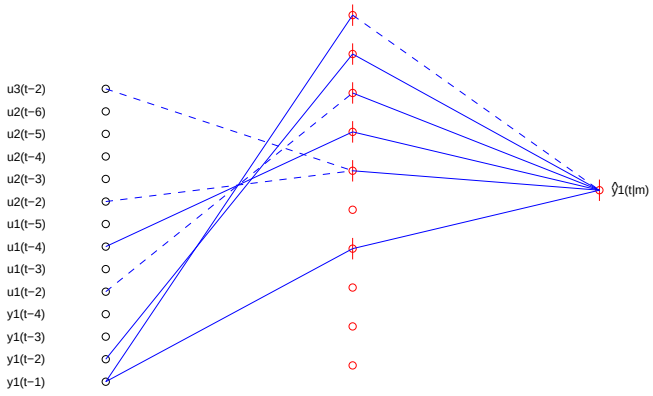


Fig. 4 *Optimal architecture of the NNARX model, with the pruned network for the heat flux through the exterior walls of the building -output $\hat{y}_1(t|m)$-. Positive weights are represented in solid lines, while a dashed line represents a negative weight. A vertical line through the neuron represents a bias.*

7. Conclusions and future work

Effective thermal insulation is an essential component of energy efficient heating systems in buildings. Thus, the possibility of improving the detection of thermal insulation failures represents a fresh challenge for building energy management.

The new methodology proposed in this study to diagnose insulation failures from the heat flux through exterior walls in the building can be used to determine the normal operating conditions of thermal insulation in buildings in Spain, which has recently became a mandatory test in the evaluation of building insulation.

The novel soft computing diagnostic system as presented here improves fault detection with respect to detection systems that rely on isolated signals -used in the industrial processes-. The detection is based in the analysis of the numeric value of the error -difference between the responses of the real heat flux and the estimated heat flux in the building- and the representation percentage of the estimated output in relation to the true output. This analysis presents a low dependency respect to the input signals.

Future work will create a standard of theoretical failures -dataset- in the normal conditions of heating, lighting, small power devices, occupancy and ventilation, so that the diagnostic system in the building -thermal system- can incorporate a global fault classifier. Moreover, automation of the diagnostic system will further improve its performance.

Acknowledgments

We would like to extend our thanks to PhD. Magnus Nørgaard for his freeware version of Matlab Neural Network Based System Identification Toolbox. This research has been partially supported through projects of the Junta of Castilla and León (JCyL): [BU006A08], TIN2010-21272-C02-01 from the Spanish Ministry of Science and Innovation, the project of the Spanish Ministry and Innovation [PID 560300-2009-11], the project of the Spanish Ministry of Education and Innovation [CIT-020000-2008-2] and [CIT-020000-2009-12], the project of the Spanish Ministry of Science and Technology [TIN2008-06681-C06-04] and Grupo Antolin Ingenieria, S.A., within the framework of project MAGNO2008 – 1028.- CENIT, also funded by the same Government Ministry.

References

[1] Alexandru M.: Building on-line diagnosis system upon artificial neural network techniques. International Journal on Non-standard Computing and Artificial Intelligence. Neural Network World, **10**, 6, 2000, pp. 523–534.

[2] Balaras C. A., Argiriou A. A.: Infrared Thermography for Building Diagnostics. Future Generation Computer Systems, **34**, 2, 2002, pp. 171–183.

[3] Batista G. E. A. P. A., Monard M. C.: An Analysis of Four Missing Data Treatment Methods for Supervised Learning. Applied Artificial Intelligence, **17**, 5, 2003, pp. 519–533.

[4] Casillas J., Cordón O., Herrera F., Villar P.: A hybrid learning process for the knowledge base of a fuzzy rule-based system. In: Proceedings of the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU 2004, Perugia, Italy, 2004, pp. 2189–2196.

[5] Chow T. W. S., Wang P., Ma E. W. M.: A New Feature Selection Scheme Using a Data Distribution Factor for Unsupervised Nominal Data. IEEE Transactions on Systems, Man and Cybernetics – PART B: Cybernetics, **38**, 2, 2008, pp. 499–509.

[6] Corchado E., Fyfe C.: Connectionist Techniques for the Identification and Suppression of Interfering Underlying Factors. International Journal of Pattern Recognition and Artificial Intelligence, **17**, 8, 2003, pp. 1447–1466.

[7] Corchado E., Fyfe C.: Orientation Selection Using Maximum Likelihood Hebbian Learning. International Journal of Knowledge-Based Intelligent Engineering Systems, **7**, 2, 2003, pp. 11–30.

[8] Corchado E., Han Y., Fyfe C.: Structuring global responses of local filters using lateral connections. Journal of Experimental & Theoretical Artificial Intelligence, **15**, 4, 2003, pp. 473–487.

[9] Corchado E., MacDonald D., Fyfe C.: Maximum and Minimum Likelihood Hebbian Learning for Exploratory Projection Pursuit. Data Mining and Knowledge Discovery, **8**, 3, 2004, pp. 203–225.

[10] Cybenko G.: Aproximation by Superpositions of Sigmoidal Function. Math. Control, Signals and System, **2**, 4, 1989, pp. 303–314.

[11] de la Cal E., Villar J. R., Sedano J.: A thermodynamical model study for an energy saving algorithm. In: Lecture Notes in Artificial Intelligence: Hybrid Artificial Intelligence Systems, volume **5572**, Springer, 2009 pp. 384–390.

[12] Deogun J., Spaulding W., Shuart B., Li D.: Towards Missing Data Imputation: A Study of Fuzzy K-means Clustering Method. In: Lecture Notes in Computer Science: Proceedings of the 4th International Conference of Rough Sets and Current Trends in Computing (RSCTC'04), volume **3066**, Uppsala (Sweden), Springer, 2004, pp. 573–579.

[13] Directive 2002/91/CE of the European Parliament and the Council of 16 December 2002 on the energy performance of buildings, Oficial Journal of the European Community, 2003.

[14] Feng H. A. B., Chen G. C., Yin C. D., Yang B. B., Chen Y. E.: A SVM regression based approach to filling in Missing Values. In: Lecture Notes in Computer Science: 9th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES2005), volume **3683**, Melbourne (Australia), Springer, 2005, pp. 581–587.

[15] Fletcher R.: Practical Methods of Optimization. Wiley & Sons, Chichester, UK, 2nd edition, 1987.

[16] Friedman J. H., Tukey J. W.: Projection Pursuit Algorithm for Exploratory Data-Analysis. IEEE Transactions on Computers, **23**, 9, 1974, pp. 881–890.

[17] Fyfe C., Baddeley R., McGregor D. R.: Exploratory Projection Pursuit: An Artificial Neural Network Approach. Research Report/94/160. University of Strathclyde, 1994.

[18] Fyfe C., Corchado E.: Maximum Likelihood Hebbian Rules. In: Proceedings of the 10th European Symposium on Artificial Neural Networks(ESANN 2002), Bruges, Belgium, D-side Publishers, 2002, pp. 143–148.

[19] Gourraud P. A., Ginin E., Cambon-Thomsen A.: Handling Missing Values in Population Data: Consequences for Maximum Likelihood Estimation of Haplotype Frequencies. European Journal of Human Genetics, **12**, 10, 2004, pp. 805–812.

[20] Grzymala-Busse J. W., Goodwin L. K., Grzymala-Busse W. J., Zheng X.: Handling Missing Attribute Values in Preterm Birth Data Sets. In: Lecture Notes in Computer Science: Proceedings of the 10th International Conference of Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC'05), volume **3642**, Regina (Canada), Springer, 2005, pp. 342–351.

[21] Han J., Lua L., Yang H.: Investigation on the thermal performance of different lightweight roofing structures and its effect on space cooling load. Applied Thermal Engineering, **29**, 11-12, 2009, pp. 2491–2499.

[22] Hansen L. K., Pedersen M. W.: Controlled Growth of Cascade Correlation Nets. In: International Conference on Artificial Neural Networks (ICANN1994), Italy, Eds. M. Marinaro and P. G. Morasso, 1994, pp. 797–800.

[23] Haralambopoulos D. A., Paparsenos G. F.: Assessing the thermal insulation of old buildings. The need for in situ spot measurements of thermal resistance and planar infrared thermography. Energy Conversion and Management, **39**, 1-2, 1998, pp. 65–79.

[24] Hassibi B., Stork D. G.: Second Order Derivatives for Network Pruning: Optimal Brain Surgeon. In: Advances in Neural Information Processing System 5, San Mateo, CA, USA, Eds. S. J. Hanson et al., 1993, pp. 164–171.

[25] Hertz J., Krogh A., Palmer R. G.: Introduction to the Theory of Neural Computation, volume **1**. Addison-Wesley, Santa Fe Institute Studies in the Sciences of Complexity, 1991.

[26] Hornik K., Stinchcombe M., White H.: Multilayer Feedforward Networks are Universal Aproximators. Neural Networks, **2**, 5, 1989, pp. 359–366.

[27] Hotelling H.: Analysis of a Complex of Statistical Variables Into Principal Components. Journal of Education Psychology, **24**, 1933, pp. 417–444.

[28] Kolokotsa D., Pouliezos A., Stavrakakis G.: Sensor fault detection in building energy management systems. Dynamics of Continuous Discrete and Impulsive Systems-Series B-Applications & Algorithms, **13**, 6, 2006, pp. 221–225.

[29] Lewis P. T., Alexander P. K.: A Flexible Model for Dynamic Building Simulation: HTB2. Building and Environment, **1**, 1990, pp. 7–16.

[30] Lin D. T., Pan D. C.: Integrating a Mixed-Feature Model and Multiclass Support Vector Machine for Facial Expression Recognition. Integrated Computer-Aided Engineering, **16**, 1, 2009, pp. 61–74.

[31] Ljung L.: System Identification. Theory for the User. Prentice-Hall, Upper Saddle River, N. J., USA, 2nd edition, 1999.

[32] Mendez G. M., Leduc-Lezama L., Colas R., Murillo-Pérez G., Ramírez-Cuellar J., López J. J.: Application of Interval Type-2 Fuzzy Logic Systems for Control of the Coiling Entry Temperature in a Hot Strip Mill. In: Lecture Notes in Computer Science: Proceedings of the Hybrid Artificial Intelligent Systems, volume **5572**, 2009, pp. 352–359.

[33] Nelles O.: Nonlinear System Identification. From Classical Approaches to Neural Networks and Fuzzy Models. Springer, Berlin, Germany, 2001.

[34] Pearson K.: On Lines and Planes of Closest Fit to Systems of Points in Space. Philosophical Magazine, **2**, 6, 1901, pp. 559–572.

[35] Real Decreto 314/2006, de 17 de Marzo, por el que se aprueba el Código Técnico de la Edificación. BOE num 74, España, 2006.

[36] Nørgaard M.: Neural network Based System Identification Toolbox. Technical Report 00-E-891, Department of Automation, Technical University of Denmark, 2000.

[37] Ribarić S., Marčetič D., Vedrina D. S.: A knowledge-based system for the non-destructive diagnostics of façade isolation using the information fusion of visual and IR images. Expert Systems with Applications, **36**, 2, 2009, pp. 3812–3823.

[38] Rodriguez F., Guzman J. L., Berenguel M., Ruiz M.: Adaptive Hierarchical Control of Greenhouse Crop Production. International Journal of Adaptive Control and Signal Processing, **22**, 2, 2007, pp. 180–197.

[39] Schoukens J., Rolain Y., Pintelon R.: Improved Approximate identification of Nonlinear Systems. In: 21st IEEE Instrumentation and Measurement Technology Conference, Como, Italy, 2004, pp. 2183–2186.

[40] Sedano J., Cal E., Curiel L., Villar J. R., Corchado E.: Soft Computing for detecting Thermal Insulation Failures in Buildings. In: Proceedings of 9th International Conference on Computational and Mathematical Methods in Science and Engineering, Gijón, Spain, 2010.

[41] Sedano J., Curiel L., Corchado E., de la Cal E., Villar J. R.: A Soft Computing Based Method for Detecting Lifetime Building Thermal Insulation Failures. Integrated Computer-Aided Engineering, IOS Press, **17**, 2, 2010, pp. 103–115.

[42] Seung H. S., Socci N. D., Lee D.: The Rectified Gaussian Distribution. Advances in Neural Information Processing Systems, **10**, 1998, pp. 350–356.

[43] Shalabi L. A., Shaaban Z., Kasasbeh B.: Data Mining: A Preprocessing Engine. Journal of Computer Science, **2**, 9, 2006, pp. 735–739.

[44] Simani S.: Fault diagnosis of a simulated industrial gas turbine via identification approach. International Journal of Adaptive Control and Signal Processing, **21**, 4, 2007, pp. 326–353.

[45] Simani S., Fantuzzi C., Beghelli S.: Diagnosis techniques for sensor faults of industrial processes. IEEE Transactions on Control Systems Technology, **8**, 5, 2000, pp. 848–855.

[46] Söderström T., Stoica P.: Nonlinear System Identification. From Classical Approaches to Neural Networks and Fuzzy Models. Prentice Hall International, London, UK, 1989.

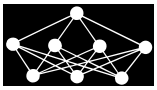
[47] Villar J. R., de la Cal E., Sedano J.: Minimizing Energy Consumption in Heating Systems under Uncertainty. In: Lecture Notes in Artificial Intelligence: Hybrid Artificial Intelligence Systems, Springer, volume **5271**, 2008, pp. 583–590.

[48] Xian G., Zeng B.: An effective and novel fault diagnosis technique based on EMD and SVM. International Journal on Non-standard Computing and Artificial Intelligence. Neural Network World, **20**, 4, 2010, pp. 427–439.

[49] Young P. C., Taylor C. J., Tych W., Pedregal D. J., McKenna P. G.: The Captain Toolbox. Centre for Research on Environmental Systems and Statistics, Lancaster University, 2004.

[50] Yu J., Yang C., Tian L., Liao D.: Evaluation on energy and thermal performance for residential envelopes in hot summer and cold winter zone of China. Applied Energy, **86**, 10, 2009, pp. 1970–1985.

[51] Zadeh L. A.: Soft computing and fuzzy logic. IEEE Software, **11**, 6, 1994, pp. 48–56.



EVALUATING THE PERFORMANCE OF EVOLUTIONARY EXTREME LEARNING MACHINES BY A COMBINATION OF SENSITIVITY AND ACCURACY MEASURES

J. Sánchez-Monedero[*], *C. Hervás-Martínez*[*], *P. A. Gutiérrez*[*],
Mariano Carbonero Ruz[†], *M. C. Ramírez Moreno*[†], *M. Cruz-Ramírez*[*]

Abstract: Accuracy alone can be deceptive when evaluating the performance of a classifier, especially if the problem involves a high number of classes. This paper proposes an approach used for dealing with multi-class problems, which tries to avoid this issue. The approach is based on the Extreme Learning Machine (ELM) classifier, which is trained by using a Differential Evolution (DE) algorithm. Two error measures (Accuracy, C, and Sensitivity, S) are combined and applied as a fitness function for the algorithm. The proposed approach is able to obtain multi-class classifiers with a high classification rate level in the global dataset with an acceptable level of accuracy for each class. This methodology is evaluated over seven benchmark classification problems and one real problem, obtaining promising results.

Key words: *Accuracy, differential evolution, extreme learning machine, multiclass classification, multiobjective, neural networks, sensitivity*

Received: 20th September 2010
Revised and accepted: 13th November 2010

1. Introduction

In recent years, the imbalanced learning problem has drawn a significant amount of interest. The fundamental issue with the imbalanced learning problem is the ability of imbalanced data to significantly compromise the performance of most standard learning algorithms [1]. If the training methods are not proper, the features representing the classes that have a small number of examples in the training

[*]J. Sánchez-Monedero, C. Hervás-Martínez, P. A. Gutiérrez, M. Cruz-Ramírez
Department of Computer Science and Numerical Analysis, University of Córdoba, 14071, Córdoba, Spain, E-mail: {i02samoj,chervas,pagutierrez,i42crram}@uco.es
[†]Mariano Carbonero Ruz, M. C. Ramírez Moreno
Department of Management and Quantitative Methods, ETEA, Córdoba, Spain, E-mail: {mariano,mcramirez}@etea.com

set may likely be ignored by the classifiers. This problem is more serious when the dataset has a high level of noise [2].

To evaluate a classifier, the machine learning community has traditionally used the correct classification rate or accuracy to measure its default performance. In the same way, accuracy has been frequently used as the fitness function in evolutionary algorithms when solving classification problems. However, the pitfalls of using accuracy have been pointed out by several authors [3]. Actually, it is enough to simply realize that accuracy cannot capture all different behavioural aspects found in two different classifiers. Therefore, we assume the premise that a good classifier should combine a high classification rate level in the testing set with an acceptable level for each class. We can consider the traditionally used accuracy (C) and the minimum of the sensitivities of all classes (S), that is, the lowest percentage of examples correctly predicted as belonging to each class with respect to the total number of examples in the corresponding class [4].

On the other hand, Huang et al. have recently proposed an original algorithm called Extreme Learning Machine (ELM) [5], which randomly chooses hidden nodes and analytically determines (by using Moore-Penrose generalized inverse) the output weights of the network. The algorithm tends to provide good testing performance at an extremely fast learning speed. However, the ELM may need a higher number of hidden nodes due to the random determination of the input weights and hidden biases. In [6], a hybrid algorithm called Evolutionary ELM (E-ELM) was proposed by using the differential evolution algorithm [7]. The experimental results obtained show that this approach reduces the number of hidden nodes and obtains more compact networks. The ELM and its extensions have been applied to microarray gene expression cancer diagnosis [8], sales forecasting [9], real-time watermarking [10] and other problems.

In this paper, the simultaneous optimization of accuracy and sensitivity is carried out by means of a slight modification of the E-ELM algorithm. The key point of this modification is the considered fitness function, which tries to take both C and S objectives into account. A convex linear combination of both tries to achieve a good balance between the classification rate level in the global dataset and an acceptable level for each class. The paper is structured as follows. First, we present the sensitivity versus accuracy pair (S, C). Secondly, some related works are presented. Then, the evolutionary approach and its characteristics are introduced. Section 4 analyses the results obtained in seven benchmark classification problems and one real problem. The last section includes the main conclusions of the work.

2. Accuracy and Sensitivity

A classification problem with Q classes and N training or testing patterns is considered, with g as a classifier obtaining a $Q \times Q$ contingency or confusion matrix $M(g) = \left\{ n_{ij};\ \sum_{i,j=1}^{Q} n_{ij} = N \right\}$, where n_{ij} represents the number of times the patterns are predicted by classifier g to be in class j when they really belong to class i. The main diagonal corresponds to the correctly classified patterns and the off-diagonal to the mistakes in the classification task.

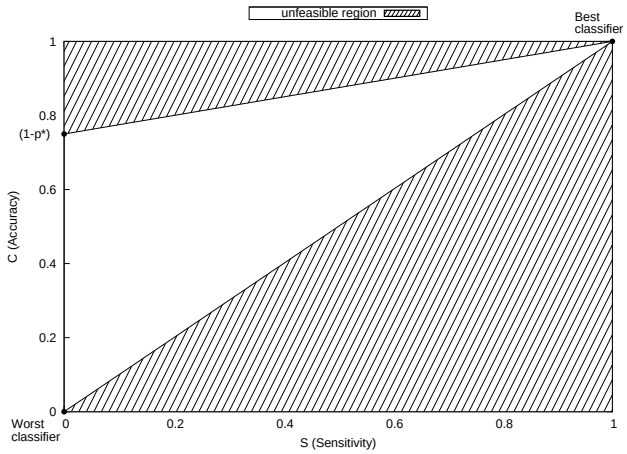


Fig. 1 *Unfeasible region in the two-dimensional space for a concrete classification problem.*

The number of patterns associated with class i can be denoted by $f_i = \sum_{j=1}^{Q} n_{ij}$, $i = 1, \ldots, Q$. Two scalar measures, which take the elements of the confusion matrix into consideration from different points of view [4, 11] are derived. Let $S_i = n_{ii}/f_i$ be the number of patterns correctly predicted to be in class i with respect to the total number of patterns in i (sensitivity for class i). Therefore, the sensitivity for class i estimates the probability of correctly predicting a class i example. From the above quantities, the sensitivity S of the classifier is defined as the minimum value of the sensitivities for each class, $S = \min\{S_i;\ i = 1, \ldots, Q\}$. Moreover, the Correct Classification Rate or Accuracy is the rate of all the correct predictions: $C = (1/N) \sum_{j=1}^{Q} n_{jj}$.

The two-dimensional measure (S, C) associated to a classifier g is an interesting alternative for representing its behaviour (S on the horizontal axis and C on the vertical axis). A classifier depicted in this space is giving information about two of its features: the global performance and the performance in each class. One point in (S, C) space dominates another if it is above and to the right, i.e., it has more accuracy and greater sensitivity.

It is straightforward to prove the following relationship between C and S (see [11]). Let us consider a Q-class classification problem. Let C and S be respectively the accuracy and sensitivity associated with a classifier g, then $S \leq C \leq 1 - (1 - S)\, p^*$, where $p^* = f_Q/N$ is the minimum of the estimated prior probabilities.

Therefore, each classifier will be represented as a point outside the shaded region in Fig. 1. Several points in (S, C) space are important to note. The lower left point $(0, 0)$ represents the worst classifier and the optimum classifier is located at the $(1, 1)$ point. Furthermore, the points on the vertical axis correspond to classifiers that are not able to predict any point in a concrete class correctly. Note that it is possible to find classifiers with a high level of C, among them, particularly in problems with small p^* [4].

Our objective is to build an evolutionary algorithm that tries to move the classifier population towards the optimum classifier located in the $(1, 1)$ point in the (S, C) space. We think an evolutionary algorithm could be an adequate scheme allowing us to improve the quality of the classifiers, measured in terms of C and S, directing the solutions towards the $(1, 1)$ point.

3. Related Works

As mentioned in Section 2, our approach tries to build classifiers with C and S simultaneously optimized. These objectives are not always cooperative, especially with high C and S values. Moreover, considering the multiobjective evolutionary framework, C and S are opposite objectives at high levels. This fact justifies the use of a multiobjective approach for the evolutionary algorithm [4] formally called MultiObjective Evolutionary Algorithms (MOEAs).

The idea of designing neural networks within a multiobjective approach was first considered by Abbass in [12, 13]. In that work, the multiobjective problem formulation essentially involved setting up of two objectives, complexity of the network and training error. For addressing that, an algorithm called Memetic Pareto Artificial Neural Networks (MPANN) which uses Pareto differential evolution was proposed, showing improvements with respect to many other MOEAs.

Férnandez et al. [4] extends the NSGA-II algorithm [14] by including C and S as the objectives in the algorithm. In addition, the NSGA-II is hybridized with iRprop$^+$ [15] as the local search procedure, but this algorithm is only applied in specific generations during the evolution, the resulting algorithm is called MPENSGA-II. This Pareto multiobjective approach has been successfully applied in order to solve predictive microbiology problems [16]. These problems are often imbalanced and in general the classes with a smaller number of patterns are the most important classes.

However, it is well-known that Pareto-based approaches are expensive in terms of computational time as pointed out in [17]. This is mainly due to the process of building the Pareto front, when nondominance must be check in a set of feasible solutions.

One alternative for addressing multiobjective problems in a more efficient strategy (in terms of computing time) is to combine objectives into a single function which is normally denominated an *aggregating function*. This option can be suitable when the behaviour of the objective functions is more or less well-known. The weighted sum approach is one alternative for implementing a aggregating function. This method consists of adding the different objective functions with different weights for each one of the functions, then the multiobjective problem is turned into a scalar optimization problem formulated as:

$$\min \sum_{i=1}^{k} w_i f_i(\mathbf{x}),$$

where f_i is an objective function, w_i is the weight coefficient representing the importance of f_i and $\sum_{i=1}^{k} w_i = 1$.

The weighted linear combination proves to be very efficient in practice for certain types of problems, for example in combinatorial multiobjective optimization. Some applications of this technique are schedule evaluation of a resource scheduler or design multiplierless IIR filters [18]. The main disadvantage is that it may be difficult to determine the proper weights.

4. The Proposed Method

4.1 Extreme learning machine and differential evolution

Let us consider the training set given by N samples $D = \{(\mathbf{x}_j, \mathbf{y}_j) : \mathbf{x}_j \in R^K, \mathbf{y}_j \in R^Q, j = 1, 2, \ldots, N\}$, where $\mathbf{x}_j$ is a $k \times 1$ input vector and $\mathbf{y}_j$ is a $Q \times 1$ target vector.

Let us consider a MultiLayer Perceptron (MLP) with M nodes in the hidden layer given by $f(\mathbf{x}, \boldsymbol{\theta}) = (f_1(\mathbf{x}, \boldsymbol{\theta}_1), f_2(\mathbf{x}, \boldsymbol{\theta}_2), \ldots, f_Q(\mathbf{x}, \boldsymbol{\theta}_Q))$:

$$f_l(\mathbf{x}, \boldsymbol{\theta}_l) = \beta_0^l + \sum_{j=1}^{M} \beta_j^l \sigma_j(\mathbf{x}, \mathbf{w}_j), l = 1, 2, \ldots, Q,$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_Q)^T$ is the transpose matrix containing all the neural net weights, $\boldsymbol{\theta}_l = (\beta^l, \mathbf{w}_1, \ldots, \mathbf{w}_M)$ is the vector of weights of the l output node, $\beta^l = \beta_0^l, \beta_1^l, \ldots, \beta_M^l$ is the vector of weights of the connections between the hidden layer and the l-th output node, $\mathbf{w}_j = (w_{1j}, \ldots, w_{Kj})$ is the vector of weights of the connections between the input layer and the j-th hidden node, Q is the number of classes in the problem, M is the number of sigmoidal units in the hidden layer and $\sigma_j(\mathbf{x}, \mathbf{w}_j)$ the sigmoidal function defined by:

$$\sigma_j(\mathbf{x}, \mathbf{w}_j) = \frac{1}{1 + \exp\left(-\left(w_{0j} + \sum_{i=1}^{K} w_{ij} x_i\right)\right)}.$$

Suppose we are training an MLP with M-nodes in the hidden layer to learn the N samples of set D. The linear system $f(\mathbf{x}_j) = \mathbf{y}_j, j = 1, 2, \ldots, N$, can be written in a more compact format as $\mathbf{H}\boldsymbol{\beta} = \mathbf{Y}$, where $\mathbf{H}$ is the hidden layer output matrix of the network.

The ELM algorithm randomly selects the $\mathbf{w}_j = (w_{1j}, \ldots, w_{Kj}), j = 1, \ldots, M$ weights and biases for hidden nodes, and analytically determines the output weights $\beta_0^l, \beta_1^l, \ldots, \beta_M^l$ for $l = 1 \ldots Q$ by finding the least square solution to the given linear system. The minimum norm least-square solution (LS) to the linear system is $\hat{\boldsymbol{\beta}} = \mathbf{H}^\dagger \mathbf{Y}$, where $\mathbf{H}^\dagger$ is the Moore-Penrose (MP) generalized inverse of matrix $\mathbf{H}$. The minimum norm LS solution is unique and has the smallest norm among all the LS solutions.

