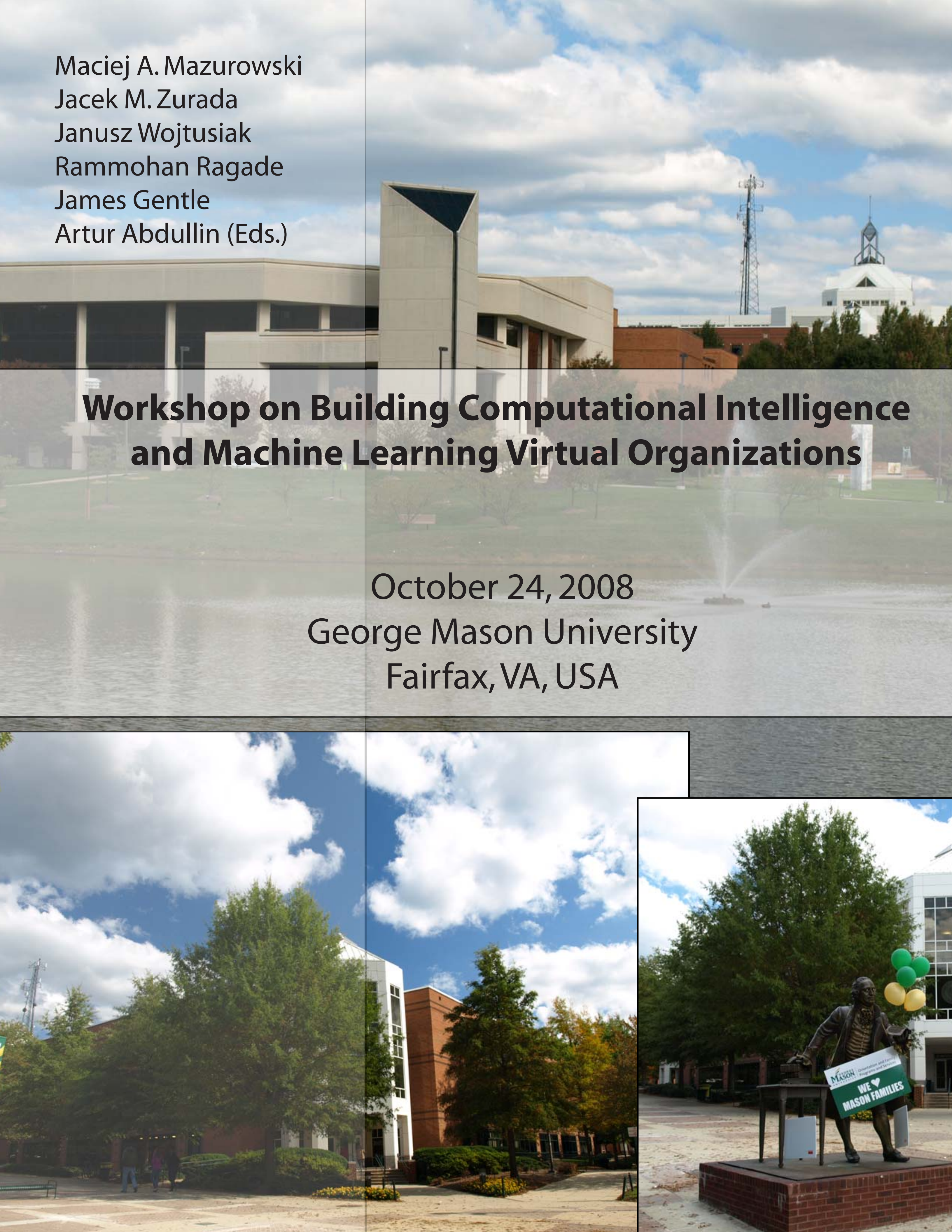


Maciej A. Mazurowski
Jacek M. Zurada
Janusz Wojtusiak
Rammohan Ragade
James Gentle
Artur Abdullin (Eds.)

Workshop on Building Computational Intelligence and Machine Learning Virtual Organizations

October 24, 2008
George Mason University
Fairfax, VA, USA

Workshop on Building Computational Intelligence and Machine Learning Virtual Organizations

Organizing committee:

General Chair:
Jacek M. Zurada

Co-Chairs:
James Gentle
Rammohan Ragade

Technical Program Chair:
Janusz Wojtusiak

Publications Chair:
Maciej A. Mazurowski

Registration and Finance Chair:
Artur Abdullin

This workshop is supported in part by the National Science Foundation Grant No. CBET 0742487. Opinions and statements expressed below neither reflect the position of the sponsoring organization nor are those of the US Government.