The Evolutionary Extreme Learning Machine (E-ELM) [6] improves the original ELM by using a Differential Evolution (DE) algorithm. Differential Evolution was proposed by Storn and Price [7] and it is known as one of the most efficient evolutionary algorithms [19]. The E-ELM uses DE to select the input weights between input and hidden layers and Moore–Penrose generalized inverse to analytically determine the output weights between hidden and output layers.

E-ELM-CS Algorithm:
Require: P (Training Patterns), T (Training Tags)
Create a random initial population $\boldsymbol{\theta} = [\mathbf{w}_1, \ldots, \mathbf{w}_k, b_1, \ldots, b_k]$ of size N
for each individual **do**
  $\hat{\boldsymbol{\beta}} = ELM\_output(\mathbf{w}, P, T)$ {Calculate output weights}
  $\phi_\lambda = Fitness(\mathbf{w}, \hat{\boldsymbol{\beta}}, P, T)$ {Evaluate individual}
end for
Select best individual of initial population
while Stop condition is not met **do**
  Mutate random individuals and apply crossover
  for each individual in the new population **do**
    $\hat{\boldsymbol{\beta}} = ELM\_output(\mathbf{w}, P, T)$ {Calculate output weights}
    $\phi_\lambda = Fitness(\mathbf{w}, \hat{\boldsymbol{\beta}}, P, T)$ {Evaluate model}
    Select new individuals for replacing individuals in old population
  end for
  Select the best model in the generation
end while
function $\hat{\boldsymbol{\beta}} = ELM\_output(\mathbf{w}, P, T)$:
Calculate the hidden layer output matrix H
Calculate the output weight $\hat{\boldsymbol{\beta}} = \mathbf{H}^\dagger \mathbf{Y}$
function $\phi_\lambda = Fitness(\mathbf{w}, \hat{\boldsymbol{\beta}}, \lambda, P, T)$:
Build training confusion matrix $\mathbf{M}$
Calculate C and S from $\mathbf{M}$
Get classifier fitness as $\phi_\lambda = \frac{1}{(1-\lambda)C+\lambda S}$

Fig. 1 *E-ELM-CS algorithm pseudocode.*

4.2 The E-ELM-CS algorithm

In this paper, we use a linear combination of C and S to obtain the maximization of both objectives. This option is a good method when there are two objectives and when the first Pareto front has a very small number of models, in some cases only one (see results from MPANN methodology in Balance and Newthyroid datasets in Tab. II). In addition, its computational cost is noticeably lower than a traditional multiobjective approach [17].

Then, we consider the fitness function defined by:

$$(1 - \lambda)C + \lambda S, \tag{1}$$

where λ is a user parameter in the range $[0, 1]$. This function evaluates the performance of a classifier depending on a weighted accuracy and sensitivity.

Our proposed method is implemented by using the Evolutionary ELM [6]. The original E-ELM for classification problems only considers the misclassification rate of the classifier. We have extended the E-ELM to consider both C and S (E-ELM-CS). Since the E-ELM considers an error measure as the fitness which should be

Dataset	Size	#Input	#Classes	Distribution	p^*
Two classes					
BreastC	286	15	2	(201,85)	0.2972
BreastCW	699	9	2	(458,241)	0.3448
HearStalog	270	13	2	(150,120)	0.4444
Multiclass					
Balance	625	4	3	(288,49,288)	0.0784
Gene	3175	120	3	(762,765,1648)	0.2400
Iris	150	4	3	(50,50,50)	0.3333
Newthyroid	215	5	3	(150,35,30)	0.1395
BTX	63	3	7	(9,9,9,9,9,9,9)	0.1429

Tab. I *Datasets used for the experiments.*

minimized, we reformulate our fitness function as:

$$\phi_\lambda = \frac{1}{(1-\lambda)C + \lambda S}.$$

The E-ELM-CS algorithm pseudocode is shown in Fig. 2. Mutation, crossover and selection operations work as described in [6]. It can be checked how the proposed fitness function is applied in order to take account of not only the accuracy of the classifier, but also of its performance over the worst classified class.

5. Experiments

We consider seven datasets with different features taken from the UCI repository [20] and one real-world problem of analytical chemistry (benzene-toluene-xylene (BTX) and their mixtures discrimination, [21]). Tab. I shows the features for each dataset. The experimental design was conducted using a stratified holdout procedure (see Prechelt [22]) with 30 runs, where approximately 75% of the patterns were randomly selected for the training set and the remaining 25% for the generalization set.

The E-ELM-CS is implemented as an extension of E-ELM source code available at the author public website[1]. For the experiments, the crossover and mutator parameters were set up as described in [6]. The number of individuals in the population were 100 and the number of generations were set up to 50. The crossover constant CR was 0.8, the constant factor F, which controls the amplification of the differential variation, was 1, and the tolerance rate was 0.02. The number of hidden nodes of the neural network was obtained by a cross-validation procedure varying the number of hidden nodes between 5 and 20. There have been considered two different experimental studies:

- Firstly, we consider the effect of the λ values over the results obtained. The objective of this study is to evaluate how the E-ELM-CS can achieve very different results depending on the λ value selected, and how different typologies of datasets can demand different λ values.
- Then, we compared the results of the proposal to those obtained by other alternative MLP design methods.

[1] `http://www3.ntu.edu.sg/home/egbhuang/`

5.1 Analysis of the effect of λ values

In this section, we briefly observe the effect of the λ value of the fitness function described in Eq. 1 on the classifier performance in terms of C and S. The results are presented in the (S, C) space described in Section 2 for BreastC, Balance and BTX datasets described in Tab. I. Both training and generalization performance results are presented. Fig. 3 presents results of E-ELM-CS for all the λ values in the set $[0.0, 0.1, \ldots, 1.0]$. The results are the mean of the best models of 30 runs for each configuration. Each subfigure in Fig. 3 shows a box containing a higher scale representation of the most interesting zone. Note that changing the values of λ value gives us different points which are similar to the Pareto front points in multiobjective problems [17].

Subfigure 1a, showing BreastC performance results, clearly proves that C and S can be competitive objectives. Observe that classifiers which are moved through higher sensitivity values lost performance in the global classification accuracy. A trade-off point between increasing minimum sensitivity without losing lots of global accuracy could be classifiers trained with $\lambda = 0.4$ or $\lambda = 0.5$. Looking at the generalization results in Subfigure 1b, it can be checked that the degree of overfitting is not excessively high, and the behaviour of the fitness function for different λ values is quite similar to that in the training set.

Subfigures 1c and 1d show a very clear example of how balance between the two objectives is necessary. The results show that when only considering C ($\lambda = 0.0$), the method cannot improve results for all the classes. Furthermore, Subfigure 1d shows that using only S ($\lambda = 1.0$) as a unique classification performance measurement is neither suitable. Then, we can consider that $\lambda = 0.4$, $\lambda = 0.5$ and $\lambda = 0.6$ have the best results for improving the two measures.

Finally, we comment on the BTX performance results. In Subfigure 1e it can be seen that λ value has not a very significant influence on the results achieved by the E-ELM-CS. However, it should be noticed that using only S ($\lambda = 1.0$) is not suitable, and the best results are obtained by using only C ($\lambda = 0.0$). This makes sense since BTX is a perfectly balanced dataset (see number of patterns per class distribution for the BTX dataset in Tab. I), with a not very high noise level, so the behaviour of the classifiers is usually very similar for all classes.

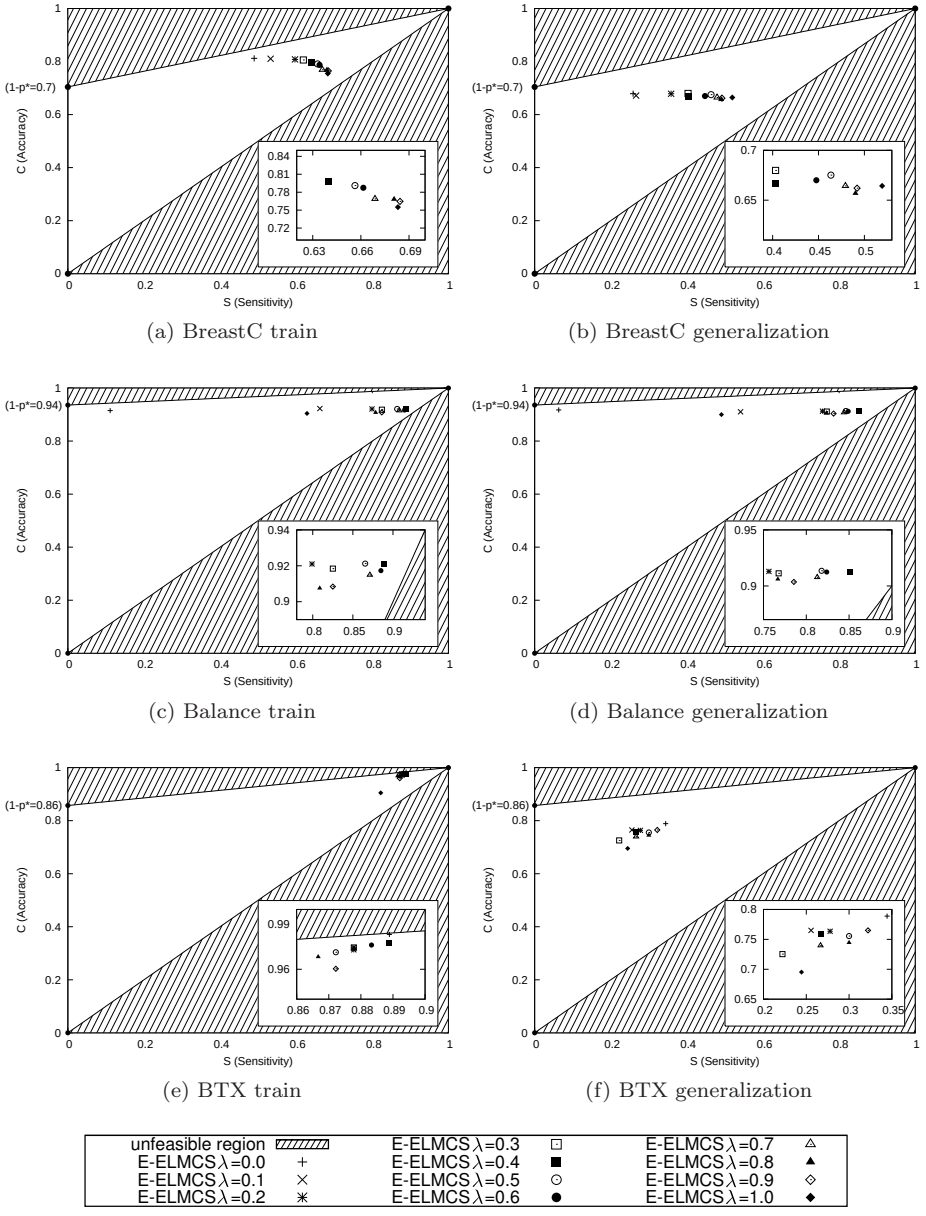
In this preliminary analysis we can conclude that there is no rule for determining the best λ value. Therefore, we propose optimizing this parameter by the cross validation procedure described in the following section.

5.2 Comparison with other MLP training methodologies

In this section, we compare the results obtained with the E-ELM-CS to other three methods: the original E-ELM algorithm, and two popular multiobjective MLP training methods. As previously stated, the original E-ELM algorithm [6] considers only C as the fitness function.

Two additional multiobjective MLP training algorithms are considered for comparison purposes:

1. MPANN [13]. MPANN is a MOEA based on Differential Evolution with two objectives; one is to minimize the Mean Squared Error (MSE), and the other



(a) BreastC train (b) BreastC generalization

(c) Balance train (d) Balance generalization

(e) BTX train (f) BTX generalization

Fig. 1 *Different λ results for BreastC, Balance and BTX databases.*

is to minimize ANN complexity (the number of hidden units). The back propagation algorithm is used in MPANN as local search. We have implemented a Java version using the pseudocode shown in [13] and the framework for evolutionary computation JCLEC [23]. We select both extremes of the Pareto front to compare the results with those from the E-ELM-CS: the methodology is named MPANN-MSE when the extreme selected is that one providing the best MSE, or it is called MPANN-HN if the extreme that is chosen has the best complexity value.

2. TRAINDIFFEVOL (Differential Evolution training algorithm for Neural Networks) [24]. TRAINDIFFEVOL is an algorithm to train feed forward MLP neural networks based on Differential Evolution. This algorithm uses the MSE and mean squared weights and biases for training the networks. To obtain the sensitivity for each class, a modification of the source code provided by the author[2] has been implemented.

From a statistical point of view, these comparisons are possible because we use the same partitions of the datasets. If not, it would be difficult to justify the equity of the comparison procedure. Regarding the settings of each algorithm that has been compared to the E-ELM-CS, we have used the algorithm values advised by the authors in their respective studies. The E-ELM-CS and E-ELM algorithms are set up with the same parameter values for the number of population, number of generations and number of hidden nodes.

In Tab. II we present the values of the mean and the standard deviation (SD) of C and S for 30 runs associated with the best model in each run using the generalization set. For C and S, the best result in each data set is in bold face whereas the second best result is highlighted in italic face.

In the E-ELM-CS, the λ parameter is a user parameter, and it has been obtained as the best result of a preliminary experimental cross-validation design (without considering the generalization set) with $\lambda \in \{0.0, 0.1, \ldots, 1.0\}$. The train data is stratified into 10 sets so 10 validation configurations can be formed. Each one of the 10 validation tests consists of different combinations of 9 sets for training and a different one for generalization. The E-ELM-CS algorithm is run with a different λ value 3 times for each one of these 10 validation tests so there are 30 runs for each λ value. Then, the λ value with maximum validation mean C is selected as the best one. Finally, the E-ELM-CS with the selected λ value is run with the original dataset holdout partitions for comparing with the other methods.

If we analyze the results for C in the generalization set, we can observe that the E-ELM-CS methodology obtains results that are, in mean, better than or similar to the results of the second best methodology in seven datasets. On the other hand, the results in mean of S show that the E-ELM-CS methodology obtains performance that is always better than the second best methodology. In general, the E-ELM-CS improves results of the E-ELM in S.

In order to determine the best methodology for training MLP neural networks (in the sense of its influence on C and S in the dataset), an ANalysis Of the VAriance of one factor (ANOVA I) statistical method or the non-parametric Kruskal-

[2] http://www.it.lut.fi/project/nngenetic/

Dataset	Algorithm	C(%) Mean±SD	S(%) Mean±SD	Means Ranking of the C	Means Ranking of the S
BreastC	E-ELM-CS$_{\lambda=0.4}$	**68.97±3.19**	**33.97±6.82**	$\mu_{\text{ELMCS}} \geq \mu_{\text{TDIF}} \geq \mu_{\text{ELM}} > \mu_{\text{MPANHN}} = \mu_{\text{MPAN}}$(*)	$\mu_{\text{ELMCS}} \geq \mu_{\text{MPAN}} \geq \mu_{\text{MPANHN}} \geq \mu_{\text{TDIF}} > \mu_{\text{ELM}}, \mu_{\text{ELMCS}} > \mu_{\text{MPANHN}}$(○), (T-test)
	E-ELM	68.36±1.98	23.33±6.42		
	TDIF	*68.92±2.89*	26.35±11.71		
	MPANN-MSE	66.53±3.07	*28.73±14.23*		
	MPANN-HN	66.53±3.07	28.41±14.34		
BreastCW	E-ELM-CS$_{\lambda=0.4}$	**96.32±0.86**	**93.87±2.28**	$\mu_{\text{ELMCS}} \geq \mu_{\text{MPANHN}} \geq \mu_{\text{MPAN}} \geq \mu_{\text{ELM}} > \mu_{\text{TDIF}}$(*)	$\mu_{\text{ELMCS}} \geq \mu_{\text{MPANHN}} \geq \mu_{\text{MPAN}} \geq \mu_{\text{ELM}} > \mu_{\text{TDIF}}$(*)
	E-ELM	95.68±1.19	92.61±3.21		
	TDIF	93.98±1.75	86.22±4.69		
	MPANN-MSE	96.04±1.08	92.75±3.40		
	MPANN-HN	*96.27±1.00*	*93.30±3.36*		
Balance	E-ELM-CS$_{\lambda=0.7}$	*91.48±1.50*	**86.74±10.01**	$\mu_{\text{MPANHN}} \geq \mu_{\text{ELMCS}}$(*), (MW)	$\mu_{\text{ELMCS}} > \mu_{\text{MPANHN}}$(*), (MW)
	E-ELM	90.56±1.38	14.00±17.73		
	TDIF	87.12±2.56	2.00±6.10		
	MPANN-MSE	**92.94±1.81**	*60.00±14.14*		
	MPANN-HN	**92.94±1.81**	*60.00±14.14*		
BTX	E-ELM-CS$_{\lambda=0.0}$	**78.89±8.17**	**34.44±23.95**	$\mu_{\text{ELMCS}} = \mu_{\text{ELM}} \geq \mu_{\text{MPAN}} \geq \mu_{\text{TDIFF}}, \mu_{\text{ELMCS}} > \mu_{\text{MPANHN}}$(*)	$\mu_{\text{ELMCS}} \geq = \mu_{\text{ELM}} \geq \mu_{\text{MPAN}} = \mu_{\text{MPANHN}}, \mu_{\text{ELMCS}} > \mu_{\text{TDIFF}}$(*), (MW)
	E-ELM	**78.89±8.17**	**34.44±23.95**		
	TDIF	71.11±3.94	1.11±6.09		
	MPANN-MSE	*72.38±10.85*	*13.33±29.81*		
	MPANN-HN	69.52±11.46	*13.33±29.81*		
Gene	E-ELM-CS$_{\lambda=0.1}$	**83.72±1.93**	**81.10±2.94**	$\mu_{\text{ELMCS}} \geq \mu_{\text{ELM}} > \mu_{\text{MPANHN}} = \mu_{\text{MPAN}} > \mu_{\text{TDIF}}$(*)	$\mu_{\text{ELMCS}} > \mu_{\text{ELM}}$(○), $\mu_{\text{ELMCS}} > \mu_{\text{MPAN}} = \mu_{\text{MPANHN}}, \mu_{\text{ELMCS}} > \mu_{\text{TDIFF}}$ (*), (MW)
	E-ELM	*83.48±1.90*	*78.89±4.97*		
	TDIF	61.18±9.20	35.10±9.13		
	MPANN-MSE	75.11±4.82	36.25±3.90		
	MPANN-HN	75.11±4.82	36.25±3.90		
Heart	E-ELM-CS$_{\lambda=0.3}$	**77.45±2.87**	**64.00±5.13**	$\mu_{\text{ELMCS}} \geq \mu_{\text{MPANHN}} = \mu_{\text{MPAN}} \geq \mu_{\text{TDIF}}, \mu_{\text{MPAN}} > \mu_{\text{ELM}}$(*)	$\mu_{\text{ELMCS}} \geq \mu_{\text{MPAN}} = \mu_{\text{MPANHN}}, \mu_{\text{ELMCS}} \geq \mu_{\text{ELM}}, \mu_{\text{ELMCS}} \geq \mu_{\text{TDIFF}}$ (*), (MW)
	E-ELM	75.29±2.51	61.78±3.69		
	TDIF	76.32±2.02	60.00±3.82		
	MPANN-MSE	*76.91±1.10*	*62.68±2.21*		
	MPANN-HN	*76.91±1.10*	*62.68±2.21*		
Iris	E-ELM-CS$_{\lambda=0.9}$	**97.41±1.76**	**94.53±11.24**	$\mu_{\text{ELMCS}} > \mu_{\text{MPANHN}}, \mu_{\text{ELMCS}} > \mu_{\text{ELM}}, \mu_{\text{TDIFF}} > \mu_{\text{MPAN}}$ (*), (MW)	$\mu_{\text{ELMCS}} > \mu_{\text{ELM}}, \mu_{\text{ELMCS}} \mu_{\text{TDIF}}$ (*), (MW)
	E-ELM	97.04±2.21	*92.18±4.98*		
	TDIF	*97.18±1.03*	91.54±3.10		
	MPANN-MSE	95.30±9.85	86.16±29.51		
	MPANN-HN	94.53±11.24	83.85±33.70		
Newthy	E-ELM-CS$_{\lambda=0.9}$	**96.23±2.31**	**80.85±11.88**	$\mu_{\text{ELMCS}} \geq \mu_{\text{MPANHN}} \geq \mu_{MPAN} \geq \mu_{\text{ELM}} > \mu_{\text{TDIF}}; \mu_{\text{ELMCS}} > \mu_{\text{MPANHN}}$(○), (T-test)	$\mu_{\text{ELMCS}} \geq \mu_{\text{MPANHN}}$(*), (MW)
	E-ELM	94.26±2.35	*75.77±10.16*		
	TDIF	91.11±4.77	59.47±22.74		
	MPANN-MSE	*94.87±3.82*	72.11±22.29		
	MPANN-HN	*94.87±3.82*	72.11±22.29		

(*),(○): The average difference is significant with $p-$value = 0.05 (*) or 0.10 (○)
(MW), (T-test): Normality hypothesis is not satisfied, so Kruskal-Wallis and Mann-Whitney or Wilcoxon T tests are applied

Tab. II *Statistical results for E-ELM-CS, E-ELM, TRAINDIFFEVOL, MPANN-MSE and MPANN-HN in generalization.*

Wallis (KW) tests were applied depending on the satisfaction of the normality hypothesis of C and S values. The levels of the factor represent the methodology applied, and they are the following: $i = 1 \ldots 5$, corresponding to E-ELM-CS (ELMCS), E-ELM (ELM), TRAINDIFFEVOL (TDIF), MPANN-MSE (MPAN) and MPANN-HN (MPANHN). The results of the ANOVA or KW analysis for C

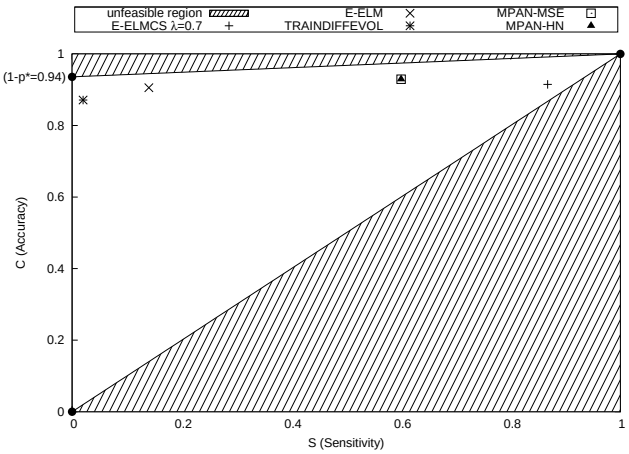


Fig. 4 *Comparison of E-ELM-CS, E-ELM, TRAINDIFFEVOL, MPANN-MSE and MPANN-HN methods for Balance database.*

and S show that, for the eight datasets, the effect of the methodologies is statistically significant at a level of 5%.

Now we apply post hoc tests because the previous tests found significant differences in mean for C and S for all datasets; a post hoc multiple comparison test of the mean C and S obtained with the different levels of the factor is performed. We have chosen a Tukey test [25] for those datasets where the normality hypothesis is satisfied, and a pair-wise Wilcoxon T-test, or a pair-wise Mann-Whitney test in other cases. Columns 5 and 6 of Tab. II present the results obtained by using the post hoc Tukey test, or the Mann-Whitney (MW) test or Wilcoxon T-test. The mean difference is significant with $p-$value $= 0.05$ (*) or 0.10 ($\circ$). In this table, $\mu_A \geq \mu_B$ means that methodology A yields better results than methodology B, but the difference is not statistically significant; $\mu_A > \mu_B$ indicates that methodology A yields better results than methodology B with significant differences. It is important to note that both the binary relations $\geq$ and $>$ are not transitive.

Observe that there is a relationship between the imbalanced degree of the dataset and the results obtained by the E-ELM-CS algorithm. It is worthwhile to point out that, for imbalanced datasets, the E-ELM-CS gets the best performance results and the highest differences in S when comparing the algorithms (see Balance and Newthyroid results in Fig. 4. On the other hand, even for two class problems we can observe the same behaviour (compare the S results of BreastC-W and BreastC datasets). Finally, our approach improves sensitivity levels with respect to the original Evolutionary Extreme Learning Machine (E-ELM), while maintaining accuracy at the same level.

As an example of usefulness of the (S, C) representation, Fig. 4 depicts the sensitivity-accuracy generalization results (in mean of the results of best individuals of 30 runs) of the four methodologies for the Balance dataset in the (S, C) space. A visual inspection of the figure allows us to easily observe the difference in the performance of E-ELM-CS with respect to E-ELM, TRAINDIFFEVOL and MPANN.

6. Conclusions

This work proposes a new approach to dealing with multi-class classification problems. Assuming that a good classifier should combine a high classification rate level in the global dataset with an acceptable level for each class, we consider traditionally used accuracy, C, and the minimum of the sensitivities of all classes, S. The Differential Evolution algorithm and the fast ELM algorithm are used for optimization of both measures in a multiobjective optimization approach, by using a fitness function built as a convex linear combination of S and C. The procedure obtains multi-class classifiers with a high classification rate level in the global dataset with a good level of accuracy for each class. The proposed method makes the ELM algorithm applicable for datasets with a high level of imbalance or with a high level of noise per each class.

Some suggestions for future research are the following: to study other fitness functions based on the (S, C) measures, to adapt the algorithm in order to deal with moderate imbalanced problems, and to evaluate the suitability of other basis functions in this context (e.g. Product Units [26], Radial Basis Functions [27, 28], ...).

Acknowledgement

This work has been partially subsidized by the TIN 2008-06681-C06-03 project of the Spanish Inter-Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P08-TIC-3745 project of the "Junta de Andalucía" (Spain). The research of Javier Sánchez-Monedero has been funded by the "Junta de Andalucía" Ph. D. Student Program.

References

[1] Haibo He, Edwardo A. Garcia: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, **9**, 21, 2009, pp. 1263–1284.

[2] Yi L. Murphey, Hong Guo: Neural learning from unbalanced data. Applied Intelligence, 21, 2004, pp. 117–128.

[3] Foster Provost, Tom Fawcett: Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In: In Proc. of the 3rd International Conference on Knowledge Discovery and Data Mining, 1997, pp. 43–48.

[4] Fernández-Caballero J. C., Martínez-Estudillo F. J., Hervás-Martínez C., Gutiérrez P. A.: Sensitivity versus accuracy in multiclass problems using memetic pareto evolutionary neural networks. IEEE Transactions on Neural Networks, **21**, 5, May 2010, pp. 750–770.

[5] Guang-Bin Huang, Lei Chen, Chee-Kheong Siew: Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Transactions on Neural Networks, **17**, 4, July 2006, pp. 879–892.

[6] Qin-Yu Zhu, A. K. Qin, P. N. Suganthan, Guang-Bin Huang: Evolutionary extreme learning machine. Pattern Recognition, **38**, 10, 2005 pp. 1759–1763.

[7] Storn R., Price K.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, **11**, 4, 1997, pp. 341–359.

[8] Runxuan Zhang, Guang-Bin Huang, N. Sundararajan, P. Saratchandran: Multicategory classification using an extreme learning machine for microarray gene expression cancer diagnosis. IEEE/ACM Transactions on Computational Biology and Bioinformatics, **4**, 3, 2007, pp. 485–495.

[9] Zhan-Li Sun, Tsan-Ming Choi, Kin-Fan Au, Yong Yu: Sales forecasting using extreme learning machine with applications in fashion retailing. Decision Support Systems, **46**, 1, 2008, pp. 411–419.

[10] Wanyu Deng, Lin Chen: Color image watermarking using regularized extreme learning machine. Neural Network World, **20**, 3, 2010, pp. 317–330.

[11] Martínez-Estudillo F. J., Gutiérrez P. A., Hervás-Martínez C., Fernández-Caballero J. C.: Evolutionary learning by a sensitivity-accuracy approach for multi-class problems. In: Proceedings of the 2008 IEEE Congress on Evolutionary Computation (CEC'08), Hong Kong, China, IEEE Press, 2008, pp. 1581–1588.

[12] Abbass H. A., Sarker R., Newton C.: PDE: a Pareto-frontier differential evolution approach formulti-objective optimization problems. In: Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, South Korea, volume **2**, 2001.

[13] Abbass H. A.: Speeding up backpropagation using multiobjective evolutionary algorithms. Neural Computation, **15**, 11, 2003, pp. 2705–2726.

[14] Kalyanmoy Deb, Amrit Pratab, Sameer Agarwal, T. Meyarivan: A fast and elitist multiobjective genetic algorithm: NSGA2. IEEE Transactions on Evolutionary Computation, **6**, 2, 2002, pp. 182–197.

[15] Igel C., Hüsken M.: Empirical evaluation of the improved Rprop learning algorithms. Neurocomputing, **50**, 6, 2003, pp. 105–123.

[16] Fernández-Caballero J. C., Hervás-Martínez C., Martínez-Estudillo F. J., Gutiérrez P. A.: Memetic Pareto Evolutionary Artificial Neural Networks to determine growth/no-growth in predictive microbiology. Applied Soft Computing, **11**, 1, 2011, pp. 534–550.

[17] Coello C. A.: An updated survey of ga-based multiobjective optimization techniques. ACM Computer Surveys, **32**, 2, 2000, pp. 109–143.

[18] Wilson P. B., Macleod M. D.: Low implementation cost IIR digital filter design using genetic algorithms. In: Workshop on Natural Algorithms in Signal Processing, Chelmsford, Essex, UK, 1993.

[19] Poláková R.: A variant of competitive differential evolution algorithm with exponential crossover. Neural Network World, **20**, 1, 2010, pp. 159–169.

[20] Asuncion A., Newman D. J.: UCI machine learning repository, 2007.

[21] Hervás-Martínez C., Silva M., Gutiérrez P. A., Serrano A.: Multilogistic regression by evolutionary neural network as a classification tool to discriminate highly overlapping signals: Qualitative investigation of volatile organic compounds in polluted waters by using headspace-mass spectrometric analysis. Chemometrics and Intelligent Laboratory Systems, **92**, 2, 2008, pp. 179–185.

[22] Prechelt L.: PROBEN1: A set of neural network benchmark problems and benchmarking rules. Technical Report 21/94, Fakultät für Informatik (Universität Karlsruhe), 1994.

[23] Ventura A., Romero C., Zafra A., Delgado J., Hervás C.: JCLEC: a Java framework for evolutionary computation. Soft Computing - A Fusion of Foundations, Methodologies and Applications, **12**, 4, February 2008, pp. 381–392.

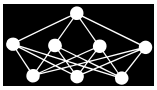
[24] Jarmo Ilonen, Joni-Kristian Kamarainen, Jouni Lampinen: Differential Evolution Training Algorithm for Feed-Forward Neural Networks. Neural Processing Letters, **17**, 1, 2003, pp. 93–105.

[25] Miller R. G.: Beyond ANOVA, Basics of App. Statistics. Chapman & Hall, London, 1996.

[26] Gutiérrez P. A., Hervás-Martínez C., Fernández J. C., Jurado-Expósito M., Peña-Barragán J. M., López-Granados F.: Structural simplification of hybrid neuro-logistic regression models in multispectral analysis of remote sensed data. Neural Network World, **19**, 1, 2009, pp. 3–20.

[27] Xiaogang Gang Zang, Xinbao Gong, Xiaogfeng Ling, Cheng Chang, Bin hua Tang: An evolutionary RBF network configuration using adaptive width adjustment based on vaccination. Neural Network World, **18**, 4, 2008, pp. 323–339.

[28] Parras-Gutierrez E., Del Jesus M. J. J., Rivas V. M., Merelo J. J.: Study of the robustness of a meta-algorithm for the estimation of parameters in radial basis function neural networks design. Neural Network World, **19**, 1, 2009, pp. 81–94.



LEARNING HOSE TRANSPORT CONTROL WITH Q-LEARNING

*Borja Fernandez-Gauna, Jose Manuel Lopez-Guede, Ekaitz Zulueta, Manuel Graña**

Abstract: Non-rigid physical links introduce highly nonlinear dynamics in Multicomponent Robotic Systems (MCRS), which can hardly be solved analytically. In this paper, we propose the use of reinforcement learning methods to allow the agents learn by themselves how to deal with this kind of elements, as opposed to classical control schemes. In this paper we deal with the simplest case: only one hose segment and one robot at the tip of the hose. The task is to move the hose tip to an approximate position in the space. Learning is performed and tested using a hose-MCRS simulation environment developed by our group.

Key words: *Linked Multicomponent Robotic Systems, Q-learning, hose control*

Received: 20th September 2010
Revised and accepted: 13th November 2010

1. Introduction

According to the Multicomponent Robotic System (MCRS) categorization proposed in [4], Linked MCRSs (L-MCRSs) are defined as a collection of autonomous robots linked by a non-rigid physical link. They are distinguished from Distributed MCRS (D-MCRS), which are uncoupled groups of robots, and Modular MCRS (M-MCRSs), which are rigidly linked modular robots.

Modeling these non-rigid links is a non-trivial issue, but it is critical for studying those systems either analytically or via simulation. Some dynamic modeling techniques for non-rigid uni-dimensional objects are reviewed in [5, 6]: differential equations [13], rigid element chains [10], spring mass systems [9], combining spline geometrical models and physical dynamical models [14], and spline models combined with the Cosserat rod theory [17], also known as Geometrically Exact Dynamic Splines (GEDS). Throughout this paper, we will use GEDS, as we believe it is the most adequate model for uni-dimensional objects.

The study of L-MCRS is a novel research and no relevant information about this subject can be found in literature. We started dealing with control and modeling of these systems in [5, 6] and [7]. The latter showed that even a simple spring model

*Borja Fernandez-Gauna, Jose Manuel Lopez-Guede, Ekaitz Zulueta, Manuel Graña
Grupo de Inteligencia Computacional (GIC), Universidad del País Vasco, Spain

of the physical-link in a cooperative control problem introduces highly nonlinear dynamics in the system, making it a hard task to control.

The paradigm of L-MCRS is exemplified by the task of carrying a hose from the origin point to a predefined destination using a collection of autonomous robots attached to the hose. Because of the inherent complexity of robot dynamics, further increased by the complex model of the hose, the system is not solvable in an analytic fashion. In this paper, we propose to solve the control system design problem using Q-Learning, which is a well-known type of Reinforcement Learning (RL) method [16]. RL methods allow autonomous agents to learn based on a reward system and sensorial information. Several authors have approached the robot navigation problem from an RL perspective: [12] applies an RL method to the path-finding problem, [2] applies a quantum computation-inspired variant of Q-Learning to indoor robot navigation, [3] fuses a fuzzy inference system and a Q-Learning algorithm to derive a fuzzy control system, which yields in efficiency and adaptability. Q-Learning was even applied to cooperative navigation in [11], but has never been applied in the presence of physical-links. RL methods can be applied both in the real environment or in a simulated environment. As RL requires a huge amount of attempts to teach the system, whenever possible, it is better to use simulation to learn the system parameters. Simulation avoids tearing down the physical system and it is faster. To approach the problem, we restrict the work in this paper to the case of a single robot managing a hose attached to the fixed end (i.e. the source).

This paper is organized as follows: Section 2 summarizes the hose GEDS model, Section 3 explains some of the specifics and decisions taken to apply the Q-Learning algorithm, and Section 4 describes the experiments carried out in this work, and the results obtained. Finally, our conclusions are given in Section 5.

2. Hose Model

In this chapter we summarize the hose physical model which is the base for the simulation used to train and test the control system of the robot moving the hose. More detailed descriptions can be found in [6, 5]. The combination of spline geometrical modeling and physical dynamical models was introduced by [14]. They allow a continuous definition of uni-dimensional objects. The drawback of the spline model is that, since it is exclusively based on the spline control points, it is unsuitable for representing the hose torsion. The work of [17] has improved the spline representation by combining the spline-based modeling with the Cosserat rod theory [1, 15], allowing to model twisting of the hose. This approach, known as Geometrically Exact Dynamic Splines (GEDS), represents the control points of the splines by three Cartesian coordinates plus a fourth coordinate representing the twisting state of the hose.

2.1 Geometry of the hose

The spline expression for a curve $\mathbf{q}(u)$ is a linear combination of control points $\mathbf{p}_i$ where the linear coefficients are the polynomials $N_i(u)$ which depend on the

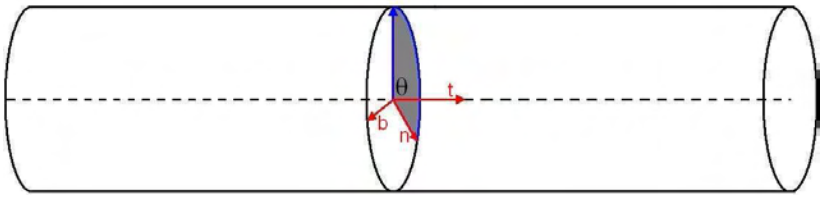


Fig. 1 *Hose section.*

parameter u defined in $[0, 1)$. In the following equation the spline definition is presented:

$$\mathbf{q}(u) = \sum_{i=0}^{n} N_i(u) \cdot \mathbf{p}_i, \tag{1}$$

where $N_i(u)$ is the polynomial associated with the control point $\mathbf{p}_i$, and $\mathbf{q}(u)$ is the point of the curve at the parameter value u. It is possible to travel over the curve by varying the value of parameter u, starting at one end for $u = 0$ and finishing at the other end for $u = 1$. In our work we used B-spline for modeling the hose, so we only need a set of control points, a set of knots and a set of coefficients, one for each control point, so that all curve segments are joined together satisfying the certain continuity condition.

Given $n + 1$ control points $\{\mathbf{p}_0, \ldots, \mathbf{p}_n\}$ and a knot vector $\mathbf{U} = \{u_0, \ldots, u_m\}$, the B-spline curve of degree p defined by these control points and knot vectors $\mathbf{U}$ is:

$$\mathbf{q}(u) = \sum_{i=0}^{n} N_{i,p}(u) \cdot \mathbf{p}_i, \tag{2}$$

where $N_{i,p}(u)$ are B-spline basis functions of degree p ($p = 3$ in this work), built using the Cox de Boor's algorithm. Because the control points of the curve will vary in time, we rewrite equation (2) in terms of the time parameter t:

$$\mathbf{q}(u, t) = \sum_{i=0}^{n} N_{i,p}(u) \cdot \mathbf{p}_i(t). \tag{3}$$

This extended model is named *Dynamic Splines,* and it is the model that we have used for modeling the hose. If we want to take the hose internal dynamics into account, we also need to include the hose twisting at each point given by the rotation of the transverse section around the axis normal to its center point, in order to compute the hose potential energy induced forces. In the GEDS approach, the hose model follows the Cosserat rod approach characterizing it by the curve given by the transverse section centers $\mathbf{c} = (x, y, z)$, and the orientation of each transverse section θ. This description is summarized by the following notation: $\mathbf{q} = (\mathbf{c}, \theta) = (x, y, z, \theta)$. In Fig, 1, the relation between the Cosserat rod director vectors and the twisting angle θ is shown; vector $\mathbf{t}$ represents the tangent to the curve at point $\mathbf{c}$, and vectors $\mathbf{n}$ and $\mathbf{b}$ determine the angle θ of the transverse section at point $\mathbf{c}$.

2.2 Hose dynamical model

Following the Cosserat representation and applying the Lagrange equation (4) a mathematical relation between the potential energy $\mathcal{U}$, the kinetic energy $\mathcal{T}$ and the generalized external forces $\mathbf{F}$ is obtained.

$$\frac{d}{dt}\left(\frac{\partial \mathcal{T}}{\partial \dot{\mathbf{p}}_i}\right) = \mathbf{F}_i - \frac{\partial \mathcal{U}}{\partial \mathbf{p}_i}. \tag{4}$$

The kinetic energy is the motion energy, while the potential energy is the energy due to the hose configuration. Let $\mathbf{F} = \{\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_n\}$ denote the model of the external forces acting on the hose spline model control points. Each $\mathbf{F}_i$ acts on control point $\mathbf{p}_i$. It is usually assumed that mass and stress are homogeneously distributed among the $n+1$ degrees of freedom of the hose spline control model.

When the objects lie in fact in the 2D space we can obviate the moments. Therefore, the potential energy $\mathcal{U}$ is defined by the following integration along the hose, from $u = 0$ up to $u = L$:

$$\mathcal{U} = \frac{1}{2}\int_0^L (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_0)^T \mathbf{F}_{\mathcal{U}} du, \tag{5}$$

where $\mathbf{F}_{\mathcal{U}} = (\mathcal{F}_s, \mathcal{F}_t, \mathcal{F}_b)^T$ are, respectively, the stretching force, and the torsion and bending moments suffered by the hose due to its configuration, $\boldsymbol{\epsilon} = (\epsilon_s, \epsilon_t, \epsilon_b)^T$ is the deformation vector.

The kinetic energy of the hose $\mathcal{T}$ is defined as:

$$\mathcal{T} = \frac{1}{2}\int_0^L \frac{d\mathbf{q}^T}{dt} J \frac{d\mathbf{q}}{dt} du, \tag{6}$$

where J is the inertial matrix. Taking derivatives of the energy expressions, and making adequate substitutions, we come to the following matrix expression of the Lagrange equation:

$$\mathbf{MA} = \mathbf{F} + \mathbf{P}, \tag{7}$$

where $\mathbf{P} = \left[\frac{\partial \mathcal{U}}{\partial \mathbf{p}_i}\right]$, the elements of matrix $\mathbf{M}$ are of the form $\mathbf{M}_{ij} = J\int_0^L (N_i(u)$ $N_j(u))du$, and $\mathbf{A} = \left[\frac{d^2\mathbf{p}_j}{dt^2}\right]$.

2.3 Hose-robots model

The whole system model, composed by the robots and the hose-like linking element, is built from the uni-dimensional element GEDS model by specifying the positions u_r of the robots along the hose. A configuration h of the hose-multi-robot system is defined as:

$$h = \{\mathbf{p}, \mathbf{U}, \mathbf{U_r}\}, \tag{8}$$

where $\mathbf{p}$ is the control point vector of the hose B-spline model, $\mathbf{U}$ is the collection of knots in the B-spline model, $\mathbf{U_r} \subset \mathbf{U}$ is the collection of knots that correspond to robot attachments to the hose. The robot knot vector $\mathbf{U_r} = \{u_{r_i}\}$ contains the

values of the arclength parameter u where the robots are attached to the hose. The spatial position of the i-th robot $\mathbf{r}_i$ is given by the expression: $\mathbf{q}(u_{r_i}) = \sum_{i=0}^{n} N_i(u_{r_i}) \cdot \mathbf{p}_i = \mathbf{r}_i$.

Equation 7 relates the acceleration at the control points with the internal energy of the hose and the external forces applied to it. Among the external forces $\mathbf{F}$ that act on the control points, we differentiate those resulting from the ones applied by the robots $\mathbf{F}_p$ from other external forces $\mathbf{F}_e$:

$$\mathbf{F} = \mathbf{F}_p + \mathbf{F}_e. \tag{9}$$

The relation between the forces applied on the robot attaching points $\mathbf{F_r}$ and the resulting forces on the control points $\mathbf{F_p}$ is given by:

$$\mathbf{F_p} = J_{pr} \cdot \mathbf{F}_r, \tag{10}$$

on the basis of the Jacobian matrix J_{pr} relating robot positions and control points:

$$J_{pr} = \begin{pmatrix} \frac{\partial \mathbf{q}(u_{r_1})}{\partial \mathbf{p}_0} & \cdots & \frac{\partial \mathbf{q}(u_{r_l})}{\partial \mathbf{p}_0} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{q}(u_{r_1})}{\partial \mathbf{p}_n} & \cdots & \frac{\partial \mathbf{q}(u_{r_l})}{\partial \mathbf{p}_n} \end{pmatrix} = \begin{pmatrix} N_0(u_{r_1}) & \cdots & N_0(u_{r_l}) \\ \vdots & \ddots & \vdots \\ N_n(u_{r_1}) & \cdots & N_n(u_{r_l}) \end{pmatrix}. \tag{11}$$

3. Reinforcement Learning

Reinforcement Learning (RL) [16] is a set of methods that enable an agent to learn from experience. Although there are many variants, certain shared elements exist: *policy*, *reward function*, *value function* and, in some cases, *model* of the environment. The *policy* describes a way an agent reacts to the perceived states and picks up an action from those available. The *reward function* inherently describes the goals of the agent, as it returns a number describing how desirable the perceived state to the agent is. This number is called a reward and it is immediate. The *value function* is a long-term version of the reward function (*return*), that is, the total amount of rewards expected from a given state. Learning tasks providing experience to the agent can be continuous or episodic, that is, separate finite episodes.

There are three main families of RL algorithms: Dynamic Programming (DP), Monte-Carlo (MC) methods and Time Difference (TD) learning, each of them having its own strengths and weaknesses. The DP algorithms are mathematically very well founded, but they are computationally expensive and require an accurate model of the environment, which is not always possible. The MC methods and TD learning algorithms do not require a model of the environment, furthermore, they both can learn just from experience, even from simulation of a simple model that provides a sample of the possible transitions among states. MC methods learn on an episode basis, that is, they need to know the actual final return for the learning to be done. On the other hand, TD algorithms are able to deal with online learning tasks, that is, they do not need to know the actual final return of the episode, but only the estimated value one time step ahead, the value prediction is made based

Algorithm 1 Q-learning algorithm

Initialize $Q(s, a)$ arbitrarily
Repeat (for each episode):
  Initialize s
  Repeat (for each step of episode):
    Choose a from s using policy derived from Q
    Take action a, observe reward r and new state s'
    $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \underset{a}{max} Q\left(s_{t+1}, a\right) - Q\left(s_t, a_t\right) \right]$
    $s \leftarrow s'$
  until s is terminal

on the prediction made one step ahead. Both MC and TD methods are known to converge to optimal control. Q-Learning, as described in the following chapter, is a TD method.

3.1 Q-Learning

In its simplest form, Q-Learning is defined by the following iteration:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \cdot \underset{a}{max}\, Q\left(s_{t+1}, a\right) - Q\left(s_t, a_t\right) \right], \qquad (12)$$

where a_t is the action taken at time t, s_t is the state assumed at time t, $Q(s_t, a_t)$ represents the learned action-value discrete map at time t and state s_t, $\alpha \in [0, 1]$ is a step-size parameter that determines how new and old information is averaged, r_{t+1} is the reward at time $t + 1$, a_t is the action taken at time t, and $\gamma \in [0, 1]$ is a discount-rate parameter that indicates the importance of future rewards. Algorithm 1 represents the basic form of the learning algorithm. Thus the learning process is composed of a succession of "episodes": each episode is a complete realization of the behavior of the system, that is, its evolution from an initial state until either the equilibrium state is reached or a stopping condition is met. Time variable t, thus, refers to the time during an episode. The whole learning process is an iteration over the whole matrix Q, which evolves along the episodes. We avoid indexing it for notational simplicity.

The system design phase involves decisions and definitions on various levels of abstraction. On the first level, as we use the reinforcement learning based technique, we have to specify some concepts:

- State: The state has to capture the reality of the scenario in which the problem solution is carried out. It is necessary to get equilibrium between the fidelity of the representation of the world and the quantity of the information that we have to deal with. The definition of the learning state may involve elements of the problem, as well as the dynamics of the system (i.e. the working space). In control processes it may include the control goal.

- Actions: Actions are a set of actions that our agent can perform in the world. They must be discrete.

- Reward system: The reward system specifies the immediate reward that the agent perceives from the environment after doing any possible action. To completely specify a reinforcement system we have to establish the immediate rewards for different scenarios:
  - the goal is reached, i.e. the extreme of the hose attached to the mobile robot reaches the destination point;
  - a failure or forbidden situation occurs, i.e. the mobile robot has collided with the hose;
  - other scenarios that are neither the goal nor failures.

On the second level of abstraction, the Q-Learning specific parameters must be set. Finally, on the third and last level of abstraction we have to consider two practical matters:

- State space and action discretization: As the relationship between state and action is a discrete map, the resolution in the discretization of the state space and the actions is critical to obtain efficient and accurate realizations. Low resolution may allow fast realizations, losing accuracy; conversely high resolutions may hinder the realization of practical experiments.
- Action selection: The mechanism for the generation of actions during the simulation or physical realization of a learning epoch.

4. Experimental Design and Results

The system we will deal with is composed of one hose segment attached to a fixed end (the source) and whose other end (the tip) is transported by a mobile robot attached to it. Fig. 2 exemplifies several configurations of this system. The fixed end is set as the middle of the configuration space. The task for the robot is to bring the tip of the hose to a destination. The working space where the tip-of-the-hose robot moves is a square of size $2 \times 2\text{m}^2$. The specific definitions of the Q-learning experiment realized are the following:

- State: We have defined the state as $S = (P_r, P_d, i)$, where
  - $P_r = (x_r, y_r)$ is the actual position of the tip-of-the-hose robot,
  - $P_d = (x_d, y_d)$ is the desired position of the tip-of-the-hose robot,
  - i is a binary variable that indicates if the line $\overline{P_r P_d}$ intersects the hose. $i = 1$ means that there is such an intersection.
- Working space discretization: In order to follow with the simplest formulation of the problem we have considered a discretization step of $0,5\text{m}$. This discretization determines the cardinality of the universe of states that we work with, and it determines the precision of the coordinates of the point P_r and P_d too. Our working space is, thus, partitioned into 16 boxes. These boxes are the minimum resolution for the placement of a robot. As the robot point

P_r can be in any of these 16 boxes, and the destination point P_d can be in any of these 16 boxes too; there are 256 combinations. Also, the state has another boolean component called i, so there could be a maximum cardinality of 512 possible states.

- Actions: In our problem we can only interact with the scenario using the mobile robot to change the position of the tip-of-the-hose, so the actions are the possible motion directions of the robot. We have chosen a small set of only four actions: $A = \{\, North,\, South,\, East,\, West \,\}$, meaning that the robot will move in this direction for a length equivalent to the size of the resolution box.

- Reward system: We used a simple reward system that gives a positive value to the agent when it reaches the goal, a negative value when the agent fails to reach the goal, and the zero value when the decision is postponed:

$$r \leftarrow \begin{cases} +1 & if \text{ the goal is reached} \\ -1 & if \text{ a failure occurs} \\ 0 & else \end{cases} .$$

  The condition reaching the goal is equivalent to "reaching the same box where the goal is located". As the motion of the tip-of-the-hose robot is of fixed step-size, it is in general impossible to meet a predefined goal point with arbitrary precision.

- α: $[0 < \alpha \leq 1]$, as we suppose that we work in a deterministic environment we can assume that the value of this parameter is 1, so the Q-table update expression 12 simplifies as follows:

$$Q\,(s_t, a_t) \leftarrow r_{t+1} + \gamma \cdot \max_a Q\,(s_{t+1}, a)\,.$$

- γ: $[0 < \gamma \leq 1]$, we have set this value to 0.9.

- Action selection: We have chosen an ϵ-greedy policy. This policy is based on the existence of a parameter ϵ that establishes the equilibrium between the use of the known information (exploitation) and the discovery of new information (exploration), and we have set this value as 0.2. This means that in each step of each episode, with the system being in the state s, we choose the action a with this criterion:

$$a \leftarrow \begin{cases} \max_{a'} Q\,(s,\, a') & \text{with probability } (1 - \epsilon) \\ any\, a' \in A & \text{with probability } \epsilon \end{cases} .$$

- Generation of the initial state: It amounts to the problem of generating a feasible configuration of the hose. To that end, we generate the positions of the spline control points in order from the working space origin (the source) outwards. We generate 10 control points, ensuring that the resulting GEDS will not have excesive bending or streching. Each episode starts from a randomly generated configuration of the hose.

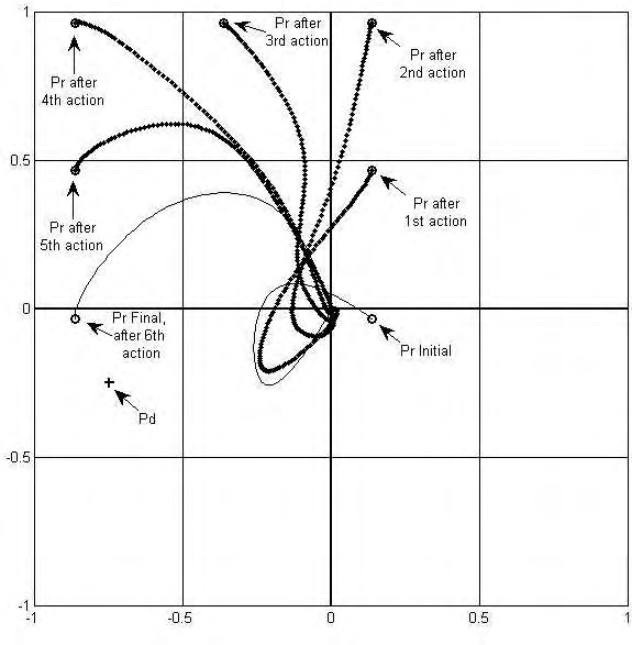


Fig. 2 *Example of the learned behavior.*

Our experiment consisted of 76.000 episodes, and performance was measured applying the learned Q matrix to new 1.000 episodes. The success rate, i.e. the percentage of episodes where the robot reaches the goal, is 73%. 0.7% test episodes concluded because the maximum allowed step count was reached. Finally, 26.3% test episodes failed either because the robot collided with the hose or because the whole system reached a non-feasible position.

In order to illustrate the behavior achieved by this learning method, we have chosen a difficult initial configuration in which the hose is placed between P_r and P_d. In Fig. 2 we show the succesive P_r positions of the robot moving the tip-of-the-hose after each of the actions taken during the episode. The initial hose conguration corresponds to a continuous line. All the intermediate hose configurations carried out until the robot reached the desired cell P_d are shown as dotted lines. It can be easily appreciated how the robot avoids collinding the hose by taking an initially suboptimal strategy (i.e. going away from the goal position). Animations of some test episodes made after the system learns are available online[1].

5. Conclusions

We have approached the hose transportation problem in a L-MCRS using reinforcement learning methods, more specifically, Q-learning. The work in this paper is restricted to a single robot moving the tip of the hose to a desired position,

[1] http://www.ehu.es/ccwintco/index.php/DPI2006-15346-C03-03-Resultados#videos

while the other end is attached to a fixed position. The results of the training computational experiment are very good.