Workshop program

9:00 – 9:10
Opening remarks
Jacek M. Zurada (Univ. Louisville)

Session I-Chair: Dr. Rammohan Ragade (Univ. of Louisville)
9:10 – 9:30
Cyber Advancing Research/Research Advancing Cyber
Scott Midkiff (National Science Foundation)

9:30 – 9:50
Building Virtual Community in Computational Intelligence and Machine Learning
Jacek M. Zurada (Univ. Louisville)

9:50 – 10:10
Toward Multidisciplinary Collaboration in the CIML Virtual Community
Jacek M. Zurada, Janusz Wojtusiak, Maciej A. Mazurowski, Devendra Mehta, Khalid Moidu, Steve Margolis (Univ. Louisville, George Mason Univ., Orlando Health, Orlando, Florida)

10:10 – 10:30
Computational Intelligence Software for Pattern Recognition and Time Series Prediction
Patricia Melin, Oscar Castillo, Olivia Mendoza (Tijuana Institute of Technology, UABC Univ. Tijuana, Mexico)

10:30 – 10:50 Coffee break

Session II-Chair: Dr. Jacek M. Zurada (Univ. of Louisville)
10:50 – 11:10
Virtual Communities-Large Scale Human-Computer Networks
Robert Kozma, Marko Puljic (Univ. of Memphis, US Air Force Research Laboratory)

11:10 – 11:30
Comparison of Learning Algorithms: Pitfalls and Challenges
Vladimir Cherkassky (Univ. Minnesota)

11:30 – 11:50
CI^2 – SBE: CyberInfrastructure (CI) and Computational Intelligence (CI) Involvement in Social, Behavioral and Economic (SBE) Sciences
Fahmida Chowdhury (National Science Foundation)

11:50 – 12:50 Lunch break

Session III-Chairs: Drs. James Gentle, Janusz Wojtusiak (George Mason Univ.)
12:50 – 1:10
Collaborative Experimentation Using Agent-based Simulation
Jan D. Gehrke (Univ. Bremen, Germany)

1:10 – 1:30
Development of the Intelligent Sensor Network Anomaly Detection System: Problems and Solutions
Leon Reznik and Carl Hoffman (Rochester Institute of Technology)

1:30 – 1:50
Conservation of Information (COI). A Concept Paper on Virtual Organizations and Communities
William F. Lawless, Donald A. Sofge (Paine College)

1:50– 2:10
Computational Intelligence Software for Interval Type-2 Fuzzy Logic
Oscar Castillo, Patricia Melin, Juan R. Castro (Tijuana Institute of Technology, Mexico)

2:10 – 2:30
Workflow Considerations in the Emerging CIML-Virtual Organization
Chris Boyle, Artur Abdullin, Maciej A. Mazurowski, Janusz Wojtusiak, Rammohan Ragade, Jacek M. Zurada (Univ. Louisville, George Mason Univ.)

2:30 – 3:20 Coffee break

2:30-3:20 Session IV-Round-Table Discussion *CI-ML VO for Education and Interdisciplinary Cooperation*, Moderator: Dr Jacek M. Zurada (Univ. Louisville)

Session V-Chairs: Drs. Mo Jamshidi (Univ. Texas San Antonio), Robert Kozma (Univ. Memphis)
3:20 – 3:40
Application Testing of Novel Neural Network Structures
Leon Reznik, Gregory Von Pless, and Tayeb Al Karim (Rochester Institute of Technology)

3:40 – 4:00
A DEVS-JAVA-XML Simulation and Implementation of a System of Rovers for Threat Detection
Mo Jamshidi (Univ.Texas San Antonio)

4:00 – 4:20
Computational Challenges in Modeling, Control and Optimization in Electric Power and Energy Systems
Ganesh K. Venayagamoorthy (Missouri Univ. of Science and Technology)

4:20 – 4:40
Demonstration and Application of Rule Discovery Methods Using iAQ
Jaroslaw Pietrzykowski (George Mason Univ.)

4:40 – 5:00
A Bayesian Machine Learning Method for Sensor Selection and Fusion with Application to On-Board Fault Diagnostics
Niranjan Subramanyah, Yung C. Shin, Peter H. Meckl (Purdue Univ.)

5:00 – 5:20
Knowledge Discovery in Date with Selected Java Open Source Software
Carlos Rojas, Olfa Nasraoui, Nurcan Durak, Leyla Zuhadar, Sofiane Sellah, Zhiyong Zhang, Basheer Hawwash (Univ. Louisville)

Table of contents

A Bayesian Machine Learning Method for Sensor Selection and Fusion with Application to On-Board Fault Diagnostics

Niranjan Subrahmanya
School of Mechanical Engineering
Purdue University
West Lafayette, Indiana 47907, U.S.A.
nsubrahm@purdue.edu

Yung C. Shin
School of Mechanical Engineering
Purdue University
West Lafayette, Indiana 47907, U.S.A.
shin@purdue.edu

Peter H. Meckl
School of Mechanical Engineering
Purdue University
West Lafayette, Indiana 47907, U.S.A.
meckl@purdue.edu

Abstract - — **In applications like feature-level sensor fusion, the problem of selecting an optimal number of sensors can lead to reduced maintenance costs and the creation of compact online databases for future use. This problem of sensor selection can be reduced to the problem of selecting an optimal set of groups of features during model selection. This is a more complex problem than the problem of feature selection, which has been recognized as a key aspect of statistical model identification. This work proposes a new algorithm based on the use of a recently proposed Bayesian framework for the purpose of selecting groups of features during regression and classification. The hierarchical Bayesian formulation introduces grouping for the parameters of a generalized linear model and the model hyper-parameters are estimated using an empirical Bayes procedure. A novel aspect of the algorithm is its ability to simultaneously perform feature selection within groups to reduce over-fitting of the data. Further, the parameters obtained from this algorithm can be used to obtain a rank-order among the selected sensors. The performance of the algorithm is then tested by using diesel engine data for fault detection (43 variables, 8-classes, 30000 records) and comparing the misclassification rates with a varying number of sensors.**

I. Introduction

Condition monitoring is gaining increased attention today in various fields such as optimization of automated systems [1], maintenance of structures [2] and on-board vehicle diagnostics. The difficulty in obtaining precise mathematical models for many uncertain non-linear industrial systems has led to the increasing role of data-based condition monitoring schemes. With the availability of a wide range of accurate sensors, the main research in condition monitoring is now focused on the processing of information obtained from these sensors. Multi-sensor data fusion seeks to increase accuracy by exploiting complementary information, while at the same time increase reliability by exploiting the redundancy provided by different sensors. Many attempts towards achieving this goal can be found in literature [3-9].

Feature-level sensor fusion involves the extraction of features by processing the raw sensor data using various signal processing methods and then using these features to develop a suitable model. It is expected that with suitable processing, the features are less noisy and contain more information about the condition of interest. A schematic of feature-level sensor fusion is shown in Figure 1. In applications like on-board diagnostics, it is possible to collect data in real time from the vehicles to monitor their conditions and also to transfer and store this data in a centralized database for future use by other vehicles. In such situations, the selection of a compact set of sensors and feature, which completely represent the fault signature, is important to reduce the amount of data being transmitted as well as to achieve efficient storage. Therefore, in order to design a feature level sensor fusion system, the following problems have to be addressed.

1. Selection of a minimal number of sensors without compromising performance. This reduces the cost of installation and simplifies the maintenance of sensors.
2. Selection of the best subset of features from the selected sensors. This ensures good generalization and also lessens the signal processing and data transfer burden in real time monitoring.
3. Training a model from data to predict the process condition

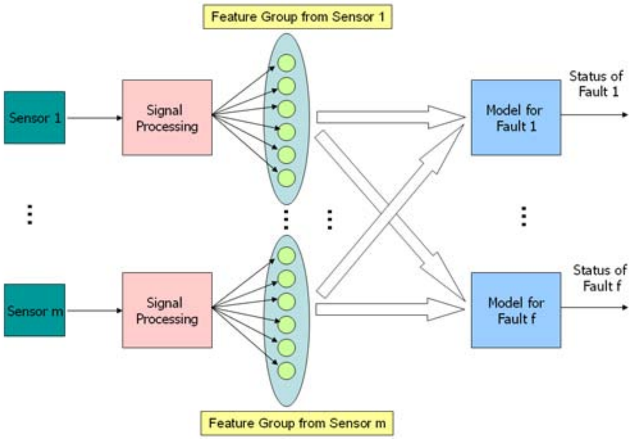


Figure 1. Sensor Selection as Feature Group Selection.

The above figure shows that selecting even a single feature from a particular sensor's feature group to model any of the faults will necessitate selecting the corresponding sensor.