The computational time required to conduct the experiments is one of the biggest issues. The hose model computational requirements are a prime factor, the other factor is the huge number of episodes needed to explore the state-action space before the learned Q-table exploitation can yield good results. Further work will focus on optimization of the state-action space representation, while keeping the most important information required for the learning purpose. The sensitivity of the approach to variations of the value of the α parameter, and the scheduling of action selection probability ε in time will be explored. We plan to apply the learned knowledge on real robots to further validate the results. Future work will involve learning control strategies for a collection of robots attached along the hose. Hierarchical design strategies [8] will be considered.

References

[1] Antman S. S.: Nonlinear Problems of Elasticity. Springer-Verlag, 1995.

[2] Chen C., Yang P., Zhou X., Dong D.: A quantum-inspired q-learning algorithm for indoor robot navigation. In: Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference, 6-8 2008, pp. 1599–1603.

[3] Duan Y., Hexu X.: Fuzzy reinforcement learning and its application in robot navigation. In: Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on, Vol. **2**, 18-21 2005, pp. 899–904.

[4] Duro R. J., Graña M., de Lope J.: On the potential contributions of hybrid intelligent approaches to multicomponen robotic system development. Information Sciences, **180**, 14, 2010, pp. 2635–2648.

[5] Echegoyen Z.: Contributions to Visual Servoing for Legged and Linked Multicomponent Robots. PhD thesis, UPV/EHU, 2009.

[6] Echegoyen Z., Villaverde I., Moreno R., Graña M., d'Anjou A.: Linked multi-component mobile robots: modeling, simulation and control. Robotics and Autonomous Systems, in press, 2010.

[7] Fernandez-Gauna B., Lopez-Guede J. M., Zulueta E.: Linked multicomponent robotic systems: Basic assessment of linking element dynamical effect. In: Maite García-Sebastián Manuel Grana Romay, Emilio S. Corchado, editor, Hybrid Artificial Intelligence Systems, Part I, Springer Verlag, Vol. **6076**, 2010, pp. 73–79.

[8] Graña M., Torrealdea F. J.: Hierarchically structured systems. European Journal of Operational Research, 25, 1986, pp. 20–26.

[9] Gregoire M., Schomer E.: Interactive simulation of one-dimensional flexible parts. Computer-Aided Design, **39**, 8, 2007, pp. 694–707.

[10] Hergenrother E., Dhne P.: Real-time virtual cables based on kinematic simulation. In: Proocedings of the WSCG, 2000.

[11] Melo F. S., Ribeiro M. I.: Reinforcement learning with function approximation for cooperative navigation tasks. In: Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, 19-23 2008, pp. 3321–3327.

[12] Miyata S., Nakamura H., Yanou A., Takehara S.: Automatic path search for roving robot using reinforcement learning. In: Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on, 7-9 2009, pp. 169–172.

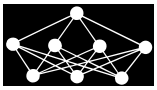
[13] Pai D. K.: Strands: Interactive simulation of thin solids using cosserat models. Computer Graphics Forum, **21**, 3, 2002, pp. 347–352.

[14] Qin H., Terzopoulos D.: D-nurbs: A physics-based framework for geomatric design. Technical report, Los Alamitos, CA. USA, 1996.

[15] Rubin M. B.: Cosserat Theories: Shells, Rods and Points. Kluwer, 2000.

[16] Sutton R. S., Barto A. G.: Reinforcement Learning: An Introduction. MIT Press, 1998.

[17] Theetten A., Grisoni L., Andriot C., Barsky B.: Geometrically exact dynamic splines. Computer-Aided Design, **40**, 1, January 2008, pp. 35–48.



COMBINING CLASSIFIERS USING TRAINED FUSER – ANALYTICAL AND EXPERIMENTAL RESULTS

*Michal Wozniak, Marcin Zmyslony**

Abstract: Combining pattern recognition is a promising direction in designing effective classifiers. There are several approaches to collective decision-making, including quite popular voting methods where the decision is a combination of individual classifiers' outputs. The article focuses on the problem of fuser design which uses discriminants of individual classifiers to make a decision. We present taxonomy of proposed fusers and discuss some of their properties. We focus on the fuser which uses weights dependent on classifier and class number, because of a pretty low computational cost of its training. We formulate the problem of fuser learning as an optimization task and propose a solver which has its origin in neural computations. The quality of proposed learning algorithm was evaluated on the basis of several computer experiments, which were carried out on five benchmark datasets and their results confirm the quality of proposed concept.

Key words: *Pattern recognition, multiple classifier system, trained fuser, neural networks*

Received: 20th September 2010
Revised and accepted: 13th November 2010

1. Introduction

Thanks to progress in computer science companies have collected huge amounts of data, whose analysis is impossible by human beings. Nowadays, simple methods of data analysis are not sufficient for efficient management of an average enterprise since knowledge hidden in data is highly required for smart decisions. A testimony of the mentioned trend is fast progress of machine learning approaches. One of the most popular data mining task is classifier designing, whose aim is to classify the object to one of the predefined categories, on the basis of its feature values. The aforementioned methods are usually applied to many practical areas, like credit approval, prediction of customer behavior, fraud detection, designing of IPS/IDS,

*Michal Wozniak, Marcin Zmyslony
Department of Systems and Computer Networks, Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland, E-mail: [michal.wozniak, marcin.zmyslony]@pwr.wroc.pl

medical diagnosis, to enumerate only a few. Numerous approaches have been proposed to construct efficient classifiers like neural networks, statistical learning, and symbolic learning [2]. For a practical decision problem we can usually have several classifiers at our disposal. It causes that methods of designing Multiple Classifier Systems (MCSs), which can exploit individual classifier strengths, are steadily growing. One of the most important issues while building MSCs is how to select a pool of classifiers. We should stress that combining similar classifiers would not contribute much to the system being constructed, apart from increasing the computational complexity. An ideal ensemble consists of classifiers with high accuracy and high diversity, i.e. they are mutually complementary.

Another important issue is the choice of a collective decision-making method. It is worth noticing that many works consider the quality of the Oracle as the limit of the quality of different fusion methods [17]. The Oracle is an abstract fusion model, where if at least one of the classifiers recognizes an object correctly, then the committee of classifiers points at the correct class too. In this paper, we will consider it is possible to produce such a method of classifier fusion which is capable of achieving higher accuracy than the Oracle. Additionally, we will systematize methods of classifier fusion on the basis of classifier responses and discriminants. Then we will consider which of the presented fusion methods might potentially outperform the Oracle. Our observations will be evaluated on the basis of analytical and experimental researches.

1.1 Fusers based on class labels

The first group of methods includes algorithms for classifier fusion at the level of their responses [18]. Initially, one could find only the majority vote in literature, but in later works more advanced methods were proposed [16, 23].

Let us assume that we have n classifiers $\Psi^{(1)},\ \Psi^{(2)},\ ...,\ \Psi^{(n)}$. For a given object x each of them decides if it belongs to class $i \in M = \{1,\ ..., M\}$. The combined classifier $\bar{\Psi}$ makes a decision on the basis of the following formulae:

$$\bar{\Psi}\left(\Psi^{(1)}(x),\ \Psi^{(2)}(x),\ ...,\ \Psi^{(n)}(x)\right) = \arg\max_{j\in M} \sum_{l=1}^{n} \delta\left(j,\ \Psi^{(l)}(x)\right) w^{(l)} \Psi^{(l)}(x), \quad (1)$$

where $w^{(l)}$ is the weight assigned to the l-th classifier and

$$\delta(j,i) = \begin{cases} 0 & \text{if} \quad i \neq j \\ 1 & \text{if} \quad i = j \end{cases} . \quad (2)$$

Let us note that weights used in (1) play the key role in establishing the quality of $\bar{\Psi}$. There is much research dedicated to weight configurations, e.g. in [22, 10] the authors proposed to train a fuser.

Let us consider the possibilities of weight assigning:

1. Weights $w^{(l)}$ assigned to the classifier – e.g., Kuncheva stated [18] that weights should be assigned according to $w^{(l)} \propto P_{a,\,l}/1 - P_{a,\,l}$ where $P_{a,l}$ denotes probability of accuracy of the l-th classifier.

2. Weights $w^{(l)}(i)$ are assigned to each classifier and each class.

3. Weights $w^{(l)}(i,\ x)$ are assigned to each classifier, each class, and additionally they are dependent on values of feature vector x.

The only model based (partially) on the class label which could achieve better results than the Oracle is a classifier which produces decisions on the basis of class labels given by set of n classifiers $\Psi^{(1)},\ \Psi^{(2)},\ ...,\ \Psi^{(n)}$ and feature vector values. In other words, in this case the decision of the combining classifier depends additionally on the value of the feature vector x i.e., $\bar{\Psi}\left(\Psi^{(1)}(x), \Psi^{(2)}(x),\ ...,\ \Psi^{(n)}(x);\ x\right)$ as distinct from (1). The described model was considered in some papers like [22, 10, 12].

1.2 Fusers based on discriminants

The second group of collective decision-making methods exploits classifier fusion based on discriminants. The main form of discriminants is the posterior probability, typically associated with probabilistic models of the pattern recognition task [5], but it could be given, e.g. by the output of neural networks or that of any other function whose values are used to establish the decision by the classifier. One should cite the work [20], in which the optimal projective fuser was presented, and one has also to mention many other works that describe analytical properties and experimental results, like [8]. The aggregating methods, which do not require a learning procedure, use simple operators, like taking the maximum or average value. However, they are typically subject to very restrictive conditions [6] which severely limit their practical use. Therefore, the design of new fusion classification models, especially those with a trained fuser block, are currently the focus of intense research.

Let us assume that each individual classifier makes a decision on the basis of the values of discriminants. Let $F^{(l)}(i,\ x)$ denote a function that is assigned to class i for a given value of x, and which is used by the l-th classifier $\Psi^{(l)}$. The combined classifier $\hat{\Psi}(x)$ uses the following decision rule [13]

$$\hat{\Psi}(x) = i \quad if \quad \hat{F}(i,\ x) = \max_{k \in M} \hat{F}(k,\ x), \tag{3}$$

where

$$\hat{F}(i,\ x) = \sum_{l=1}^{n} w^{(l)} F^{(l)}(i,\ x) \quad and \quad \sum_{i=1}^{n} w^{(l)} = 1. \tag{4}$$

Let us consider the possibilities of weight assigning:

1. Weights dependent on classifier – this is a traditional approach where weights are connected with classifier and each discriminant of the l-th classifier is weighted by the same value $w^{(l)}$. The estimation of probability error of such classifier could be found in, e.g. [26].

2. Weights dependent on classifier and feature vector – weight $w^{(l)}(x)$ are assigned to the l-th classifier and for a given x have the same value for each dicriminants used by it.

3. Weights dependent on classifier and class number – weight $w^{(l)}(i)$ are assigned to the l-th classifier and the i-th class. For given classifier weights assigned for different classes could be different.

4. Weights dependent on classifier, class number, and feature vector – weight $w^{(l)}(i,x)$ are assigned to the l-th classifier but for given x its value could be diverse for different discriminants assigned to each class.

If we consider the two-class recognition problem only for the last two cases where weights are dependent on classifier and class number it is possible to produce compound classifier which could achieve better quality than Oracle one. But when we take into consideration more than two class problem we could see that it is possible in all aforementioned cases to get results better than Oracle one. In the next section, we will show some analytical properties of fusion methods based on discriminants.

2. Analytical Properties of Fusion Methods

Let us take into consideration two-class recognition problem where i denotes correct class and $\bar{i}$ wrong one. Let us focus on fuser which uses discriminants of individual classifiers multiplied by weights dependent on classifier and feature vector x only. Let us assume that all individual classifiers make wrong decisions, then it is not possible to produce such a fuser which could classifies object correctly, i.e. it is not possible to outperform the Oracle.

Theorem 1.

$$\forall_x \;\; if \quad \forall_{l\in\{1,\ldots n\}} \;\; \Psi^{(l)}(x)=\bar{i} \;\; then \;\; \Psi^{(l)}(x)=\bar{i} \;\; i.e., \;\; \hat{F}(i,x)<\hat{F}(\bar{i},x). \tag{5}$$

Proof It means that

$$\sum_{k=1}^{n} w^{(k)} * F^{(k)}(i \mid x) < \sum_{k=1}^{n} w^{(k)} * F^{(k)}(\bar{i} \mid x). \tag{6}$$

Let us write (6) as

$$\sum_{k=1}^{n} w^{(k)} * (F^{(k)}(i \mid x) - F^{(k)}(\bar{i} \mid x)) < 0. \tag{7}$$

Because

$$\forall_{k\in\{1,\ldots,n\}} \;\; w^{(n)} > 0$$

and all individual classifiers make mistakes

$$\forall_{k\in 1,\ldots,n} \;\; (F^{(k)}(i \mid x) - F^{(k)}(\bar{i} \mid x)) < 0$$

therefore (5) is always true. □

Creating fuser where weights dependent only on classifiers gives also the same results, because it is a special case of the aforementioned model. For three-class

recognition problem situation looks different and it is possible to get correct result even if all classifiers point at wrong classes, which could be presented by the following example.
Let us consider three-class recognition problem and we have 3 individual classifiers at our disposal. Let us assume that a given x belongs to class 3. The supports for each class and classifier are presented in Tab.I.

Classifier	Support for class		
	1	2	3
$\Psi^{(1)}(x)$	0.34	0.36	0.30
$\Psi^{(2)}(x)$	0.50	0.10	0.40
$\Psi^{(3)}(x)$	0.09	0.46	0.45

Tab. I *Exemplary support functions' values.*

Let us note that $\Psi^{(1)}(x) = 2$, $\Psi^{(2)}(x) = 1$, and $\Psi^{(3)}(x) = 2$, which means that all classifiers make mistakes about x, i.e. that every fuser based on class number only cannot classify object correctly. Let us consider a combined classifier based on discriminants which uses averages of support functions given by individual classifiers. In our case, the supports given by this classifier for each class look as follows:

$$\hat{F}(1,x) = 0.31,\ \hat{F}(2,x) = 0.31,\ \hat{F}(3,x) = 0.38,$$

which means that x is classified correctly. We would like to stress that we show only possibility that fuser based on discriminants could produce correct decision even if all individual classifiers are wrong but this approach does not guarantee that we produce fuser which outperforms Oracle classifier. This observation is very interesting because this model is known as "mixture of expert" and several works, such as [14] recognize it as a very flexible and effective approach to produce trained fusers.
Let us consider similar two-class recognition problem again but in this case we use weights which are dependent on classifier and class number. Let

$$W = \left[W^{(1)},\ W^{(2)},\ ...,\ W^{(n)}\right] \tag{8}$$

which consists of weights assigned to each classifier and each class number

$$W^{(l)} = \left[w^{(l)}(1),\ w^{(l)}(2), \ldots, w^{(l)}(M)\right]^{T}. \tag{9}$$

Let us assume that all individual classifiers make wrong decision. Then it is possible to produce such a fuser which points at the correct class, i.e. we could produce fuser which outperforms the Oracle.
Theorem 2.

$$\exists_W \ \ if \ \ \forall_{l\in\{1,\ldots n\}} \ \ \Psi^{(l)}(x) = \bar{i} \ \ then \ \ \hat{\Psi}^{(l)}(x) = i \ \ i.e., \ \ \hat{F}(i,x) < \hat{F}(\bar{i},x). \tag{10}$$

Proof Because combined classifier points at correct class that means

$$\sum_{l=1}^{n} w^{(l)}(i) * F^{(l)}(i \mid x) > \sum_{l=1}^{n} w^{(l)}(i) * F^{(l)}(\bar{i} \mid x). \tag{11}$$

Let us assume that weights and support functions are normalized, i.e.

$$\forall_{l\in\{1,\ldots,n\}}\forall_x \; F^{(l)}(i \mid x) + F^{(l)}(\bar{i} \mid x) = 1 \;\; \textit{and} \; w^{(l)}(i) + w^{(l)}(\bar{i}) = 1. \tag{12}$$

therefore

$$\sum_{l=1}^{n} w^{(l)}(i) * (1 - F^{(l)}(\bar{i}\,|x)) > \sum_{l=1}^{n} w^{(l)}(\bar{i}) * F^{(l)}(\bar{i}\,|x) \tag{13}$$

and finally

$$w^{(1)}(i) + \ldots + w^{(n)}(i) > F^{(1)}(\bar{i}\,|x) + \ldots + F^{(n)}(\bar{i}\,|x)\,.\square \tag{14}$$

From the final form of the inequality we can see that it is possible to get correct final result even if all classifiers are wrong. If the sum of weights assigned to classifiers which point at correct class is bigger than the sum of support functions for incorrect one. Let us notice that this conclusion covers the case when weights dependent on classifiers, class number and feature vector value because it is a special case of the aforementioned model.

We should underline that the theorem 2 shows only possibility to get such a result. In practical terms, it is usually impossible to assign weights in the analytical way. Then let us focus on the problem of establishing weights dependent on classifier and class number only because this case looks very promising and does not need additional prior knowledge about weights contrary to the case where weights are additionally dependent on feature values. If weights depend on x, then they are *de facto* functions, and their estimation is more complicated and usually requires prior knowledge about them.

For the case under consideration, a fuser training task leads to the problem how to establish the vector W (8). The aim is to find out such a fuser which assures the lowest misclassification rate of $\hat{\Psi}$.

In order to solve the aforementioned optimization task, we could use one of a variety of widely used algorithms, e.g. evolutionary or neural approach. In this paper, we present approach which has its origin in neural computations because neural networks are used to model complex relationships between inputs and outputs. Well-known methods of the network training solve the optimization problem by finding such a set of interesting weights. In our study, we decided to use one layer neural network which is an appropriate model for the problem under consideration, and it is presented in Fig. 1.

Trying to solve this optimization problem, some computer experiments were carried out and their results are presented in the next section.

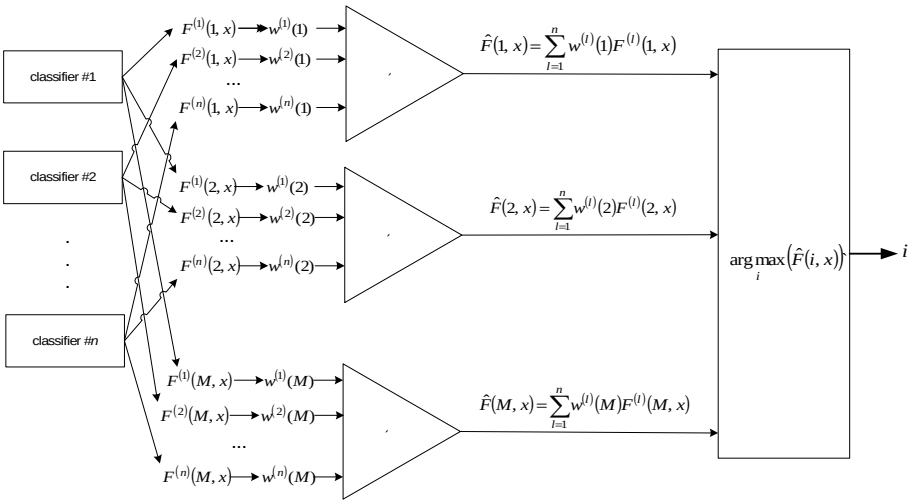


Fig. 1 *One layer neural network as a fuser which uses weights dependent on classifiers and class numbers.*

3. Experimental Investigation

The aim of the experiments is to evaluate the performance of fuser based on weights dependent on classifier and class number.

All the experiments were carried out in *Matlab* environment using *PRTools* toolbox [7] and our own software. The experiments were carried out on 5 benchmark datasets from UCI Machine Learning Repository [3], which are described in Tab. II.

	Dataset	Number of Attributes	Number of Classes	Number of Examples
1	Breast Cancer	10	2	699
2	Connectionist	10	11	528
3	Glass	9	7	214
4	MAGIC	10	2	17117
5	Yeast	10	2	17177

Tab. II *Databases' description.*

For the purpose of this experiment, five neural networks were prepared that could be treated as individual classifiers. They were slightly undertrained (the training process was stopped early for each classifier and we guarantee that classification error of each individual classifiers was lower than random guessing) to ensure their diversity. Classification errors of individual classifiers used during experiments (denoted as C1, C2, C3, C4, C5) are presented in Tab. III.

Dataset	C1	C2	C3	C4	C5
Breast cancer	38,5	34,2	12,7	34,4	16,6
Connectionist	46,1	40,0	40,9	41,9	41,0
Glass	49,7	48,2	49,2	47,6	46,1
MAGIC	40,3	36,0	33,1	32,8	36,0
Yeast	51,5	34,4	21,4	35,2	22,9

Tab. III *Classification errors of individual classifiers used in experiments.*

The remaining details of used neural nets are as follows:

- Five neurons in the hidden layer,
- Sigmoidal transfer function,
- Back propagation learning algorithm,
- Number of neurons in the last layer equals number of classes of given experiment.

During the experiments we wanted to compare quality of two trained fusers

- FCCNV – fuser based on weights dependent on classifier, class number, and feature vector,
- FCCN – fuser based on weights dependent on classifier and class number

with the quality of Oracle classifier. For trained fuser, realized according the idea depicted in Fig. 1, number of training iterations was fixed to 1500. Classification error of individual classifier and fuser models were estimated using 10 Fold cross-validation method [15]. Statistical differences between the performances of the classifiers were evaluated using 10-Fold cv Paired t Test [2] at the significant level 0.05. The results of experimets are presented in Tab. IV.

Dataset	Oracle	FCCNV	FCCN	FCVP1	FCVP2	FCVP3
Breast cancer	2,22	34,39	34,39	0,00	-49,55	-49,55
Connectionist	20,91	15,06	17,65	1,06	3,11	2,76
Glass	39,76	36,64	35,07	0,42	0,41	0,69
MAGIC	13,82	15,52	20,00	14,51	-3,98	-5,99
Yeast	1,8	15,53	17,92	9,32	-73,97	-65,47

Tab. IV *Results of experiments. The first column presents dataset name, columns labeled Oracle, FCCNV, FCCN show classification error of the Oracle, FCCNV, and FCCN respectively. Columns FCVP1, FCVP2, FCVP3 present 10 Fold cv Paired t-statistics for FCCNV vs. FCCN, Oracle vs. FCNV, and Oracle vs. FCNN, respectively.*

The following conclusions can be drawn on the basis of the results of the experiments:

- Classification errors of FCCNV and FCNN are smaller than Oracle classifier for Connectionist dataset only.
- FCCNV outperformed FCNN for MAGIC and Yeast datasets.
- Oracle classifier achieved better quality of classification than FCCNV and FCNN for MAGIC, Breast cancer, and Yeast datasets.
- For Glass dataset we cannot confirm that any classifier is statistically significantly better than the other.

The results of our experiments prove that proposed neural approach is an efficient tool for solving optimization problem of establishing fuser weights. As stated before, when weights depend on the classifier and the class number, it is possible to achieve results that are better than the Oracle classifier. Unfortunately, we can not formulate general conclusions on the basis of the experiments carried out because we still do not know what conditions should be fulfilled to produce high quality combining classifier used proposed fusion method. We would like to underline that a fuser based on weights dependent on classifier and class number could outperform Oracle classifier but chosing such a fuser does not guarantee this property. Proposed fusers (FCCNV and FCCN) outperformed Oracle in one case only, but for the remaining experiments they achieved, except the Yeast dataset, pretty good results.

4. Conclusions

Several methods of classifier fusion were discussed in this paper and two of them were applied into the real decision problem. Obtained results justify the use of weighted combination. As we mentioned above, we still have not discovered conditions which should be fulfilled to produce desirable fuser. Probably it depends on conditional probability distributions of classes for given classification problem, which we have confirmed by our analytical research partly. Unfortunately, for the practical cases as stated above, it is not possible to determine values of weights in the analytical way, therefore, using heuristic methods like neural or evolutionary approaches of optimization seems to be a promising direction of research.

Acknowledgement

This work is supported in part by the Polish State Committee for Scientific Research under a grant for the period 2010-2013.

References

[1] Alexandre L. A., Campilho A. C., Kamel M.: Combining Independent and Unbiased Classifiers Using Weighted Average, Proc. of the 15th Internat. Conf. on Pattern Recognition, vol. **2**, 2000, pp. 495-498.

[2] Alpaydin E.: Introduction to Machine Learning. 2nd edn. The MIT Press, 2004.

[3] Asuncion A., Newman D. J.: UCI ML Repository, Irvine, CA: University of California, School of Information and Computer Science, 2007.
[http://www.ics.uci.edu/˜mlearn/MLRepository.html]

[4] Biggio B., Fumera G., Roli F.: Bayesian Analysis of Linear Combiners, Lecture Notes in Computer Science, vol. 4472, 2007, pp. 292-301.

[5] Bishop Ch. M.: Pattern Recognition and Machine Learning, Springer, 2006.

[6] Duin R. P. W.: The Combining Classifier: to Train or Not to Train?, Proc. of the ICPR2002, Quebec City, 2002.

[7] Duin R. P. W. et al.: PRTools4, A Matlab Toolbox for Pattern Recognition, Delft University of Technology, 2004.

[8] Fumera G., Roli F.: A Theoretical and Experimental Analysis of Linear Combiners for Multiple Classifier Systems, IEEE Trans. on PAMI, 27, 2005, pp. 942-956.

[9] Giacinto G.: Design Multiple Classifier Systems, PhD. thesis, Universita Degli Studi di Salerno, 1998.

[10] Hansen L. K., Salamon P.: Neural Networks Ensembles, IEEE Trans. on PAMI, Vol. 12, No. 10, 1990, pp. 993-1001.

[11] Hashem S.: Optimal linear combinations of neural networks, Neural Networks, 10, 1997, pp. 599-614.

[12] Inoue H., Narihisa H.: Optimizing a Multiple Classifier Systems, LNCS, Vol. 2417, 2002, pp. 285-294.

[13] Jackobs R. A.: Methods for combining experts' probability assessment, Neural Computation, No. 7, 1995, pp. 867-888.

[14] Jordan, M. I., Jacobs R. A.: Hierarchical mixtures of experts and the EM algorithm. Neural Computation, **6**, 2, 1994, pp. 181-214.

[15] Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proc. of the 14th Int. Joint Conf. on Artificial Intell., San Mateo, 1995, pp. 1137-1143.

[16] Kuncheva L. I., Bezdek J. C., Duin R. P. W.: Decision templates for multiple classifier fusion: an experimental comparison, Pattern Recognition, 34, 2001, pp. 299-314.

[17] Kuncheva L. I., Whitaker C. J., Shipp C. A., Duin R. P. W.: Limits on the Majority Vote Accuracy in Classier Fusion, Pattern Analysis and Applications, 6, 2003, pp. 22-31.

[18] Kuncheva L. I.: Combining pattern classifiers: Methods and algorithms, Wiley, 2004.

[19] Marcialis G. L., Roli F.: Fusion of Face Recognition Algorithms for Video-Based Surveillance Systems. In: Foresti G. L., Regazzoni C., Varshney P. (Eds.), Multisensor Surveillance Systems: The Fusion Perspective, Kluwer Academic Publishers, 2003.

[20] Rao N. S. V.: A Generic Sensor Fusion Problem: Classification and Function Estimation, Lecture Notes in Computer Science, Vol. 3077, 2004, pp. 16-30.

[21] Raudys S.: Trainable fusion rules. I. Large sample size case, Neural Networks, 19, 2006, pp. 1506-1516.

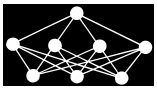
[22] Tumer K., Ghosh J.: Analysis of Decision Boundaries in Linearly Combined Neural Classifiers, Pattern Recognition, 29, pp. 341-348, 1996.

[23] Van Erp, Vuurpijl L. G., Schomaker L. R. B.: An overview and comparison of voting methods for pattern recognition, Proc. of IWFHR. 8, Canada, 2002, pp. 195-200.

[24] Wolpert D. H.: The supervised learning no-free-lunch theorems, Proceedings of the 6th Online World Conference on Soft Computing in Industrial Applications, 2001.

[25] Woods K., Kegelmeyer W. P.: Combination of multiple classifiers using local accuracy estimates, Pattern Analysis and Machine Intelligence, IEEE Transactions, Vol. **19**, Issue 4, Apr 1997, pp. 405-410.

[26] Wozniak M.: Experiments on linear combiners, In: Pietka E, Kawa J. (Eds.) Information technologies in biomedicine, Springer, 2008, pp. 445-452.



NEURAL CLASSIFIERS FOR SCHIZOPHRENIA DIAGNOSTIC SUPPORT ON DIFFUSION IMAGING DATA

Alexandre Savio[*], *Juliette Charpentier*[†], *Maite Termenón*[*], *Ann K. Shinn*[‡], *Manuel Graña*[*]

Abstract: Diagnostic support for psychiatric disorders is a very interesting goal because of the lack of biological markers with sufficient sensitivity and specificity in psychiatry. The approach consists of a feature extraction process based on the results of Pearson correlation of known measures of white matter integrity obtained from diffusion weighted images: fractional anisotropy (FA) and mean diffusivity (MD), followed by a classification step performed by statistical support vector machines (SVM), different implementations of artificial neural networks (ANN) and learn vector quantization (LVQ) classifiers. The most discriminant voxels were found in frontal and temporal white matter. A total of 100% classification accuracy was achieved in almost every case, although the features extracted from the FA data yielded the best results. The study has been performed on publicly available diffusion weighted images of 20 male subjects.

Key words: *DWI, schizofrenia, neural classifiers, fractional anisotropy, mean diffusivity*

Received: 20th September 2010
Revised and accepted: 13th November 2010

1. Introduction

There is growing research effort devoted to the development of automated diagnostic support tools that may help clinicians perform their work with greater accuracy and efficiency. In medicine, diseases are often diagnosed with the aid of biological markers, including laboratory tests and radiologic imaging. The process of diagnosis becomes more difficult, however, when dealing with psychiatric disorders, in which diagnosis relies primarily on the patient's self-report of symptoms and

[*]A. Savio, M. Termenón, M. Graña
Grupo de Inteligencia Computacional (GIC), Universidad del Pas Vasco, Spain
[†]J. Charpentier
Institut Superieur de BioSciences de Paris (ISBS), ESIEE, Universit Paris-Est, France
[‡]A. K. Shinn
McLean Hospital, Belmont, Massachusetts; Harvard Medical School, Boston, Massachusetts, USA

the presence or absence of characteristic behavioral signs. Schizophrenia is a disabling psychiatric disorder characterized by hallucinations, delusions, disordered thought/speech, disorganized behavior, emotional withdrawal, and functional decline [3]. Currently, diagnosis is made almost exclusively on subjective measures like self-report, observation, and clinical history.

A large number of magnetic resonance imaging (MRI) morphological studies have shown subtle brain abnormalities to be present in schizophrenia. Structural studies have found enlargement of the lateral ventricles, particularly the temporal horn of the lateral ventricles [30], reduced volumes of medial temporal structures (hippocampus, amygdala, and parahippocampal gyrus) [5, 18, 31], superior temporal gyrus [18], prefrontal cortex [16, 34], and inferior parietal lobule [29, 15]; and reversal of normal left greater than right volume in male patients with schizophrenia [26, 13]. In 1984, Wernicke [37] proposed that schizophrenia might involve altered connectivity of distributed brain networks that are diverse in function and that work in concert to support various cognitive abilities and their constituent operations. Consistent with the "dysconnectivity hypothesis", studies have found correlations between prefrontal and temporal lobe volumes [38, 8] and disruptions of functional connectivity between frontal and temporal lobes in schizophrenia [24]. These findings strongly point to widespread problems of connectivity in schizophrenia.

Diffusion tensor imaging (DTI) is an MRI method that allows more direct investigation into the integrity of white matter (WM) fibers, and thus into the anatomical connectivity of different brain regions. DTI depends upon the motion of water molecules to provide structural information in vivo [27, 6], and yields measures like fractional anisotropy (FA) and mean diffusivity (MD). The most commonly demonstrated DTI abnormalities in schizophrenia are decreased FA in the uncinate fasciculus (a tract connecting temporal and frontal regions and involved in decision-making, emotions, and episodic memory), the cingulum bundle (a tract interconnecting limbic regions which are involved in attention, emotions, and memory), and the arcuate fasciculus (a tract connecting language regions) [22]. Lower anisotropic diffusion within white matter may reflect loss of coherence of WM fiber tracts, to changes in the number and/or density of interconnecting fiber tracts, or to changes in myelination [20, 23, 2, 21].

The present paper will focus on the application of machine learning (ML) algorithms for the computer aided diagnosis (CAD) of schizophrenia, on the basis of feature vectors extracted from DTI measures of WM integrity, FA and MD. This feature extraction method is based on Pearson correlation, and is simpler than others found in the literature [14, 12]. These features will be the input for statistical SVM and artificial neural networks (ANN) classifiers. We found literature on the application of ML algorithms to the discrimination of schizophrenia patients from healthy subjects. A minimum recognition error of 17,8% using geometry features and FA of DTI from a database of 36 healthy subjects and 34 patients with schizophrenia was reported in [36]. A study of the effect of principal component analysis (PCA) and discriminant PCA (DPCA) was carried on FA volumes reaching a minimum one-leave-out validation classification error 20% using Fisher linear discriminant (FLD) in [10]. Good classification results were also obtained in structural MRI (sMRI) studies [39, 12].

Section 2 gives a summary of the classification algorithms used for this study. Section 3 describes the materials and methods in the study: characteristics of the subjects conforming the database for the study, the acquisition protocol, the preprocessing steps of the MRI and DTI volumes, and the feature extraction process. Section 4 gives the results of our computational experiments. Section 5 gives our final comments and conclusions.

2. Neural Network and Statistical Classification Algorithms

We deal with two class classification problems, given a collection of training/testing input feature vectors $X = \{\mathbf{x}_i \in \mathbb{R}^n,\ i = 1, \ldots, l\}$ and the corresponding labels $\{y_i \in \{-1, 1\},\ i = 1, \ldots, l\}$, which sometimes can be better denoted in aggregated form as a binary vector $\mathbf{y} \in \{-1, 1\}^l$. The algorithms described below build some classifier systems based on this data. The simplest algorithm is the 1-nearest neighbor (1-NN) which involves no adaptation and uses all the training data samples. The classification rule is of the form:

$$c\left(\mathbf{x}\right) = y_{i^*} \text{ where } i^* = \arg\min_{i=1,\ldots,l} \left\{\|\mathbf{x} - \mathbf{x}_i\|\right\},$$

that is, the assigned class is that of the closest training vector. To validate their generalization power we use ten-fold cross-validation.

2.1 Support vector machines

The support vector machine (SVM) [35] approach to build a classifier system from the given data consists in solving the following optimization problem:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i, \tag{1}$$

subject to

$$y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq (1 - \xi_i),\ \xi_i \geq 0,\ i = 1, 2, \ldots, n. \tag{2}$$

The minimization problem is solved via its dual optimization problem:

$$\min_{\alpha} \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha}, \tag{3}$$

subject to

$$\mathbf{y}^T\boldsymbol{\alpha} = 0,\ 0 \leq \alpha_i \leq C,\ i = 1, \ldots, l. \tag{4}$$

Where $\mathbf{e}$ is the vector of all ones, $C > 0$ is the upper bound on the error, Q is an $l \times l$ positive semidefinite matrix, whose elements are given by the following expression:

$$Q_{ij} \equiv y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \tag{5}$$

where

$$K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j), \tag{6}$$

is the kernel function that describes the behavior of the support vectors. Here, training vectors $\mathbf{x}_i$ are mapped into a higher (maybe infinite) dimensional space by the function $\phi(\mathbf{x}_i)$. The decision function is:

$$sgn(\sum_{i=1}^{l} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b). \tag{7}$$

The regularization parameter C is used to balance the model complexity and the training error. It was always set to 1 in this case study.

The chosen kernel function results in different kinds of SVM with different performance levels, and the choice of the appropriate kernel for a specific application is a difficult task. In this study we only needed to use a linear kernel, defined as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = 1 + \mathbf{x}_i^T \mathbf{x}_j, \tag{8}$$

this kernel shows good performance for linearly separable data.

2.2 Backpropagation

Backward propagation of errors, or backpropagation (BP), [28, 17] is a non-linear generalization of the squared error gradient descent learning rule for updating the weights of artificial neurons in a single-layer perceptron, generalized to feed-forward networks, also called multi-Layer perceptron (MLP). Backpropagation requires that the activation function used by the artificial neurons (or "nodes") is differentiable with its derivative being a simple function of itself. The backpropagation of the error allows to compute the gradient of the error function relative to the hidden units. It is analytically derived using the chain rule of calculus. During on-line learning, the weights of the network are updated at each input data item presentation. We have used the resilient backpropagation which uses only the derivative sign to perform the weight updating.

We restrict our presentation of BP to train the weights of the MLP for the current two class problem. Let the instantaneous error E_p be defined as:

$$E_p(\mathbf{w}) = \frac{1}{2}\left(y_p - z_K(\mathbf{x}_p)\right)^2, \tag{9}$$

where y_p is the p-th desired output y_p, and $z_K(\mathbf{x}_p)$ is the network output when the p-th training exemplar $\mathbf{x}_p$ is inputted to the MLP composed of K layers whose weights are aggregated in the vector $\mathbf{w}$. The output of the j-th node in layer k is given by:

$$z_{k,j}(\mathbf{x}_p) = f\left(\sum_{i=0}^{N_{k-1}} w_{k,j,i} z_{k-1,i}(\mathbf{x}_p)\right), \tag{10}$$

where $z_{k,j}$ is the output of node j in layer k, N_k is the number of nodes in layer k, $w_{k,j,i}$ is the weight which connects the i-th node in layer $k-1$ to the j-th node in layer k, and $f()$ is the sigmoid nonlinear function, which has a simple derivative:

$$f'(\alpha) = \frac{df(\alpha)}{d\alpha} = f(\alpha)(1 - f(\alpha)). \tag{11}$$

The convention is that $z_{0,j}(\mathbf{x}_p) = \mathbf{x}_{p,j}$. Let the total error E_T be defined as follows:

$$E_T(\mathbf{w}) = \sum_{p=1}^{l} E_p(\mathbf{w}), \tag{12}$$

where l is the cardinality of X. Note that E_T is a function of both the training set and the weights in the network. The backpropagation learning rule is defined as follows:

$$\Delta w(t) = -\eta \frac{\partial E_p(\mathbf{w})}{\partial w} + \alpha \Delta w(t-1), \tag{13}$$

where $0 < \eta < 1$, which is the learning rate, the momentum factor α is also a small positive number, and w represents any single weight in the network. In the above equation, $\Delta w(t)$ is the change in the weight computed at time t. The momentum term is sometimes used ($\alpha \neq 0$) to improve the smooth convergence of the algorithm. The algorithm defined by equation (13) is often termed as *instantaneous backpropagation* because it computes the gradient based on a single training vector. Another variation is *batch backpropagation* which computes the weight update using the gradient based on the total error E_T.

To implement this algorithm we must give an expression for the partial derivative of E_p with respect to each weight in the network. For an arbitrary weight in layer k this can be written using the Chain Rule:

$$\frac{\partial E_p(\mathbf{w})}{\partial w_{k,j,j}} = \frac{\partial E_p(\mathbf{w})}{\partial z_{k,j}(\mathbf{x}_p)} \frac{\partial z_{k,j}(\mathbf{x}_p)}{\partial w_{k,j,i}}. \tag{14}$$

Because the derivative of the activation function follows equation 11, we get:

$$\frac{\partial z_{k,j}(\mathbf{x}_p)}{\partial w_{k,j,i}} = z_{k,j}(\mathbf{x}_p)(1 - z_{k,j}(\mathbf{x}_p)) z_{k-1,j}(\mathbf{x}_p), \tag{15}$$

and

$$\frac{\partial E_p(\mathbf{w})}{\partial z_{k,j}(\mathbf{x}_p)} = \sum_{m=1}^{N_{k+1}} \frac{\partial E_p(\mathbf{w})}{\partial z_{k+1,m}(\mathbf{x}_p)} z_{k+1,m}(\mathbf{x}_p)(1 - z_{k+1,m}(\mathbf{x}_p)) w_{k+1,m,j},$$

which at the output layer corresponds to the output error:

$$\frac{\partial E_p(\mathbf{w})}{\partial z_K(\mathbf{x}_p)} = z_L(\mathbf{x}_p) - y_p. \tag{16}$$

2.3 Radial basis function networks

Radial basis function networks (RBF) [11] are a type of ANN that use radial basis functions as activation functions. RBFs consist of a two layer neural network, where each hidden unit implements a radial activated function. The output units compute a weighted sum of hidden unit outputs. Training consists of the unsupervised training of the hidden units followed by the supervised training of the output units weights. RBFs have their origin in the solution of a multivariate interpolation problem [9]. Arbitrary function $g(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ can be approximated by a map defined by a RBF network with a single hidden layer of K units:

$$\hat{g}_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{j=1}^{K} w_j \phi(\sigma_j, \|\mathbf{x} - \mathbf{c}_j\|), \tag{17}$$

where $\boldsymbol{\theta}$ is the vector of RBF parameters including $w_j, \sigma_j \in \mathbb{R}$, and $\mathbf{c}_j \in \mathbb{R}^n$; let us denote $\mathbf{w} = (w_1, w_2, \ldots, w_p)^T$, then the vector of RBF parameters can be expressed as $\boldsymbol{\theta}^T = \left(\mathbf{w}^T, \sigma_1, \mathbf{c}_1^T, \ldots, \sigma_K, \mathbf{c}_K^T\right)$. Each RBF is defined by its center $\mathbf{c}_j \in \mathbb{R}^n$ and width $\sigma_j \in \mathbb{R}$, and the contribution of each RBF to the network output is weighted by w_j. The RBF function $\phi()$ is a nonlinear function that monotonically decreases as $\mathbf{x}$ moves away from its center $\mathbf{c}_j$. The most common RBF used is the isotropic Gaussian:

$$\hat{g}_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{j=1}^{p} w_j \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_j\|^2}{2\sigma_j^2}\right).$$

The network can be thought as the composition of two functions $\hat{g}_{\boldsymbol{\theta}}(\mathbf{x}) = W \circ \Phi(\mathbf{x})$, the first one implemented by the RBF units $\Phi : \mathbb{R}^n \to \mathbb{R}^K$ performs a data space transformation which can be a dimensionality reduction or not, depending on whether $K > n$. The second function corresponds to a single layer linear Perceptron $W : \mathbb{R}^K \to \mathbb{R}$ giving the map of the RBF transformed data into the class labels. Training is accordingly decomposed into two phases. First a clustering algorithm is used to estimate the Gaussian RBF parameters (centers and variances). Afterwards, linear supervised training is used to estimate the weights from the hidden RBF to the output. In order to obtain a binary class label output, a hard limiter function is applied to the continuous output of the RBF network.

2.4 Probabilistic neural networks

A probabilistic neural network (PNN) [33] uses a kernel-based approximation to form an estimate of the probability density function of categories in a classification problem. In fact, it is a generalization of the Parzen windows distribution estimation, and a filtered version of the 1-NN classifier. The distance of the input feature vector $\mathbf{x}$ to the stored patterns is filtered by a RBF function. Let us denote the data sample partition as $X = X_1 \cup X_{-1}$, where $X_1 = \left\{\mathbf{x}_1^1, \ldots, \mathbf{x}_{n_1}^1\right\}$ and $X_1 = \left\{\mathbf{x}_1^{-1}, \ldots, \mathbf{x}_{n_{-1}}^{-1}\right\}$. That is, superscripts denote the class of the feature vector and $n_1 + n_{-1} = n$. Each pattern $\mathbf{x}_j^i$ of training data sample is interpreted as the weight of the j-th neuron of the i-th class. Therefore, the response of the neuron

is computed as the probability of the input feature vector according to a Normal distribution centered at the stored pattern:

$$\Phi_{i,j}(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}\sigma^n} \exp\left[-\frac{\left\|\mathbf{x}-\mathbf{x}_j^i\right\|}{2\sigma^2}\right]. \tag{18}$$

Therefore, the output of the neuron is inside $[0, 1]$. The tuning of a PNN network depends on selecting the optimal sigma value of the spread σ of the RBF functions which can be different for each class. In this paper, an exhaustive search for the optimal spread value in the range (0, 1) for each training set has been carried out. The output of the PNN is an estimation of the likelihood of the input pattern $\mathbf{x}$ being from class $i \in \{-1, 1\}$ by averaging the output of all neurons that belong to the same class:

$$p_i(x) = \frac{1}{n_i}\sum_{j=1}^{n_i}\Phi_{i,j}(\mathbf{x}). \tag{19}$$

The decision rule based on the output of all the output layer neurons is simply:

$$\hat{y}(\mathbf{x}) = \arg\max_i \{p_i(\mathbf{x})\}, \quad i \in \{-1, 1\}, \tag{20}$$

where $\hat{y}(\mathbf{x})$ denotes the estimated class of the pattern $\mathbf{x}$. If the a priori probabilities for each class are the same, and the losses associated with making an incorrect decision for each class are the same, the decision layer unit classifies the pattern $\mathbf{x}$ in accordance with the optimal Bayes' rule.

2.5 Learning vector quantization neural network

Learning vector quantization (LVQ), as introduced by Kohonen [19], represents every class $c \in \{-1, 1\}$ by a set $W(c) = \{\mathbf{w}_i \in \mathbb{R}^n; i = 1, \ldots, N_c\}$ of weight vectors (prototypes) which tessellate the input feature space. Let us denote W the union of all prototypes, regardless of class. If we denote c_i the class the weight vector $\mathbf{w}_i \in W$ is associated with, the decision rule that classifies a feature vector $\mathbf{x}$ is as follows:

$$c(\mathbf{x}) = c_{i^*},$$

where

$$i^* = \arg\min_i \{\|\mathbf{x}-\mathbf{w}_i\|\}.$$

The training algorithm of LVQ aims at minimizing the classification error on the given training set, i.e., $E = \sum_j (y_j - c(\mathbf{x}_j))^2$, modifying the weight vectors on the presentation of input feature vectors. The heuristic weight updating rule is as follows:

$$\Delta\mathbf{w}_{i^*} = \begin{cases} \epsilon(\mathbf{x}_j - \mathbf{w}_{i^*}) & \text{if } c_{i^*} = y_j \\ -\epsilon(\mathbf{x}_j - \mathbf{w}_{i^*}) & \text{otherwise} \end{cases}, \tag{21}$$

that is, the input's closest weight is adapted either toward the input if their classes match, or away from it if not. This rule is highly unstable, therefore, the practical

approach consists in performing an initial clustering of each class data samples to obtain an initial weight configuration using equation 21 to perform the fine tuning of the classification boundaries. This equation corresponds to a LVQ1 approach. The LVQ2 approach involves determining the two input vector's closest weights. They are moved toward or away from the input according to the matching of their classes.

3. Materials and Methods

Structural MRI and DTI data from twenty men (aged 21-55 yr), ten patients and ten controls, from a publicly available database from the National Alliance for Medical Image Computing (NAMIC)[1] were the subjects of this study in this experiment. The imaging parameters and demographic information about the subjects can be obtained from the web site, we omit them for lack of space. A technical description of the feature extraction method and the data will be available[2], because many of the difficulties found have no place in an academic paper, but are important for the reproducibility of the results.

3.1 Scalar features of diffusion tensors

In DTI, a diffusion tensor at a voxel is a 3×3 positive-definite symmetric matrix D which can be represented by its decomposition as $D = \lambda_1 \mathbf{g}_1 \mathbf{g}_1^T + \lambda_2 \mathbf{g}_2 \mathbf{g}_2^T + \lambda_3 \mathbf{g}_3 \mathbf{g}_3^T$, where $\lambda_1 \geq \lambda_2 \geq \lambda_3$ and $\mathbf{g}_1$, $\mathbf{g}_2$, $\mathbf{g}_3$ are the eigenvalues and eigenvectors of D, respectively. Two scalar measures were extracted [7] from the voxels diffusion tensors: the mean diffusivity (MD) and the fractional anisotropy (FA). The first corresponds to the average eigenvalue:

$$MD = \frac{\mathrm{Tr}\,(D)}{3} = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3}. \tag{22}$$

The FA measures the fraction of the magnitude of D that can be related to anisotropic diffusion in a mean-squared sense (i.e. the extent of deviation from isotropic diffusivity in all direction). Its magnitude is also rotationally invariant, and independent from sorting of the eigenvalues. The FA is calculated as follows:

$$FA = \sqrt{\frac{1}{2}} \frac{\sqrt{(\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_3 - \lambda_1)^2}}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}. \tag{23}$$

Thus, isotropic diffusion is imaged as zero value and FA maximum value is one. Fig. 1 show slices of FA and MD volumes of one study subject.

3.2 Image preprocessing

Feature extraction requires that the diffusion related data is spatially normalized, in order to compute the correlation measure and to extract the values of the feature vectors. Our starting point was the nonlinear registration [4] of the T1-weighted

[1] http://www.insight-journal.org/midas/collection/view/190?path_navigation=17
[2] http://www.ehu.es/ccwintco/index.php/GIC-experimental-databases

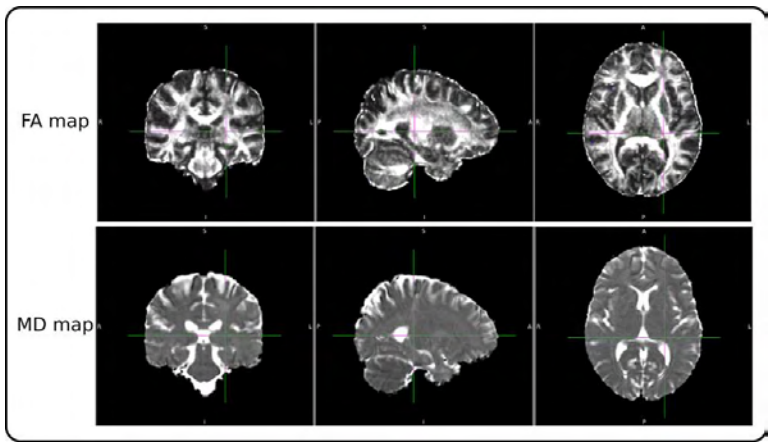


Fig. 1 *FA and MD maps of one subject.*

sMRI skull stripped volumes of each subject to the Montreal Neurological Institute (MNI152) standard template, using the ANTS[3] nonlinear elastic registration algorithm. For the elastic registration, a probabilistic correlation similarity metric was chosen with window radius 4 and gradient step length 1. The optimization has been performed over three resolutions with a maximum of 100 iterations at the coarsest level, 100 at the next coarsest and 10 at the full resolution. The optimization stops when either the distance between both images cannot be further minimized or the maximum number of iterations is reached. We used a Gaussian regularization with sigma parameter value 3 which operates only on the deformation field and not on the similarity gradient. In addition, a previous histogram matching step has been performed. The deformation fields of this registration were used afterwards for the spatial deformation of the FA and MD volumes.

The DWI scans were already noise filtered and corrected for eddy currents and head motion by the group that originally acquired the scans. A brain mask was obtained for each DWI data volume to calculate the FA and MD maps of each subject [7]. The FA and MD maps were linearly registered to the sMRI skull stripped volumes [32] of each subject and then non-linearly registered to MNI applying the deformation fields obtained from the sMRI data nonlinear registration. All of the FA and MD volumes were then considered spatially normalized.

3.3 Feature extraction

Once the FA and MD maps were spatially normalized, we processed them independently. We considered each voxel site independently, forming a vector at the voxel site across all the subjects. Then, we computed the Pearson correlation coefficient between this vector and the control variable with the labels (patient=1, control=−1). Thus we obtained for FA and MD data two independent volumes containing correlation values at each voxel. For each volume we estimated the em-

[3] `http://www.picsl.upenn.edu/ANTS`http://www.picsl.upenn.edu/ANTS

pirical distribution of the absolute correlation values and determined a selection threshold corresponding to a percentile of this absolute correlation distribution. Voxel sites with absolute value of the correlation above this threshold were retained, and the feature vector for each subject was composed of the FA or MD values at these voxel sites. In Tab. I, we show the percentiles and the number of voxels selected for each feature vector.

Database	Percentile	DT Measure	Number of voxels
A	99.990%	FA	241
		MD	241
B	99.992%	FA	193
		MD	193
C	99.995%	FA	121
		MD	121
D	99.997%	FA	72
		MD	72
E	99.999%	FA	24
		MD	24

Tab. I *Databases considered, percentile on the correlation distribution and size of the feature vectors.*

Although the voxel sites selected to build the feature vectors (the feature mask) were localized in many different regions of the subject brains, we found that most were concentrated in regions of characteristic abnormalities found for schizophrenia shown in the literature (see [20] for references). The features voxel locations[4] were different for FA and MD maps. In the case of FA, the selected voxels were localized mainly in parietal and temporal lobes, but also in the cerebellum and occipital lobe. More specifically, in WM we found discriminant voxel values in the cingulum bundle, superior and inferior longitudinal fasciculus and in the inferior fronto-occipital fasciculus. On the other hand, in the MD maps, the most discriminant voxel values were the ones localized in frontal and parietal lobes, more specifically the cingulum bundle, inferior fronto-occipital and longitudinal fasciculus, and superior longitudinal fasciculus.

3.4 Classifiers parameters

All classifiers were calculated with a maximum iteration number (epochs) of 100. For the 1-NN classifier, we used the nearest neighbor rule with Euclidean distance. In the SVM algorithm, a linear kernel function was used as well as a sequential minimal optimization for the separating hyperplane method. For BPNN, the number of neurons in the hidden layer was 4, the learning rate was set to 0.05, tan-sigmoid transfer function, and training and learning functions were gradient descent with

[4]This specification of the voxel locations was obtained with the "atlasquery" tool from FMRIB's FSL (`http://www.fmrib.ox.ac.uk/fsl/`http://www.fmrib.ox.ac.uk/fsl/) using the "MNI Structural Atlas" and the "JHU White-Matter Tractography Atlas".

momentum. LVQ2 was trained with 2 hidden neurons, learning rate set to 0.01. The training function used for RBF was according to resilient backpropagation algorithm. In the case of PNN, random order incremental training was used. For the last three algorithms (BPNN, LVQ2 and RBF) zeros were set as initial input and layer delay conditions. These parameters have been selected after a sensitivity analysis.

We tested several cross-validation strategies, because the small database size may have an influence on the results obtained with each of these cross-validation processes. Cross-validation partitions were computed 40 times, and we show average accuracy, sensitivity, and specificity for the 10-fold cross-validation procedure.

Database		FA	MD
A	1-NN	1.00 (1.00-1.00)	1.00 (1.00-1.00)
	SVM	1.00 (1.00-1.00)	1.00 (1.00-1.00)
	BP	0.75 (0.67-1.00)	0.78 (0.69-1.00)
	RBF	0.98 (0.97-1.00)	1.00 (1.00-1.00)
	PNN	1.00 (1.00-1.00)	0.54 (0.54-0.54)
	LVQ2	1.00 (1.00-1.00)	1.00 (1.00-1.00)
B	1-NN	1.00 (1.00-1.00)	1.00 (1.00-1.00)
	SVM	1.00 (1.00-1.00)	1.00 (1.00-1.00)
	BP	0.75 (0.66-1.00)	0.78 (0.70-1.00)
	RBF	1.00 (1.00-1.00)	1.00 (1.00-1.00)
	PNN	1.00 (1.00-1.00)	0.52 (0.52-0.52)
	LVQ2	1.00 (1.00-1.00)	1.00 (1.00-1.00)
C	1-NN	1.00 (1.00-1.00)	1.00 (1.00-1.00)
	SVM	1.00 (1.00-1.00)	1.00 (1.00-1.00)
	BP	0.77 (0.68-1.00)	0.77 (0.68-1.00)
	RBF	1.00 (1.00-1.00)	1.00 (1.00-1.00)
	PNN	1.00 (1.00-1.00)	0.52 (0.52-0.52)
	LVQ2	1.00 (1.00-1.00)	1.00 (1.00-1.00)
D	1-NN	1.00 (1.00-1.00)	1.00 (1.00-1.00)
	SVM	1.00 (1.00-1.00)	1.00 (1.00-1.00)
	BP	0.77 (0.68-1.00)	0.77 (0.68-1.00)
	RBF	1.00 (1.00-1.00)	0.84 (0.79-0.90)
	PNN	0.99 (0.99-1.00)	0.55 (0.55-0.55)
	LVQ2	1.00 (1.00-1.00)	1.00 (1.00-1.00)
E	1-NN	0.94 (0.90-0.99)	1.00 (1.00-1.00)
	SVM	0.95 (0.90-1.00)	1.00 (1.00-1.00)
	BP	0.76 (0.67-1.00)	0.77 (0.68-1.00)
	RBF	0.92 (0.90-0.94)	0.89 (0.91-0.88)
	PNN	0.94 (0.90-0.99)	0.52 (0.52-0.52)
	LVQ2	0.97 (0.94-1.00)	1.00 (1.00-1.00)

Tab. II *10-fold cross-validation results. Accuracy (Sensitivity, Specificity).*

4. Results

The results are presented in Tab. II. The most striking result is that we found optimal performance of almost all classifiers built from the provided feature vectors. The only exceptions were the results of PNN on MD data; tuning of the Gaussian kernel variance was more difficult than applying the training algorithm of other approaches. Also BP shows lower performance than the others. The second general result is that MD features seem to perform slightly better than FA features, disregarding the anomaly of PNN classifiers. In the experimental design, we wanted to test if decreasing the size of the feature vectors had an impact on the classifiers performance. We found that performance was not affected down to the smallest feature vector (database E) where decreases in performance can be appreciated in all the classifiers for the FA data, while 1-NN, SVM and LVQ2 maintain their performance for MD data.

5. Conclusion

The goal of this paper was to test the hypothesis that classification algorithms constructed using statistical and Neural Network approaches can discriminate between schizophrenia patients and control subjects on the basis of features extracted from DTI data. The way to build the feature vectors has been the direct selection of voxels from the DTI-derived FA and MD scalar valued volumes that show a high correlation with the control variable that labels the subjects. The selected voxels roughly correspond to findings reported in the medical literature. Surprisingly, all the classifiers obtain near perfect results. Despite the simplicity of our feature extraction process, the results compare well with other results found in the literature [10, 36]. We think that appropriate pre-processing of the data is of paramount importance and cannot be disregarded, trusting that ensuing statistical or machine learning processes may cope with the errors introduced by lack of appropriate data normalization. Therefore, our main conclusion is that the proposed feature extraction is very effective in providing a good discrimination between schizophrenia patients that can easily be exploited by the classifier construction algorithms. The main limitation of this study is that the results come from a small database. Therefore, more extensive testing will be needed to confirm our conclusions. Nevertheless, we are making available[5] the actual data employed in the computational experiments to allow for independent validation of our results.

Acknowledgements

Thanks to the National Alliance for Medical Image Computing and the Brigham & Women's Hospital for making the database used for this study publicly available.

[5] http://www.ehu.es/ccwintco/index.php/GIC-experimental-databases

References

[1] Albizuri F. X., d'Anjou A., Graña M., Torrealdea F. J., Hernandez M. C.: The high order boltzmann machine: learned distribution and topology. IEEE Transactions. on Neural Networks, **6**, 3 1995, pp. 767–770.

[2] Andreone N., Tansella M., Cerini R., Versace A., Rambaldelli G., Perlini C., Dusi N., Pelizza L., Balestrieri M., Barbui C., Nosé M., Gasparini A., Brambilla P.: Cortical white-matter microstructure in schizophrenia. diffusion imaging study. The British Journal of Psychiatry: The Journal of Mental Science, 191, Aug. 2007, pp. 113–119. PMID: 17666494.

[3] American Psychiatric Association. DSM-IV-TR: Diagnostic and Statistical Manual of Mental Disorders (Diagnostic & Statistical Manual of Mental Disorders. American Psychiatric Press Inc., 4th text revision edition, July 2000.

[4] Avants B., Epstein C., Grossman M., Gee J.: Symmetric diffeomorphic image registration with Cross-Correlation: evaluating automated labeling of elderly and neurodegenerative brain. Medical image analysis, **12**, 1, Feb. 2008, pp. 26–41. PMID: 17659998 PMCID: 2276735.

[5] Barta P., Pearlson G., Brill L., Royall R., McGilchrist I., Pulver A., Powers R., Casanova M., Tien A., Frangou S., Petty R.: Planum temporale asymmetry reversal in schizophrenia: replication and relationship to gray matter abnormalities. The American Journal of Psychiatry, **154**, 5, May 1997, pp. 661–667. PMID: 9137122.

[6] Basser P., Mattiello J., LeBihan D.: MR diffusion tensor spectroscopy and imaging. Biophysical Journal, **66**, 1, Jan. 1994, pp. 259–267. PMID: 8130344 PMCID: 1275686.

[7] Behrens T., Woolrich M., Jenkinson M., Johansen-Berg H., Nunes R., Clare S., Matthews P., Brady J., Smith S.: Characterization and propagation of uncertainty in diffusion-weighted MR imaging. Magnetic Resonance in Medicine, **50**, 5, 2003, pp. 1077–1088.

[8] Breier A., Buchanan R., Elkashef A., Munson R., Kirkpatrick, B., Gellad F.: Brain morphology and schizophrenia: A magnetic resonance imaging study of limbic, prefrontal cortex, and caudate structures. Archives of General Psychiatry, 49, 12, Dec. 1992, pp. 921–926.

[9] Broomhead D., Lowe D.: Multivariable functional interpolation and adaptive networks. Complex Systems, 2, 1988, pp. 321–355.

[10] Caprihan A., Pearlson G., Calhoun V.: Application of principal component analysis to distinguish patients with schizophrenia from healthy controls based on fractional anisotropy measurements. NeuroImage, **42**, 2, Aug. 2008, pp. 675–682.

[11] Chen S., Cowan C., Grant P.: Orthogonal least squares learning algorithm for radial basis function networks. IEEE Transactions on Neural Networks, **2**, 2, 1991, pp. 302–309.

[12] Fan Y., Shen D., Gur R., Gur R., Davatzikos C.: COMPARE: classification of morphological patterns using adaptive regional elements. IEEE Transactions on Medical Imaging, **26**, 1, Jan. 2007, pp. 93–105. PMID: 17243588.

[13] Frederikse M., Lu A., Aylward E., Barta P., Sharma T., Pearlson G.: Sex differences in inferior parietal lobule volume in schizophrenia. The American Journal of Psychiatry, **157**, 3, Mar. 2000, pp. 422–427.

[14] García-Sebastián M., Savio A., Graña M., Villanúa J.: On the use of morphometry based features for Alzheimer's disease detection on MRI. In: Proceedings of the 10th International Work-Conference on Artificial Neural Networks: Part I: Bio-Inspired Systems: Computational and Ambient Intelligence, Salamanca, Spain, 2009, Springer-Verlag, pp. 957–964.

[15] Goldstein J., Goodman J., Seidman L., Kennedy D., Makris N., Lee H., Tourville J., Caviness V., Faraone S., Tsuang M.: Cortical abnormalities in schizophrenia identified by structural magnetic resonance imaging. Archives of General Psychiatry, **56**, 6, June 1999, pp. 537–547.

[16] Gur R., Cowell P., Latshaw A., Turetsky B., Grossman R., Arnold S., Bilker W., Gur R.: Reduced dorsal and orbital prefrontal gray matter volumes in schizophrenia. Archives of General Psychiatry, **57**, 8, Aug. 2000, pp. 761–768.

[17] Haykin S.: Neural Networks: A Comprehensive Foundation, 2nd ed. Prentice Hall, July 1998.

[18] Holinger D., Shenton M., Wible C., Donnino R., Kikinis R., Jolesz F., McCarley R.: Superior temporal gyrus volume abnormalities and thought disorder in left-handed schizophrenic men. The American Journal of Psychiatry, **156**, 11, Nov. 1999, pp. 1730–1735.

[19] Kohonen T.: Learning vector quantization. In: The handbook of brain theory and neural networks. MIT Press, 1998, pp. 537–540.

[20] Kubicki M., McCarley R., Westin C., Park H., Maier S., Kikinis R., Jolesz F., Shenton M.: A review of diffusion tensor imaging studies in schizophrenia. Journal of Psychiatric Research, **41**, 1-2, 2007, pp. 15–30. PMID: 16023676 PMCID: 2768134.

[21] Kubicki M., Park H., Westin C., Nestor P., Mulkern R., Maier S., Niznikiewicz M., Connor E., Levitt J., Frumin M., Kikinis R., Jolesz F., McCarley R., Shenton M.: DTI and MTR abnormalities in schizophrenia: Analysis of white matter integrity. NeuroImage, **26**, 4, July 2005, pp. 1109–1118. PMID: 15878290 PMCID: 2768051.

[22] Kubicki M., Westin C., McCarley R., Shenton M.: The application of DTI to investigate white matter abnormalities in schizophrenia. Annals of the New York Academy of Sciences, **1064**, 1, 2005, pp. 134–148.

[23] Kyriakopoulos M., Bargiotas T., Barker G., Frangou S.: Diffusion tensor imaging in schizophrenia. European Psychiatry, **23**, 4, June 2008, pp. 255–273.

[24] McGuire P., Frith C.: Disordered functional connectivity in schizophrenia. Psychological Medicine, **26**, 4, July 1996, pp. 663–667. PMID: 8817700.

[25] Graña M., D'Anjou A., Albizuri F. X., Hernandez M., Torrealdea F. J., Gonzalez A. I.: Experiments of fast learning with High Order Boltzmann Machines. Applied Intelligence, **7**, 4, 1997, pp. 287–303.

[26] Niznikiewicz M., Donnino R., McCarley R., Nestor P., Iosifescu D., O'Donnell B., Levitt J., Shenton M.: Abnormal angular gyrus asymmetry in schizophrenia. The American Journal of Psychiatry, **157**, 3, Mar. 2000, pp. 428–437.

[27] Pierpaoli C., Jezzard P., Basser P., Barnett A., Chiro G. D.: Diffusion tensor MR imaging of the human brain. Radiology, **201**, 3, Dec. 1996, pp. 637–648. PMID: 8939209.

[28] Rumelhart D., Hinton G., Williams R.: Learning internal representations by error propagation. MIT Press, 1986, pp. 318–362.

[29] Schlaepfer T., Harris G., Tien A., Peng L., Lee S., Federman E., Chase G., Barta P., Pearlson G.: Decreased regional cortical gray matter volume in schizophrenia. The American Journal of Psychiatry, **151**, 6, June 1994, pp. 842–848.

[30] Shenton M., Dickey C., Frumin M., McCarley R.: A review of MRI findings in schizophrenia. Schizophrenia Research, **49**, 1-2, Apr. 2001, pp. 1–52. PMID: 11343862 PMCID: 2812015.

[31] Shenton M., Kikinis R., Jolesz F., Pollak S., LeMay M., Wible C., Hokama H., Martin J., Metcalf D., Coleman M., McCarley R.: Abnormalities of the left temporal lobe and thought disorder in schizophrenia. New England Journal of Medicine, **327**, 9, 1992, pp. 604–612.

[32] Smith S., Jenkinson M., Woolrich M., Beckmann C., Behrens T., Johansen-Berg H., Bannister P., Luca M. D., Drobnjak I., Flitney D., Niazy R., Saunders J., Vickers J., Zhang Y., Stefano N. D., Brady J., Matthews P.: Advances in functional and structural MR image analysis and implementation as FSL. NeuroImage, **23**, Supplement 1, 2004, pp. S208–S219.

[33] Specht D.: Probabilistic neural networks. Neural Networks, **3**, 1, 1990, pp. 109–118.

[34] Szeszko P., Bilder R., Lencz T., Pollack S., Alvir J., Ashtari M., Wu H., Lieberman J.: Investigation of frontal lobe subregions in first-episode schizophrenia. Psychiatry Research, **90**, 1, Feb. 1999, pp. 1–15. PMID: 10320207.

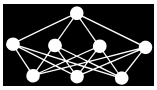
[35] Vapnik V.: Statistical Learning Theory. Wiley-Interscience, Sept. 1998.

[36] Wang P., Verma R.: On classifying Disease-Induced patterns in the brain using diffusion tensor images. In: Medical Image Computing and Computer-Assisted Intervention - MICCAI 2008, 2008, pp. 908–916.

[37] Wernicke C.: Grundriss der Psychiatrie in klinischen Vorlesungen/von Carl Wernicke. VDM Verlag Dr. Mller, Saarbrcken, 2007.

[38] Wible C., Shenton M., Hokama H., Kikinis R., Jolesz F., Metcalf D., McCarley R.: Prefrontal cortex and schizophrenia: A quantitative magnetic resonance imaging study. Archives of General Psychiatry, **52**, 4, Apr. 1995, pp. 279–288.

[39] Yoon U., Lee J., Im K., Shin Y., Cho B., Kim I., Kwon J., Kim S.: Pattern classification using principal components of cortical thickness and its discriminative pattern in schizophrenia. NeuroImage, **34**, 4, Feb. 2007, pp. 1405–1415.



THE NEW UPPER BOUND ON THE PROBABILITY OF ERROR IN A BINARY TREE CLASSIFIER WITH FUZZY INFORMATION

*Robert Burduk**

Abstract: The paper considers the mixture of randomness and fuzziness in a binary tree classifier. This model of classification is based on fuzzy observations, the randomness of classes and the Bayes rule. In this work, we present a new upper bound on the probability of error in a binary tree classifier. The obtained error for fuzzy observations is compared with the case when observations are not fuzzy, as a difference of errors. Additionally, the obtained results are compared with the bound on the probability of error based on information energy of fuzzy events. For interior nodes of decision tree, the new bound is twice as precise as the bound based on information energy.

Key words: *Binary tree classifier, probability of error, fuzzy observations, Bayes rule*

Received: 20th September 2010
Revised and accepted: 13th November 2010

1. Introduction

Many papers have previously covered the aspect of fuzzy and imprecise information in pattern recognition [3], [4], [10], [11], [12]. In the real-world recognition and classification problems we are faced with imprecise information that is connected to diverse facets of human thinking. The origins of randomness and fuzziness sources are related to labels expressed in feature space as well as to labels of classes taken into account in classification procedures. There are many cases where the available information is a mixture of randomness and fuzziness. In [7] the pattern recognition problem with fuzzy classes and fuzzy information is formulated. This paper considers the following three situations:

- fuzzy classes and exact information,

*Robert Burduk
Department of Systems and Computer Networks, Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50–370 Wroclaw, Poland, E-mail: robert.burduk@pwr.wroc.pl

- exact classes and fuzzy information,
- fuzzy classes and fuzzy information.

In this paper, we consider the classification error problem for exact classes and fuzzy information of object features. The classification error is the ultimate measure of the classifier performance. Competing classifiers can also be evaluated based on their error probabilities. Several studies have previously described the Bayes probability of error for a single-stage classifier [1], [2], [13] and for a hierarchical classifier [5], [6]. We consider the problem of classification for the case in which observations of the features are represented by the fuzzy sets. Additionally, the a priori probabilities of classes and class-conditional probability density functions are random. For such assumptions we consider the binary tree classifier. The obtained error for fuzzy observations is compared with the case where observations are not fuzzy. The difference of errors for these two cases is the subject of this paper. Additionally, the obtained results are compared with the bound on the probability of error based on the information energy of fuzzy events.

The contents of the work are as follows: Section 2 introduces the necessary background and describes the Bayes hierarchical classifier. In Section 3, the introduction to fuzzy sets is presented. In Section 4, we present the difference between the probability of misclassification of the fuzzy and crisp data in the binary tree classifier. Section 5 contains a numerical example that shows the error for this classifier. Section 6 concludes the work.

2. Bayes Hierarchical Classifier

In the paper [6], the Bayesian hierarchical classifier is presented. The synthesis of a multistage classifier is a complex problem. It involves specification of the following components:

- the decision logic, i.e. hierarchical ordering of classes,
- the feature used at each stage of decision,
- the decision rules (strategy) for performing the classification.

This paper focuses on the last problem. This means that we will only consider the presentation of decision algorithms, assuming that both the tree structure and the feature used at each non-terminal node have been previously specified.

The procedure in the Bayesian hierarchical classifier consists of the following sequences of operations, as presented in Fig. 1. At the first stage, some specific features x_0 are measured. They are chosen from all accessible features x, which describe the pattern that will be classified. These data constitute the basis for making a decision i_1. This decision, being the result of the recognition at the first stage, defines a certain subset in the set of all classes and simultaneously indicates features x_{i_1} (from x) which should be measured in order to make a decision at the next stage.

Now, at the second stage, features x_{i_1} are measured, which together with i_1 constitute a basis for making the next decision i_2. This decision – like i_1 – indicates

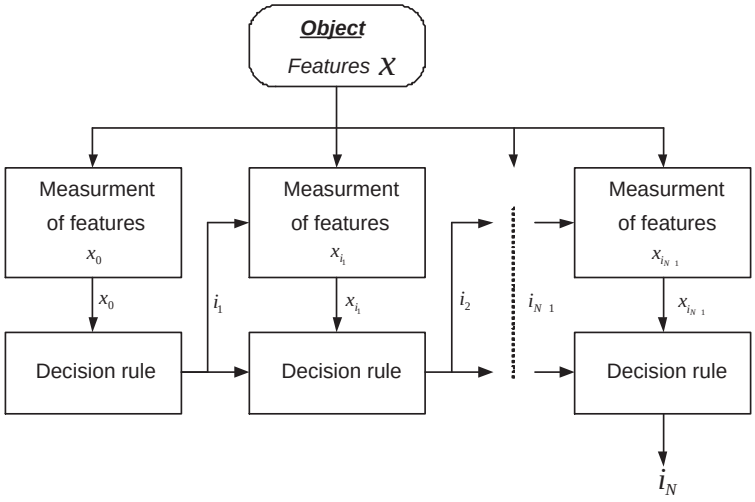


Fig. 1 *Block diagram of the hierarchical classifier.*

features x_{i_2} that are necessary to make the next decision (at the third stage, as in the previous stage) that in turn defines a certain subset of classes, not in the set of all classes, but in the subset indicated by the decision i_2, and so on. The whole procedure ends at the N-th stage, where the decision made i_N indicates a single class, which is the final result of this multistage recognition.

2.1 Decision problem statement

Let us consider a pattern recognition problem, in which the number of classes equals M. Let us assume that classes are organized in a $(N+1)$ horizontal decision tree. Let us number all the nodes of the decision tree constructed with consecutive numbers of $0, 1, 2, \ldots$, reserving 0 for the root-node, and let us assign numbers of classes from the $\mathcal{M} = \{1, 2, \ldots, M\}$ set to terminal nodes so that each of them can be labeled with the class number connected to that node. This allows us to introduce the following notation:

- $\mathcal{M}(n)$ – the set of nodes, whose distance from the root is n, $n = 0, 1, 2, \ldots, N$. In particular $\mathcal{M}(0) = \{0\}$, $\mathcal{M}(N) = \mathcal{M}$,
- $\overline{\mathcal{M}} = \bigcup_{n=0}^{N-1} \mathcal{M}(n)$ – the set of internal nodes (non terminal),
- $\mathcal{M}_i \subseteq \mathcal{M}(N)$ – the set of class labels attainable from the i-th node ($i \in \overline{\mathcal{M}}$),
- $\mathcal{M}^i$ – the set of nodes of immediate descendant node i ($i \in \overline{\mathcal{M}}$),
- m_i – the node of direct predecessor of the i-th node ($i \neq 0$),
- $s(i)$ – the set of nodes on the path from the root-node to the i-th node, $i \neq 0$.

We will continue to adopt the probabilistic model of the recognition problem, i.e. we will assume that the class label of the pattern being recognised as $j_N \in \mathcal{M}(N)$ and its observed features x are the realizations of a couple of random variables $\boldsymbol{J}_N$ and $\boldsymbol{X}$. The complete probabilistic information denotes the knowledge of a priori probabilities of classes:

$$p(j_N) = P(J_N = j_N), \quad j_N \in \mathcal{M}(N) \tag{1}$$

and class-conditional probability density functions:

$$f_{j_N}(x) = f(x/j_N), \quad x \in X, \quad j_N \in \mathcal{M}(N)\ . \tag{2}$$

Let

$$x_i \in X_i \subseteq R^{d_i}, \quad d_i \leq d, \quad i \in \mathcal{M} \tag{3}$$

denote the vector of features used at the i-th node, which have been selected from the vector x.
Our aim now is to calculate the so-called multistage recognition strategy $\pi_N = \{\Psi_i\}_{i \in \overline{\mathcal{M}}}$, which is the set of recognition algorithms in the form:

$$\Psi_i : X_i \to \mathcal{M}^i, \quad i \in \overline{\mathcal{M}}\ . \tag{4}$$

Formula (4) is a decision rule (recognition algorithm) used at the i-th node that maps observation subspace to the set of immediate descendant nodes of the i-th node. Analogically, the decision rule (4) partitions observation subspace X_i into disjoint decision regions $D^k_{x_i},\ k \in \mathcal{M}^i$, so that observation x_i is allocated to the node k if $k_i \in D^k_{x_i}$, namely:

$$D^k_{x_i} = \{x_i \in X_i : \Psi_i(x_i) = k\},\ k \in \mathcal{M}^i, \quad i \in \overline{\mathcal{M}}. \tag{5}$$

Our aim is to minimise the expected risk function (expected loss function $L(I_N, J_N)$) denoted by:

$$R^*(\pi_N) = \min_{\pi_N} R(\pi_N) = \min_{\pi_N} E[L(I_N, J_N)], \tag{6}$$

where π_N is the strategy of the decision tree classifier. The π_N is the set of classifying rules used at a particular node $\pi_N = \{\Psi_i\}_{i \in \overline{\mathcal{M}}}$.

Globally optimal strategy π^*_N. This strategy minimises the mean probability of misclassification throughout the whole multistage recognition process and leads to an optimal global decision strategy, whose recognition algorithm at the n-th stage is as follows:

$$\Psi^*_{i_n}(x_{i_n}) = i_{n+1} \quad \text{if} \tag{7}$$

$$i_{n+1} = \arg \max_{k \in \mathcal{M}^{i_n}} Pc(k)p(k)f_k(x_{i_n}),$$

where $Pc(k)$ is the empirical probability of correct classification at the next stages if at the n-th stage decision i_{n+1} is made.

3. Basic Notions of Fuzzy Sets Theory

The concept of a fuzzy set was introduced in 1966 [14] as an extension of the classical notion of a set. For any classical set, it is possible to define a characteristic function. This function takes either the values 0 or 1. For a fuzzy set, the characteristic function can take any value between 0 and 1. A fuzzy set A is defined by the set of tuples $A = (x, \mu_A(x)|x \in X)$, where $\mu_A(x)$ is a membership function of the fuzzy set and may be continuous, or the set contains only discrete elements assessed by membership values.

Fuzzy number A is a fuzzy set defined on the set of real numbers $\mathbb{R}$ characterized by means of a membership function $\mu_A(x)$, $\mu_A : \mathbb{R} \to [0, 1]$. In this study, special kinds of fuzzy numbers including triangular fuzzy numbers are employed. Triangular fuzzy numbers can be defined by a triplet $A = (a_1, a_2, a_3)$.

Fuzzy information $\mathcal{A}_k \in \Re^d$, $k = 1, ..., d$ (d is the dimension of the feature vector) is a set of fuzzy events $\mathcal{A}_k = \{A_k^1, A_k^2, ..., A_k^{n_k}\}$ characterized by membership functions

$$\mathcal{A}_k = \{\mu_{A_k^1}(x_k), \mu_{A_k^2}(x_k), ..., \mu_{A_k^{n_k}}(x_k)\}. \tag{8}$$

The value of index n_k defines the possible number of fuzzy events for x_k (for the k-th dimension of feature vector). In addition, assume that for each observation subspace x_k the set of all available fuzzy observations (8) satisfies the orthogonality constraint [7]:

$$\sum_{l=1}^{n_k} \mu_{A_k^l}(x_k) = 1. \tag{9}$$

Let $(\Omega, \mathcal{F}, \mathbf{P})$ be the probability space, where $\mathcal{F}$ is the σ-field of Borel sets in R^n and $\mathbf{P}$ is a probability measure over R^n. Then, a fuzzy event in R^n is a fuzzy set A in R^n whose membership function $\mu_A(x)$ is Borel measurable. The probability of the fuzzy event is defined by the integral [15]:

$$P(A) = \int_{\Re^d} \mu_A(x) f(x) dx. \tag{10}$$

The probability $P(A)$ of a fuzzy event A defined by (10) represents a crisp number in the interval $[0, 1]$.

4. The Bound on the Probability of Error in a Binary Tree Classifier with Fuzzy Information

4.1 Exact difference between the probability of misclassification for the fuzzy and non fuzzy observations of features

The decision algorithms for the zero-one loss function in the case of the global optimal strategy of multistage recognition for non fuzzy observations of features are as follows [6]:

$$\Psi_{i_n}^*(x_{i_n}) = i_{n+1} \quad \text{if} \tag{11}$$

$$i_{n+1} = \arg \max_{k \in \mathcal{M}^{i_n}} \sum_{j_N \in \mathcal{M}_k} p(j_N) q^*(j_N/k, j_N) f_{j_N}(x_{i_n})$$

for $i_n \in \mathcal{M}(n)$, $n = 0, 1, 2, \ldots, N-1$, where $q^*(j_N/i_{n+1}, j_N)$ denotes the probability of accurate object classification of the class j_N at further stages using π_N^* strategy rules on condition that on the n-th stage the i_{n+1} decision has been made.

As a consequence of Bayes' theorem, the probability of error $Pe(\pi_N^*)$ for non fuzzy observations of features for globally optimal strategy π_N^* of multistage classifier is represented by [6]:

$$Pe(\pi_N^*) = 1 - \sum_{j_N \in \mathcal{M}(N)} p(j_N) \prod_{i \in s(j_N) - \{j_N\}} Pc(i), \tag{12}$$

where

$$Pc(i) = \int_{\Re^{\mathcal{M}^i}} \arg\max_{\mathcal{M}^i} P(\omega_{\mathcal{M}^i}) p(x_{\mathcal{M}^i} | \omega_{\mathcal{M}^i}) dx_{\mathcal{M}^i}.$$

For fuzzy observation of features, where the A_{i_n} denotes the fuzzy value of an object feature observed in i_n node, the decision algorithms for the zero-one loss function in the case of the global optimal strategy of multistage recognition is as follows [4]:

$$\Psi_{i_n}^*(A_{i_n}) = i_{n+1} \quad \text{if} \tag{13}$$

$$i_{n+1} = \arg \max_{k \in \mathcal{M}^{i_n}} \sum_{j_N \in \mathcal{M}_k} p(j_N) q^*(j_N/k, j_N) \int_{\Re^d} \mu_{A_{i_n}}(x_{i_n}) f_{j_N}(x_{i_n}) dx_{i_n}.$$

Similarly, if (9) holds the probability of error $Pe_F(\pi_N^*)$ for multistage classifier with fuzzy observations is as follows:

$$Pe_F(\pi_N^*) = 1 - \sum_{j_N \in \mathcal{M}(N)} p(j_N) \prod_{i \in s(j_N) - \{j_N\}} Pc_F(i), \tag{14}$$

where

$$Pc_F(i) = \sum_{A_i \in \mathcal{A}_i} \arg\max_{\mathcal{M}^i} \int_{\Re^{\mathcal{M}^i}} \mu_{A_i}(x_{\mathcal{M}^i}) P(\omega_{\mathcal{M}^i}) p(x_{\mathcal{M}^i} | \omega_{\mathcal{M}^i}) dx_{\mathcal{M}^i}.$$

When we use fuzzy information on object features instead of exact information, we deteriorate the classification accuracy. The difference between the probability of misclassification for the fuzzy $Pe_F(\pi_N^*)$ and the non fuzzy observation of features $Pe(\pi_N^*)$ for the globally optimal strategy of multistage recognition π_N^* is as follows:

$$Pe_F(\pi_N^*) - Pe(\pi_N^*) = \sum_{j_N \in \mathcal{M}(N)} p(j_N) \prod_{i \in s(j_N) - \{j_N\}} \varepsilon_i, \tag{15}$$

where

$$\varepsilon_i = \sum_{A_i \in \mathcal{A}_i} \int_{\Re^i} \mu_A(x_i) \max_{k \in \mathcal{M}^i} \{f_k(x_i)\} dx_i - \max_{k \in \mathcal{M}^i} \left\{ \int_{\Re^i} \mu_A(x_i) f_k(x_i) dx_i \right\}.$$

This is the exact difference between the probability of misclassification for the fuzzy $Pe_F(\pi_N^*)$ and non fuzzy observations of features $Pe(\pi_N^*)$. This result is received for full probabilistic information.

4.2 Error bounds in terms of information energy

Some studies pertaining to bounds on the probability of error in fuzzy concepts are presented in [9], [8]. They are based on information energy for fuzzy events. The marginal probability distribution on fuzzy information $\mathcal{A}$ of the fuzzy event A is given by:

$$P_m(A) = \int_{\Re^d} \mu_A(x)p(x)dx, \tag{16}$$

where $p(x)$ is the unconditional likelihood.

The conditional information energy (in node i) given by the fuzzy event A is as follows:

$$E_i(P(\mathcal{M}^i|A_i)) = \sum_{k\in\mathcal{M}^i} (P(k|A_i))^2, \tag{17}$$

where $P(k|A_i) = \frac{P(k)\int_{\Re^d}\mu_{A_i}(x_i)f_k(x_i)dx_i}{P_i(A_i)}$.

The conditional information energy (in node i) of $\mathcal{M}^i$ given the fuzzy information $\mathcal{A}_i$ is as follows:

$$E_i(\mathcal{A}_i, \mathcal{M}^i) = \sum_{A_i\in\mathcal{A}_i} E_i(P(\mathcal{M}^i|A_i))P_i(A_i). \tag{18}$$

For such a definition of conditional information energy, the upper and lower bounds on probability of error for fuzzy data in node i, similarly as in [8], are represented by:

$$\frac{1}{2}(1 - E_i(\mathcal{A}_i, \mathcal{M}^i)) \leq Pe_F^{IE}(i) \leq (1 - E_i(\mathcal{A}_i, \mathcal{M}^i)). \tag{19}$$

Hence, the upper bound on the probability of misclassification for the fuzzy observations $Pe_F^{IE}(\pi_N^*)$ (in terms of information energy) for the globally optimal strategy of multistage recognition π_N^* is as follows:

$$Pe_F^{IE}(\pi_N^*) \leq 1 - \sum_{j_N\in\mathcal{M}(N)} p(j_N) \prod_{i\in s(j_N)-\{j_N\}} E_i(\mathcal{A}_i, \mathcal{M}^i). \tag{20}$$

4.3 The new upper bound on the probability of error in a binary tree classifier

The upper bound on the probability of error represented by (20) is very inaccurate and applies to every decision tree. Now we present a new upper bound on the probability of error for a binary decision tree. The new bound is tighter than the previous bound.

Theorem 1 *For a binary tree classifier, the upper bound on the probability of error for the fuzzy observations $Pe_F^{BT}(\pi_N^*)$ and for the globally optimal strategy of multistage recognition π_N^* is as follows*:

$$Pe_F^{BT}(\pi_N^*) \leq 1 - \sum_{j_N\in\mathcal{M}(N)} p(j_N) \prod_{i\in s(j_N)-\{j_N\}} (1 - Pe_F^{BT}(i)), \tag{21}$$

where

$$Pe_F^{BT}(i) = 1 - 0.25(1 + 2E_i(\mathcal{A}_i, \mathcal{M}^i) + \sum_{A_i \in \mathcal{A}_i} (|P_1(\omega_{\mathcal{M}^i}|A_i) - P_2(\omega_{\mathcal{M}^i}|A_i)|)P_m(A_i)).$$

In a binary tree classifier each of interior nodes has only two descendant nodes. Then the $P_1(\omega_{\mathcal{M}^i}|A_i)$ and $P_2(\omega_{\mathcal{M}^i}|A_i)$ are a posteriori probabilities of the immediate descendant node i.

Proof. For a binary tree classifier the probability of error in the interior node i can be expressed as:

$$Pe_F^{BT}(i) = 1 - \sum_{A_i \in \mathcal{A}_i} \max[P_1(\omega_{\mathcal{M}^i}|A_i), P_2(\omega_{\mathcal{M}^i}|A_i)]P_m(A_i). \tag{22}$$

For two numbers a, b if $a \in [0,1],\ b \in [0,1]$ the following inequalities occur:

$$\max[a,b] \geq 0.25(2a^2 + 2b^2 + |a-b| + a + b) \geq (a^2 + b^2). \tag{23}$$

The inequality $\max[a,b] \geq (a^2 + b^2)$ is used in [9] to prove that $Pe_F^{BT}(i) \leq 1 - E_i(\mathcal{A}_i, \mathcal{M}^i)$ holds. For the considered problem of pattern recognition from (22) and (23) we received:

$$Pe_F^{BT}(i) \leq 1 - 0.25 \sum_{A_i \in \mathcal{A}_i} (2P_1(\omega_{\mathcal{M}^i}|A_i)^2 + 2P_2(\omega_{\mathcal{M}^i}|A_i)^2 + \tag{24}$$

$$+|P_1(\omega_{\mathcal{M}^i}|A_i) - P_2(\omega_{\mathcal{M}^i}|A_i)| + P_2(\omega_{\mathcal{M}^i}|A_i) + P_2(\omega_{\mathcal{M}^i}|A_i))P_m(A).$$

From (18) and from $\sum_{A_i \in \mathcal{A}_i} P_1(\omega_{\mathcal{M}^i}|A_i) + P_2(\omega_{\mathcal{M}^i}|A_i) = 1$ the upper bound on probability of error in the node i is as follow:

$$\begin{aligned} Pe_F^{BT}(i) \quad &\leq 1 - 0.25(1 + 2E_i(\mathcal{A}_i, \mathcal{M}^i) + \\ &+ \textstyle\sum_{A_i \in \mathcal{A}_i} (|P_1(\omega_{\mathcal{M}^i}|A_i) - P_2(\omega_{\mathcal{M}^i}|A_i)|)P_m(A_i). \end{aligned} \tag{25}$$

The above estimation of error in the node i proves Theorem 1.

The bound represented by (21) is tighter than the previously introduced bound based on information energy (20). In the interior node i it is twice as precise as the previous bond. Now, we present the theorem for this relationship.

Theorem 2 *For the interior node of binary decision tree the bound in (25) is twice as precise as the bound in (19).*

Proof. For a binary tree classifier the probability of error in the interior node i can be expressed as:

$$Pe_F^{BT}(i) = \sum_{A_i \in \mathcal{A}_i} \min[P_1(\omega_{\mathcal{M}^i}|A_i), P_2(\omega_{\mathcal{M}^i}|A_i)]P_m(A_i). \tag{26}$$

For two numbers a, b if $a \in [0,1],\ b \in [0,1]$ the following inequality occurs:

$$\min[a,b] = 0.5(a + b - |a-b|). \tag{27}$$

From the last equality it follows that

$$\begin{aligned} Pe_F^{BT}(i) = \quad & 0.5 \sum_{A_i \in \mathcal{A}_i} (P_1(\omega_{\mathcal{M}^i}|A_i) + P_2(\omega_{\mathcal{M}^i}|A_i) - \\ & - |P_1(\omega_{\mathcal{M}^i}|A_i) - P_2(\omega_{\mathcal{M}^i}|A_i)|) P_m(A_i). \end{aligned} \tag{28}$$

It follows that

$$\sum_{A_i \in \mathcal{A}_i} P_1(\omega_{\mathcal{M}^i}|A_i) P_m(A_i) + \sum_{A_i \in \mathcal{A}_i} P_2(\omega_{\mathcal{M}^i}|A_i) P_m(A_i) = 1. \tag{29}$$

From the last expression we have

$$Pe_F^{BT}(i) = 0.5 - 0.5 \sum_{A_i \in \mathcal{A}_i} (|P_1(\omega_{\mathcal{M}^i}|A_i) - P_2(\omega_{\mathcal{M}^i}|A_i)|) P_m(A_i). \tag{30}$$

The right side of inequality (25) equals:

$$\begin{aligned} & 1 - 0.25(1 + 2E_i(\mathcal{A}_i, \mathcal{M}^i) + \textstyle\sum_{A_i \in \mathcal{A}_i} (|P_1(\omega_{\mathcal{M}^i}|A_i) - P_2(\omega_{\mathcal{M}^i}|A_i)|) P_m(A_i) = \\ & = -0.5E_i(\mathcal{A}_i, \mathcal{M}^i) + 1 - 0.25(1 + \textstyle\sum_{A_i \in \mathcal{A}_i} (|P_1(\omega_{\mathcal{M}^i}|A_i) - \\ & - P_2(\omega_{\mathcal{M}^i}|A_i)|) P_m(A_i)) = -0.5E_i(\mathcal{A}_i, \mathcal{M}^i) + 0.75 - 0.25 \times \\ & \times (\textstyle\sum_{A_i \in \mathcal{A}_i} (|P_1(\omega_{\mathcal{M}^i}|A_i) - P_2(\omega_{\mathcal{M}^i}|A_i)|) P_m(A_i)) = -0.5E_i(\mathcal{A}_i, \mathcal{M}^i) + 0.5 + \\ & + (0.25 - 0.25(1 + \textstyle\sum_{A_i \in \mathcal{A}_i} (|P_1(\omega_{\mathcal{M}^i}|A_i) - P_2(\omega_{\mathcal{M}^i}|A_i)|) P_m(A_i))) = \\ & = -0.5E_i(\mathcal{A}_i, \mathcal{M}^i) + 0.5 + 0.5 Pe_F^{BT}(i) = (1 - E_i(\mathcal{A}_i, \mathcal{M}^i) + Pe_F^{BT}(i))/2. \end{aligned}$$

This is the half value of the upper bound (19).

5. An Illustrative Example

Let us consider the two-stage binary tree classifier. Four classes have identical a priori probabilities that equal 0.25. We use 3-dimensional data $x = [x^{(1)},\ x^{(2)},\ x^{(3)}]$ where class-conditional probability density functions are normally distributed. For performing the classification at the root-node 0, the first coordinate was used, and components $x^{(2)}$ and $x^{(3)}$ were used at the nodes 5 and 6 respectively. In the data, covariance matrices are equal for every class $\sum_{j_2} = 1I,\ j_2 \in \mathcal{M}(2)$, and the expected values are as follows: $\mu_1 = [1,\ 1,\ 0]$, $\mu_2 = [1,\ 2,\ 0]$, $\mu_3 = [3,\ 0,\ 1.5]$, $\mu_4 = [3,\ 0,\ 3]$. In experiments, the following sets of fuzzy numbers were used:

$$\mathcal{A}_1 = \mathcal{A}_2 = \mathcal{A}_3 = \{A^1 = (-\infty,\ 0,\ 0.5),\ A^2 = (0,\ 0.5,\ 1), \ldots, A^8 = (3,\ 3.5,\ \infty)\}$$

Tab. I shows the error in the interior nodes of decision tree. This error is calculated for non fuzzy observations of features $1 - Pc(i)$ and for fuzzy observations. The bound of error for the fuzzy data presented in this paper $Pe_F^{BT}(i)$ is twice as

	node		
	0	5	6
$1 - Pc(i)$	0.159	0.309	0.227
$1 - Pc_F(i)$	0.168	0.315	0.228
$1 - E_i(\mathcal{A}_i, \mathcal{M}^i)$	0.231	0.402	0.316
$Pe_F^{BT}(i)$	0.199	0.309	0.227

Tab. I *The error in the interior nodes of decision tree.*

$Pe(\pi_N^*)$	0.384
$Pe_F(\pi_N^*)$	0.394
$Pe_F^{IE}(\pi_N^*)$	0.507
$Pe_F^{BT}(\pi_N^*)$	0.451

Tab. II *The probability of misclassification for the global optimal strategy.*

precise as the bound based on information energy $1 - E_i(\mathcal{A}_i, \mathcal{M}^i)$. Tab. II shows the same error for the global optimal strategy of binary tree classifier. These results are calculated for full probabilistic information.

The obtained results show deterioration in the classification quality when we use fuzzy information on object features instead of exact information in a binary tree classifier. The first series of rows in Tab. I and Tab. II relates to the precise observation – not the fuzzy one. For these rows the calculated values of errors are the smallest. The second series of rows shows the deterioration of the quality of classification when we have fuzzy observations rather than the precise ones. These values are accurate. In the next series of rows we estimate these exact values. In the third series of rows, there is the estimation of error based on information energy. The last series of rows contains the estimation discussed in this work.

6. Conclusion

In this present paper, we have concentrated on the binary tree classifier. Assuming full probabilistic information, we have presented the difference between the probability of misclassification for fuzzy and crisp data. In the paper we presented a new upper bound on the probability of error in the binary tree classifier. The obtained results are compared with the bound based on the information energy of fuzzy events. For interior nodes of decision tree, the new bound is twice as precise as the bound based on information energy. The obtained results were demonstrated on a numerical example. The performance of Bayes classifiers is expressed in terms of the probability of error. In this work we showed how to improve the estimation of error, which concerns the binary tree classifier with exact classes and fuzzy information on object features.

Acknowledgements

This research is supported by The Polish State Committee for Scientific Research under the grant, which is realized in 2010–2013.

References

[1] Antos A., Devroye L., Gyorfi L.: Lower bounds for Bayes error estimation. IEEE Trans. Pattern Analysis and Machine Intelligence, 21, 1999, pp. 643–645.

[2] Avi-Itzhak H., Diep T.: Arbitrarily tight upper and lower bounds on the bayesian probability of error. IEEE Trans. Pattern Analysis and Machine Intelligence, 18, 1996, pp. 89–91.

[3] Burduk R., Kurzyński M.: Two-stage binary classifier with fuzzy-valued loss function. Pattern Analysis and Applications, **9**, 4, 2006, pp. 353–358.

[4] Burduk R.: Classification error in Bayes multistage recognition task with fuzzy observations. Pattern Analysis and Applications, **13**, 1, 2010, pp. 85–91.

[5] Kulkarni A.: On the mean accuracy of hierarchical classifiers. IEEE Transactions on Computers, 27, 1978, pp. 771–776.

[6] Kurzyński M.: On the multistage Bayes classifier. Pattern Recognition, 21, 1988, pp. 355–365.

[7] Okuda T., Tanaka H., Asai K.: A formulation of fuzzy decision problems with fuzzy information using probability measures of fuzzy events. Information and Control, 38, 1978, pp. 135–147.

[8] Pardo J. A, Taneja I. J.: On the Probability of Error in Fuzzy discrimination Problems. Kybernetes, **21**, 6, 1992, pp. 43–52.

[9] Pardo L., Menendez M. L.: Some Bounds on Probability of Error in Fuzzy Discrimination Problems. European Journal of Operational Research, 53, 1991, pp. 362–370.

[10] Pedrycz W.: Fuzzy Sets in Pattern Recognition: Methodology and Methods. Pattern Recognition, 23, 1990, pp. 121–146.

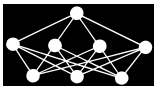
[11] Stańczyk U.: Dominance-Based Rough Set Approach Employed in Search of Authorial Invariants, Advances in Intelligent and Soft Computing 57, Springer-Verlag, Berlin Heidelberg, 2009, pp. 293–301.

[12] Supriya K. D., Ranjit B., Akhil R. R.: An application of intuitionistic fuzzy sets in medical diagnosis. Fuzzy Sets and Systems, **117**, 2, 2001, pp. 209–213.

[13] Woźniak M.: Experiments on linear combiners. Advances in Soft Computing, Springer-Verlag, Berlin Heidelberg, 47, 2008, pp. 445–452.

[14] Zadeh L. A.: Fuzyy sets. Information and Control, **8**, 3, 1965, pp. 338–353.

[15] Zadeh L. A.: Probability measures of fuzzy events. Journal of Mathematical Analysis and Applications, 23, 1968, pp. 421–427.



RANKED TAG RECOMMENDATION SYSTEMS BASED ON LOGISTIC REGRESSION

*J. R. Quevedo, E. Montañés, J. Ranilla, I. Díaz**

Abstract: This work proposes an approach to tag recommendation based on a learning system. The goal of this method is to support users of current social network systems by providing a rank of new meaningful tags for a resource. This system provides a ranked tag set and it feeds on different posts depending on the resource for which the user requests the recommendation. This research studies different approaches depending on both the posts selected to form the training set and the features with which they are represented. The performance of these approaches are tested according to several evaluation measures; one of them is proposed in this paper F_1^+ which takes into account the positions where the system has ranked the positive tags at the same time that it considers the cases where positive tags could not be ranked. These experiments show that this learning system outperforms certain benchmark recommenders.

Key words: *Recommendation systems, logistic regression, ranking systems*

Received: 20th September 2010
Revised and accepted: 13th November 2010

1. Introduction and Related Work

Tagging can be defined as the process of assigning short textual descriptions, called tags, to information resources, which allows the user to organize the content. This becomes very popular and helpful for large-scale systems such as Folksonomies. A Folksonomy [8] is a collection of resources entered by users in posts. Each post consists of a resource and set of keywords (tags), attached by a user. Generally, the resource is specific to the user who added it to the system, but all users are invited to label it with tags. These systems can be distinguished according to the kind of resources they support. `Flickr`, for instance, allows sharing photos, `Del.icio.us` shares bookmarks, and Bibsonomy allows to share both bookmarks and bibtex entries.

*J. R. Quevedo, E. Montañés, J. Ranilla, I. Díaz
Artificial Intelligence Center, University of Oviedo, Spain, E-mail: `quevedo@uniovi.es, montaneselena@uniovi.es, ranilla@uniovi.es, sirene@uniovi.es`

This paper proposes an approach to tag recommendation based on a learning process. The work starts from the hypothesis that a learning process improves the performance of the recommendation task. In this sense, the learner is fed on several examples. It also analyzes the usefulness and suitability of recent posts in recommending new tags.

Different approaches have been proposed to support users during the tagging process depending on the purpose for which they were built. Some of them make recommendations by analyzing content [1], analyzing tag co-occurrences [17] or studying graph-based approaches [10].

Brooks et al. [4] analyze the effectiveness of tags for classifying blog entries by measuring the similarity of all articles that share a tag. Jäschke et al. [10] adapt a user-based collaborative filtering as well as a graph-based recommender built on top of FolkRank. Basile et al. [3] propose a smart TRS able to learn from past user interaction as well from as the content of the resources to annotate. Katakis et al. [12] model the automated tag suggestion problem as a multi-label text classification task. Sigurbjörnsson et al. [17] present the results by means of a tag characterization focusing on how users tags photos of `Flickr` and what information is contained in the tagging.

Most of these systems require information associated with the content of the resource itself [3]. Others simply suggest a set of tags as a consequence of a classification rather than providing a ranking of them [12]. Some of them require a large quantitative of supporting data [17]. The proposal of this work avoids these drawbacks through a novel approach which ranks the tags using a machine learning approach based on Logistic Regression.

The remainder of the paper is structured as follows. Our approach is put in context in Section 2, while the proposed method is provided in Section 3. Section 4 details some novel performance evaluation metrics. The results conducted on public data sets are presented and analyzed in Section 5. Finally, Section 6 draws conclusions and points out some possible challenges to address in the near future.

2. Tag Recommender Systems (TRS)

A folksonomy is a tuple $F := (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{Y})$ where $\mathcal{U}$, $\mathcal{T}$ and $\mathcal{R}$ are finite sets, whose elements are respectively called users, tags and resources, and $\mathcal{Y}$ is a ternary relation between them, i. e., $\mathcal{Y} \subseteq \mathcal{U} \times \mathcal{T} \times \mathcal{R}$, whose elements are tag assignments (posts). When a user adds a new or existing resource to a folksonomy, it could be helpful to recommend him/her relevant tags.

TRS usually take the users, resources and the ratings of tags into account to suggest a list of tags to the user. According to [14] a TRS can briefly be formulated as a system that takes as input a given user $u \in \mathcal{U}$ and a resource $r \in \mathcal{R}$ and produces a set $\mathcal{T}(u, r) \subset \mathcal{T}$ of tags as output.

Jäschke et al. in [10] defines a post of a folksonomy as a user, a resource and all tags that this user has assigned to that resource. This work slightly modifies this definition in the sense that it restricts the set of tags to the tags used simultaneously to tag a resource by a user.

There exist some simple but frequently used TRS [10] based on providing a list of ranked tags extracted from the set of posts connected with the current annotation.

- MPT (Most Popular Tags): For each tag t_i, the posts with t_i are counted and the top tags (ranked by occurrence count) are utilized as recommendations.
- MPTR (Most Popular Tags by Resource): For a resource r_i it is counted for every tag in how many posts they occur together with r_i. The tags that occurred most often together with r_i are then proposed as recommendations.
- MPTU (Most Popular Tags by User): For a user u_i the amount of posts in which they occur together with u_i is counted. The tags occurring most often together with u_i are taken as recommendations.
- MPTRU (Most Popular Tags by Resource or User): For a resource r_i the number of posts in which they occur together with r_i is counted. In addition, for a user u_i the amount of posts in which they occur together with u_i is counted as well. The tags occurring most often together with either r_i or u_i are taken as recommendations.

The introduction of a learning system is expected to improve their performance.

3. Learning to Recommend

This section depicts the whole procedure followed for providing a set of ranked tags for a user and a resource. Such recommendations are based on a learning process which leans upon how everyone has tagged resources before. The core of the method is a supervised learning algorithm based on SVM with probabilistic output [5]. This paper studies different training sets built according to the user and resource for which the recommendations are provided.

The key points of the system are the following:

- The test set is not fixed. Instead it is randomly built.
- The training set depends on each test set and it is built specifically for each test set.
- Several training sets are built according to different criteria and afterwards compared and evaluated.
- The learning system adopted was LIBLINEAR [5], which provides a probabilistic distribution before the classification. This probability distribution is exerted to rank the tags, taking as most suitable tag the one with highest probability value.
- The tags of the ranking will be all that were in the categories of the training set. This entails that some positive tags of a test set might not be ranked.

3.1 Definition of the test set

Several works follow the traditional splitting data into training and test sets. Thus, they learn from a fixed training set and recommend a tag set for each post in test set [12]. The approach adopted in this paper is quite different in the sense that

several test posts are randomly selected from the original data set, and an *ad hoc* training set is provided to each test set.

A folksonomy is composed of a set of posts. Each post is formed by a user, a resource and a set of tags, i.e.,

$$p_i = (u_i, r_i, \{t_{i_1}, \ldots, t_{i_k}\}).$$

Each post of a folksonomy is a candidate to become a test post. Each test post is then turned into as many examples as tags used to label the resource. Therefore, post p_i is split into k test examples

$$\begin{aligned} e_1 &= (u_i, r_i, t_{i_1}) \\ &\vdots \\ e_k &= (u_i, r_i, t_{i_k}). \end{aligned} \tag{1}$$

Example 1 Suppose the following folksonomy

post	*date*	*User*	*Resource*	*Tags*
p_1	d_1	u_1	r_1	t_1
p_2	d_2	u_1	r_2	t_2
p_3	d_3	u_2	r_1	t_1
p_4	d_4	u_3	r_1	t_3
p_5	d_5	u_2	r_2	t_4
p_6	d_6	u_2	r_1	t_2, t_3
p_7	d_7	u_2	r_2	t_2, t_5
p_8	d_8	u_3	r_2	t_1

(2)

Let $p_7 = (u_2, r_2, \{t_2, t_5\})$ be a randomly selected test post at instant d_7. Therefore, the test set is formed by

example	*date*	*User*	*Resource*	*Tags*
e_1	d_7	u_2	r_2	t_2
e_2	d_7	u_2	r_2	t_5

(3)

3.2 Definition of the training set

Whichever learning system strongly depends on the training set used to learn. In fact, the ideal situation is that the distribution of the categories in both training and test sets are as similar as possible to guarantee a better learning. Therefore, the selection of an adequate training set is not a trivial task that must be carefully carried out.

Once the test set is randomly selected, an *ad hoc* training set is dynamically selected from the posts posted *before* the test post. The proposal selects the training set from the N most recent posts. The parameter N is experimentally fixed.

This characteristic makes the TRS to suggest the on-fashion folksonomy tags and it produces a more scalable system, since the number of posts in the training set does not increase according to the number of posts posted before the test post. This characteristic makes feasible the problem by the learning system.

Therefore, the selection of the training set for a given test post is reduced to define the criterion the posts must satisfy to be included in the training set. This work studies different approaches criteria.

Approach 1. TR Let $p_i = (u_i, r_i, \{t_{i_1}, \ldots, t_{i_k}\})$ be a test post. Let $\mathcal{R}_{u_i}$ be the subset of posts associated to a resource r_i and

$$\mathcal{R}^t_{r_i} = \{p_i / p_i \in \mathcal{R}_i \; and \; it \; was \; posted \; before \; t\}.$$

TR approach selects as training set the N most modern posts of $\mathcal{R}^{d_i}_{r_i}$, being d_i the date when p_i was posted.

Approach 2. TU Let $p_i = (u_i, r_i, \{t_{i_1}, \ldots, t_{i_k}\})$ be a test post. Let $\mathcal{P}_{u_i}$ be the personomy (the subset of posts posted by a user constitutes the so-called personomy) associated to a user u_i and

$$\mathcal{P}^t_{u_i} = \{p_i / p_i \in \mathcal{P}_{u_i} \; and \; it \; was \; posted \; before \; t\}.$$

TU approach selects as training set the N most modern posts of $\mathcal{P}^{d_i}_{u_i}$, being d_i the date when p_i was posted.

Approach 3. TRU The above training sets do not take into account that the learned model is used after a resource is presented to the user. Hence, this approach proposes to go further and to add as training examples those concerning with the resource for which the recommendations are demanded.

Since it makes no sense to recommend to a user those tags that he has previously used to label the resource, the examples whose tags have been previously assigned to the resource by the user to whom the recommendations are provided have been removed.

Therefore, the training set associated to p_i is formed by

$$UR^{d_i}_{u_i, r_i} = \{\mathcal{P}^d_i \cup \mathcal{R}^d_i\} \backslash \{p_j / p_j = (u_i, r_i, \{t_1, ..., t_n\})\}.$$

Example 2 Let us show an example of each training set for the test set of Example 1.

- Approach 1. TR. Firstly the set $\mathcal{R}^{d_7}_{r_2} = \{p_2, p_5\}$ is computed. Therefore, the training set is

$$\begin{array}{ccccc} example & date & User & Resource & Tags \\ e_2 & d_2 & u_1 & r_2 & t_2 \\ e_5 & d_5 & u_2 & r_2 & t_4 \end{array} \quad (4)$$

- Approach 2. TU

  $\mathcal{P}_{u_2} = \{p_3, p_5, p_6, p_7\}$ and $\mathcal{P}^{d_7}_{u_2} = \{p_3, p_5, p_6\}$

since these posts were posted by user u_2 before p_7.

example	*date*	*User*	*Resource*	*Tags*
e_3	d_3	u_2	r_1	t_1
e_5	d_5	u_2	r_2	t_4
e_{6_1}	d_6	u_2	r_1	t_2
e_{6_2}	d_6	u_2	r_1	t_3

(5)

- Approach 3. TRU. In this case, the training set is computed as follows.

$$UR^{d_7}_{u_2,r_2} = \{\mathcal{P}^{d_7}_{u_2} \cup \mathcal{R}^{d_7}_{r_2}\} \backslash \{p_j / p_j = (u_i, r_i, \{t_1, ..., t_n\})\} =$$
$$\{\{p_3, p_5, p_6\} \cup \{p_2, p_5\}\} \backslash \{p_5\} = \{p_2, p_3, p_6\}.$$

Therefore, the training set is defined as follows.

example	*date*	*User*	*Resource*	*Tags*
e_2	d_2	u_1	r_2	t_2
e_3	d_3	u_2	r_1	t_1
e_{6_1}	d_6	u_2	r_1	t_2
e_{6_2}	d_6	u_2	r_1	t_3

(6)

3.3 Example representation

Once both the training and test sets are defined, it is necessary to transform them into a computable form understandable for a machine learning system. Therefore, we have to define the features characterizing the examples as well as the class of each example.

The features which characterize the examples are the tags previously used to tag the resource in the folksonomy. Hence, each example will be represented by a Boolean vector V of size M (the number of tags of the folksonomy), where $v_j = 1$ if and only if t_j was used to tag the resource before and 0 otherwise, where $j \in 1, \ldots, M$. The class of an example will be the tag with which the user has tagged the resource in this moment.

Example 3 As an illustration of how to represent a example, let us represent example e_{6_1} of Example 2. The class of e_{6_1} is t_2, which is its corresponding tag. The features are t_1 and t_3, since the resource 2 of e_{6_1} was also tagged before by t_1 in e_1 and e_3 and by t_3 in e_4.

The representation of example e_{6_1} is then $\{1, 0, 1, 0\}$.

Approach 4. Feature Selection (TRUTR) Removing redundant or non-useful features which add noise to the system is usually helpful to increase both the effectiveness and efficiency of the classifiers. The example representation based on tags as features makes it possible to perform a simple feature selection in the training set consisting of just keeping those tags which represent the test.

Obviously, this is possible just in case the information about the resource of the test post is considered for building the training set, that is, for TR and TRU approaches. This approach is based on the fact that in a linear system, as the one adopted here, the weights of the features neither represent the test post nor contribute to obtain the ranking for this post. Therefore, they could be considered as irrelevant features beforehand. This fact can be assumed only for a particular test post. So this is another advantage of building a training set particularly for each test post. Let us consider the test post of Example 1 and the training set of Approach 3 in Example 2.

Example 4 The features for the test post are t_2 and t_4, hence, the training set of Approach 3 in Example 2 will be reduced to be represented at most with these two tags. Originally, that training set has the following representation:

example	*date*	*resource*	*features*	*category*
e_2	d_2	r_2	$\emptyset$	t_2
e_3	d_3	r_1	t_1	t_1
e_{6_1}	d_6	r_1	t_1, t_3	t_2
e_{6_2}	d_6	r_1	t_1, t_2, t_3	t_3

(7)

In the folksonomy represented in Example 1, resource r_2 does not have any tag assigned before instant d_2, then its representation is an empty set of features. Analogously, resource r_1 has only be tagged before instant d_3 with t_1, particularly in instant d_1 by user u_1, then it is represented only by feature t_1. Special attention has been paid to resource r_1 tagged in instant d_6. Since, this resource has been tagged before d_6 with t_1 and t_3, then both tags are included in its representation. Besides, in example e_{6_1} when the category is t_2, the tag t_3 is also added because it is a tag assigned in the same instant. In the same way, in example e_{6_2} when the category is t_3, the tag t_2 is included, since it is a tag assigned in the same instant.

Reducing such representation to the tags of the test post, the results of this new approach are

example	*date*	*resource*	*features*	*category*
e_2	d_2	r_2	$\emptyset$	t_2
e_3	d_3	r_1	$\emptyset$	t_1
e_{6_1}	d_6	r_1	$\emptyset$	t_2
e_{6_2}	d_6	r_1	t_2	t_3

(8)

3.4 Learning system

The key point of this paper is to provide a ranked set of tags adapted to a user and to a resource. Therefore, it could be beneficial to have a learning system able to rank the tags, indicating to the user which tag is the best and which one is the worst for the resource. Taking into account this fact, a preference learning system can not be applied, since that kind of methods yield a ranking of the examples (posts), rather than a ranking of categories (tags) [11]. As the input data are

multi-category, a multi-category system is expected to be used. However, these systems do not provide a ranking.

The system we need must provide a global ranking of labels. Therefore, a multi-label system could be used, but again they need an adaptation to deal with ranking problems. In fact, some multi-label classification systems perform a ranking and then they obtain the multi-label classification [18]. Hence, it is possible to obtain from them a ranking directly. Elisseeff and Weston [6] propose a multi-label system based on SVM, which generates a ranking of categories. The drawback is that the complexity is cubic and although they perform an optimization to reduce the order to be quadratic, they admit that such complexity is too high to apply to real data sets. Platt [16] uses SVM to obtain a probabilistic output, but just for a binary classification and not for multi-category.

With regard to the problem of tag recommendation, Godbole and Sarawagi in [7] present an evolution of SVM based on extending the original data set with extra features containing the predictions of each binary classifier and on modifying the margin of SVMs in multi-label classification problems. The main drawback is that they perform a classification rather than a ranking.

In this framework, LIBLINEAR [5] is an open source library[1] based on SVM which is a recent alternative able to accomplish multi-category classification through logistic regression, providing a probabilistic distribution before the classification. This probability distribution is exerted to rank the tags, taking as most suitable tag the one with highest probability value. In the same sense, the most discordant tag will be the one with lowest probability.

This work uses the default LIBLINEAR configuration after a slight modification of the output. The evaluation in this case takes place when a resource is presented to the user. Then, a ranking of tags (the tags of the ranking will be all that were in the categories of the training set) is provided by the learning model, and afterwards the tags that such user has been previously posted to such resource are removed, since it has no sense to recommend a tag for a resource to a user who has previously tagged this resource with this tag.

If such resource has not been previously tagged, the ranking is generated according to a priori probability distribution, which consists of ranking the tags of the user according to the frequency this user has used with them before. So, no learning process is performed in this last case.

4. Performance Evaluation

So far, no consensus about an adequate metric to evaluate a recommender is stated [10]. Some works do not include quantitative evaluation [19] or they include it partially [15]. However, the so called LeavePostOut or LeaveTagsOut proposed in [14] and [10] sheds light on this issue. They pick up a random post for each user and they provide a set of tags for this post based on the whole folksonomy except

[1] available at http://www.csie.ntu.edu.tw/∼cjlin/liblinear/

such post. Then, they compute the precision and recall ([12]) as follows

$$recall(T) = \frac{1}{|D|} \sum_{(u,r)\in D} \frac{|\mathcal{T}^+(u,r) \cap \mathcal{T}(u,r)|}{|\mathcal{T}^+(u,r)|} \tag{9}$$

$$precision(T) = \frac{1}{|D|} \sum_{(u,r)\in D} \frac{|\mathcal{T}^+(u,r) \cap \mathcal{T}(u,r)|}{|\mathcal{T}(u,r)|}, \tag{10}$$

where D is the test set, $\mathcal{T}^+(u,r)$ are the set of tags user u has assigned to resource r (positive tags) and $\mathcal{T}(u,r)$ are the set of tags the system has recommended to user u to assign resource r. The F_1 measure could be computed from them as

$$F_1 = \frac{1}{|D|} \sum_{(u,r)\in D} \frac{2|\mathcal{T}^+(u,r) \cap \mathcal{T}(u,r)|}{|\mathcal{T}(u,r)| + |\mathcal{T}^+(u,r)|}. \tag{11}$$

The main drawback of this process of evaluation is that it just evaluates the performance of a classification rather that the performance of a ranking, since the positions where the system has ranked the positive tags are not considered. But, a TRS able to return the positive tags at the top of the ranking is obviously preferred than one that returns the positive tags at the bottom of the ranking. Hence, defining an evaluation metric able to quantify both the tags a TRS recommends and the order in which it ranks them is an expected challenge to cope with.

The Area Under the ROC Curve (AUC) [13] and Average Precision (AP) [2] are measures to evaluate a ranking a priori. AUC is the probability of a correct ranking; in other words, it is the probability that a randomly chosen subject of class +1 is (correctly) ranked with greater output than a randomly chosen subject of class -1. AP is the average of the precision computed in the positions where a positive tag is ranked.

The Normalized Discounting Cumulative Gain (NDCG) [9] is another evaluation measure for Information Retrieval (IR) systems that compute the cumulative gain a user obtains by examining the retrieval result up to a given ranked position.

It has two particularities. Firstly, it applies a discount factor on the relevance in order to devaluate late-retrieved documents. Secondly, it computes a relative score with regard to the ideal cumulative gain.

But, both of them present some drawbacks in tag recommendation because the rankings to compare could not have the same number of tags, and it is possible that some tags of the test never appear in the training. In fact, all of them are thought to compare permutations of a predefined set of tags.

Let us illustrate these statements presenting some situations. Imagine that the number of positive tags is g and a ranking R of length l a sequence of positive (p) and negative (n) tags.

- Situation 1. Let us compare rankings of the form $[n^b p^a]$ with $a \geq 0$ and $b \geq 1$. Then, for a given value of b, the better the ranking the greater the value of a. In this situation, both AP and NDCG satisfy that condition, but AUC is always zero, since all positive tags are not correctly ranked since there is at least one negative tag in the first position.

- Situation 2. Let us compare rankings of the form $[p^a n^b]$ with $a \geq 1$ and $b \geq 0$. Then, for a given value of b, the better the ranking the greater the value of a. NDCG satisfies it, but both AUC and AP take value to 1. This happens because all positive tags are ranked correctly without any negative tag in between. Hence, they do not differ a ranking that contains all positive tags than other which only contains a few.
- Situation 3. Let us compare now the ranking $R_1 = [p]$ with the ranking $R_2 = [pn^b p^a]$ with $a \geq 1$ and $b \geq 1$. In this situation, it is necessary to establish a trade-off between precision and recall, and it is not easy to state a simple rule. AUC and AP always grant greater value to R_1 and NDCG to R_2 independently of the values of a and b. Hence, an ideal measure would establish a limit to b from which R_1 would be better ranking than R_2.

This paper proposes an alternative which tries to overcome those drawbacks. It is similar to the LeavePostOut mentioned above since several pairs of user and resource are randomly chosen, but it takes into account the positions the TRS provides the positive tags. It consists of computing the F_1 measure for all possible cutoffs of the ranking for which a positive tag is returned and for choosing the highest one. It will be denoted by F_1^+ and it is defined by

$$F_1^+ = \max_{0<=i<=r} (F_1)_i, \tag{12}$$

where r is the size of the ranking, that is, the number of tags returned by the system, $(F_1)_i$ is the F_1 of the classification assuming that the system has classified the first i tags as positive ones, and the rest as negative ones. Notice that i ranges from 0 (which means that the system has not returned any tag as positive) to r (which means that the system has returned all the tags as positive).

Since a negative tag in the ranking does not lead to an improvement of the F_1, only computing F_1^+ for the cutoffs where a positive tag is placed is required. Hence, this metric gives an optimal position of the ranking. Notice that it does not vary if negative tags are added to the tail of the ranking, as it happens to AP. But, it takes into account all positive tags and not just the positive tags which appear on the top of the ranking, as AUC also does. Let us take up again the situations discussed above and find out what F_1^+ establishes.

- Situation 1. In this situation, $F_1^+ = \frac{2a}{a+b+g}$ and it satisfies that for a given value of b the better the ranking the greater the value of a.
- Situation 2. In this situation, the better cutoff takes place just before the first negative tag is ranked, hence $F_1^+ = \frac{2a}{a+g}$, and it also satisfies that for a given value of b the better the ranking the greater the value of a.
- Situation 3. In this situation, $F_1^+ = \frac{2}{1+g}$ for R_1 and $F_1^+ = \max\{\frac{2}{1+g}, \frac{2(b+1)}{a+b+1+g}\}$ for R_2 depending on where the cutoff is performed. Hence, R_2 is preferred to R_1 if $a < gb$.

5. Experiments

5.1 Data sets

The experiments were carried out over the collections `bm08` and `bt08`, which is a dataset formed respectively by bookmarks bibtex posts extracted from ECML PKDD Discovery Challenge 2008 (extracted from Bibsonomy bookmarking) and publication-sharing system that enables users to tag web documents as well as bibtex entries of scientific publications.

A user may store and organize bookmarks (web pages) and bibtex entries. The main tool provided for content management in Bibsonomy is tagging. Users can freely assign tags to bookmark or bibtex resources when they submit them to the system.

Before using the data sets, tag cleaning was made according to PKDD Discovery Challenge 2008. The preprocessing phase included removing useless tags (e.g., system:unfiled), changing all letters to lower case and removing non-alphabetical and non-numerical characters from tags. After this preprocessing, some tags become empty, hence all the posts with such tags are removed from the data set. Finally, the number of users, tags, resources and posts of both collections have been counted and shown in Tab. I.

Dataset	users	tags	resources	posts
`bm08`	1,953	42,302	177,387	563,990
`bt08`	1,206	29,739	96,616	278,008

Tab. I *Statistics of data sets.*

5.2 Discussion of results

This section deals with the experiments carried out. To test the methods, 1000 test posts were randomly selected. For each one, several ranked tag sets are provided depending on: the approach that builds the training set, which leads to four possible TRS: `TR`, `TU`, `TRU` and `TRUTR`, and on the cardinality (N) of the training set ($N = i * 500$ with $i = 1, 2, \ldots, 10$). Therefore, the size of the training set is also tested. Hence, each recommender has been trained with N examples posted immediately before the post for which the recommendation has been demanded. Thus, the conditions of each way of building the training set are satisfied.

Tab. II shows the behavior of the benchmark TRSs according to the evaluation metrics detailed in 4. It seems that both `MPTU` and `MPTRU` are considerably better than the rest benchmark TRSs for both data sets, although the latter is better. Hence, `MPTRU` is a good choice to consider as reference from now on to check if, indeed, a learning process helps to recommend tags more efficiently.

Fig. 1 (Fig. 2) shows the F_1^+, the AUC, the AP and the NDCG evaluation measures for the collection **bst08** (**bmst08**). Clearly, all the evaluation measures enhance the performance when the training set is formed by the posts added by

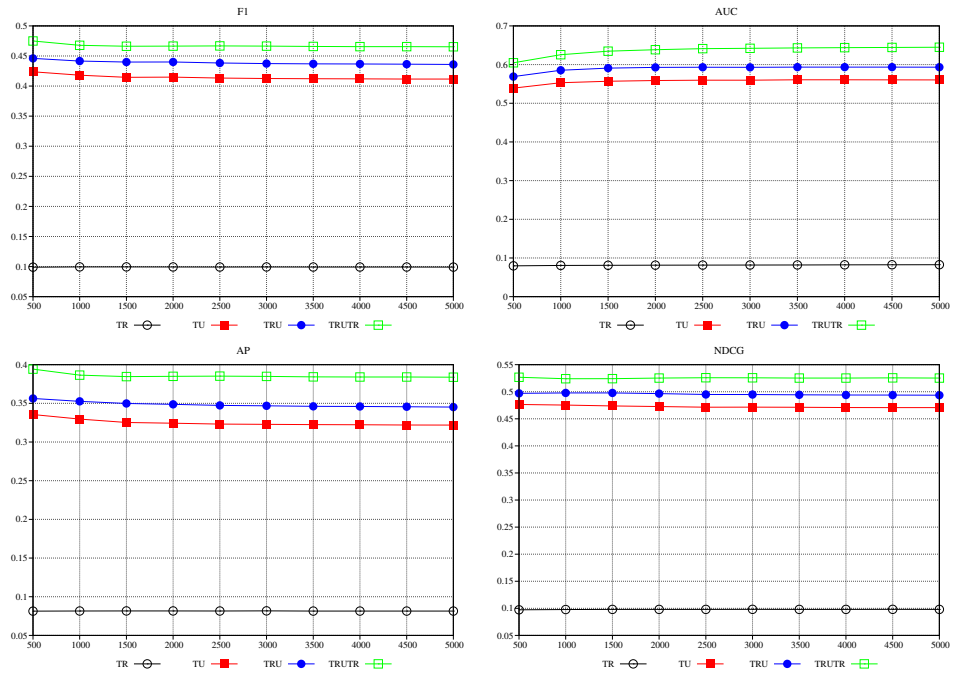


Fig. 1 *F1, AUC, AP and NDCG for btst08 data set.*

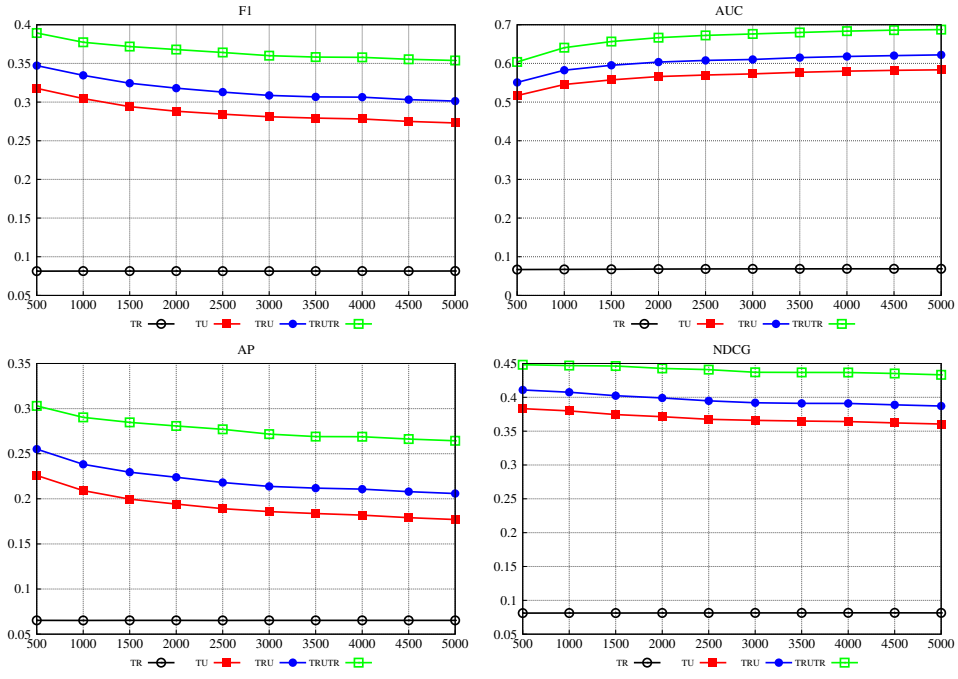


Fig. 2 *F1, AUC, AP and NDCG for bmst08 data set.*

	F_1^+	*AUC*	AP	NDCG
		bm08		
MPT	7.0%	5.9%	3.8%	6.7%
MPTR	5.2%	3.7%	3.5%	5.0%
MPTU	23.4%	65.7%	15.7%	31.4%
MPTRU	23.7%	66.7%	16.0%	32.0%
		bt08		
MPT	6.7%	5.8%	4.6%	6.6%
MPTR	7.8%	5.2%	5.7%	7.7%
MPTU	37.2%	56.3%	28.8%	42.5%
MPTRU	38.2%	57.9%	29.7%	43.7%

Tab. II *Performance of the benchmark TRSs for all post collections.*

the user of the test post. In addition, the effectiveness of the recommender is also improved if the posts related to the resource of the test post are included in the training set (described according to the test representation).

Let us analyze now the influence of the size of the training set (N): All evaluation metrics keep more or less steady for `bt08` collection. However, F_1^+ and AP slightly decrease as N increases, while AUC keeps steady when N is over 2500 in case of `bm08`. This peculiar behavior of AUC might be because it increases at the same time the negative posts that are ranked on the tail of the ranking. In any case, a steady situation is probably easy to reach.

6. Conclusions and Future Work

This work proposes a TRS based on a novel approach which learns to rank tags from previous posts in a folksonomy using a SVM with probabilistic output. This TRS is trained with three different sets, obtaining three different versions. This system feeds on different posts depending on the strategy used. They were tested over 2 different data sets and then compared to other TRSs.

In addition, a new evaluation measure is proposed, namely F_1^+. It takes into account the positions where the system has ranked the positive tags at the same time that it considers the cases where positive tags could not be ranked. In this way, it overcomes the drawbacks of other ranking evaluation metrics, such as AUC, AP and NDCG.

The TRSs proposed are compared to the best benchmark TRS (`MPTRU`). The results show a significant improvement of all the TRSs with regard to `MPTRU`, being the one which takes into account test representation (`TRUTR`) of the best of the four versions.

On the other hand, the cardinality of the training was ranged from 500 to 5000. The results show that the size of the training set hardly has effect on the performance. Only AUC seems to be sensitive for low values. In any case, it is possible to keep the performance without making the learning process slow down so much, since it is not necessary to add a huge amount of training examples.

Therefore, the introduction of a learning system becomes beneficial for recommending tags. Additionally, it is possible to state that for recommending suitable and useful tags, the training set should contain both the posts related to the *test* user or resource. However, what is really helpful is to represent the post only with tags that also represent the resource of the post for which the recommendations are provided. Since example and feature selection improves the performance of a learning based TRS, it would be interesting to explore new approaches in that direction as future work.

Acknowledgement

Authors acknowledge financial support by Grants TIN2007-61273 from the Ministry of Education and Science and TIN2010-14971 from the Ministry of Science and Innovation.

References

[1] Adrian B., Sauermann L., Roth-Berghofer T.: Contag: A semantic tag recommendation system. In: Proceedings of I-Semantics' 07, 2007, pp. 297–304.

[2] Aslam J. A., Yilmaz E., Pavlu V.: A geometric interpretation of r-precision and its correlation with average precision. In: Ricardo A. Baeza-Yates, Nivio Ziviani, Gary Marchionini, Alistair Moffat, and John Tait, editors, SIGIR, ACM, 2005, pp. 573–574.

[3] Basile P., Gendarmi D., Lanubile F., Semeraro G.: Recommending smart tags in a social bookmarking system. In: Bridging the Gep between Semantic Web and Web 2.0 (SemNet 2007), 2007, pp. 22–29.

[4] Brooks C. H., Montanez N.: Improved annotation of the blogosphere via autotagging and hierarchical clustering. In: WWW '06: Proceedings of the 15th international conference on World Wide Web, New York, NY, USA, 2006, ACM, pp. 625–632.

[5] Keerthi S. S., Lin C. J., Weng R. C.: Trust region newton method for logistic regression. Journal of Machine Learning Research, 9, 2008, pp. 627–650.

[6] Elisseeff A., Weston J.: A kernel method for multi-labelled classification. In: Advances in Neural Information Processing Systems, MIT Press, 14, 2001, pp. 681–687.

[7] Shantanu Godbole, Sunita Sarawagi: Discriminative methods for multi-labeled classification. In: Honghua Dai, Ramakrishnan Srikant, and Chengqi Zhang, editors, PAKDD, vol. **3056** of Lecture Notes in Computer Science, Springer, 2004, pp. 22–30.

[8] Hotho A., Jäschke R., Schmitz C., Stumme G.: Trend detection in folksonomies, 2006, pp. 56–70.

[9] Järvelin K., Kekäläinen J.: Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf. Syst., **20**, 4, 2002, pp. 422–446.

[10] Jäschke R., Marinho L., Hotho A., Schmidt-Thieme L., Stumme G.: Tag recommendations in folksonomies. In: Alexander Hinneburg, editor, Proceedings of LWA 2007, Sep. 2007, pp. 13–20.

[11] Joachims T.: Optimizing search engines using clickthrough data. In: KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM, 2002, pp. 133–142.

[12] Katakis I., Tsoumakas G., Vlahavas I.: Multilabel text classification for automated tag suggestion. In: Proceedings of ECML PKDD Discovery Challenge (RSDC08), 2008, pp. 75–83.

[13] Luaces J., Quevedo J. R., Taboada F., Albaiceta G. M., Bahamonde A.: Prediction of probability of survival in critically ill patients optimizing the area under the ROC curve. In: Manuela M. Veloso, editor, IJCAI, 2007, pp. 956–961.

[14] Marinho L., Schmidt-Thieme L.: Collaborative tag recommendations. In: Springer Berlin Heidelberg, editor, Studies in Classification, Data Analysis, and Knowledge Organization, 2008, pp. 533–540.

[15] Mishne G.: Autotag: a collaborative approach to automated tag assignment for weblog posts. In: Proceedings of WWW'06, ACM, New York, 2006, pp. 953–954.

[16] Platt J. C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers, MIT Press, 1999, pp. 61–74.

[17] Sigurbjörnsson B., van Zwol R.: Flickr tag recommendation based on collective knowledge. In: Proceedings of WWW '08, New York, ACM, 2008, pp. 327–336.

[18] Tsoumakas G., Katakis I.: Multi label classification: An overview. International Journal of Data Warehousing and Mining, **3**, 3, 2007, pp. 1–13.

[19] Xu Z., Fu Y., Mao J., Su D.: Towards the semantic web: Collaborative tag suggestions. In: Proceedings of WWW2006, Edinburgh, Scotland, 2006.

NEURAL NETWORK WORLD

Volume 20 2010 Number 7

CONTENTS