Unfortunately, all the 3 problems stated above are interlinked as the evaluation of each stage is dependent on other stages. One way to overcome this problem is to group all the features belonging to a sensor together, which reduces the problem of sensor selection to the problem of feature group selection. The problem of feature group selection in fact has a larger scope with potential application in many new fields.

Although a significant amount of research in the area of feature selection may be found in the literature [10], relatively few works on feature group selection are available. While it is possible to come up with certain straightforward extensions of feature selection strategies to feature group selection, they usually do not perform well especially as the number of features in each group becomes large. An extension of the popularly used filter techniques for feature selection based on correlation [10] or mutual information [10] to group feature selection is severely affected in such cases. As a result, recently there have been some attempts made at extending embedded feature selection methods [10, 11] to group feature selection. Specifically, for models which are linear in parameter (models of the form $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$), assuming the feature values to be normalized, the model parameters **w** may be considered as scaling factors as each feature is multiplied by the corresponding component of **w** to get the term $\mathbf{w}^T\mathbf{x}$, making the magnitude of $\mathbf{w}_i$ indicative of the importance of the i^{th} feature. The essence of embedded selection methods lies in the design of penalty terms for **w**, which induce sparsity in terms of groups in the final parameter estimates.

Many such approaches, which are suitable for linear models, have been proposed to date [12-17]. A particular approach that has been gaining popularity lately is the "group lasso", which generalizes the lasso penalty [18] for feature selection to group feature selection. This was originally proposed in [16] as a solution to variable selection for regression when categorical variables also need to be considered. It has since been used for grouping of variables in many applications such as feature selection for multi-output regression [14], micorarray data analysis [19] and for logistic regression [20]. It has been observed that although the group lasso results in setting the weights of many groups to zero, it does not return the smallest possible set of groups that are sufficient to obtain an accurate model [15]. In [21], a modified $\ell_1(\mathbf{w})$ penalty function using the inverse of a pseudo-adjacency graph of feature relations is used to group features together based on their proximity information extracted using SPECT (Single photon emission computed tomography) perfusion imaging [21]. All these methods approach the problem of group feature selection from a regularization point of view. Therefore they require the manual tuning of a trade-off parameter between the regularization term and the error term. Moreover, only point estimates are obtained after training.

Although fully Bayesian and analytical frameworks have been proposed for automatic feature selection [22, 23], no such attempt has been made for group feature selection. Recently, we proposed a novel model for the problem of feature group selection using a hierarchical Bayesian formulation and gave an algorithm to infer posterior distributions over the parameters and hyper-parameters using variational inferencing [24]. This algorithm brings with it the well known advantages of a fully Bayesian paradigm such as

- Automatic inference of hyper-parameters from data without cross-validation
- Good performance with small data sets
- Probability distributions (and hence confidence intervals) are obtained for the parameters as well as model output instead of point estimates.

In this paper, we follow the same problem formulation but propose a simpler algorithm based on maximizing the log-likelihood with respect to the hyper-parameters of the model. This inference scheme is known by many names such as "*type II maximum likelihood*" method, or the "*evidence for hyper-parameters*" method or the "*empirical Bayes*" method and the reason for maximizing the log-likelihood with respect to the hyper-parameters (rather than the parameters themselves) is the belief that the hyper-parameters cannot over-fit the training data [25]. Thus this method may be considered an extension of the original relevance vector machine (RVM) [22], which was designed to automatically select the most relevant basis vectors for classification and regression using generalized linear models. For group feature selection, the problem involves two stages; the first one being the selection of the most relevant groups followed by the selection the most relevant features from the selected groups. It will be shown that it is possible to incorporate these prior preferences for parameter selection by using additional hyper-parameters in the problem formulation. In this spirit we call the proposed method the Relevant Group Selector (RGS).

Section II presents the new hierarchical formulation of the prior over parameters and provides a discussion on how it reflects the requirements for group feature selection. Section III presents the hyper-parameter estimation algorithm. Results from the application of the proposed framework to the problem of sensor selection and fusion in diesel engine fault diagnostics are presented in Section IV, while the conclusions and scope for utilizing an Engineering Virtual Organization (EVO) for data standardization and software implementation to setup this algorithm are discussed in Section V.

II. Hierarchical Bayesian Formulation

Given a training set $\mathbf{D} = \{(\mathbf{x}_i, \mathbf{t}_i) \in \chi \times \{-1,1\} : i = 1,.....,n\}$ for binary classification or $\mathbf{D} = \left\{(\mathbf{x}_i, \mathbf{t}_i) \in \chi \times \mathbb{R} : i = 1,.....,n\right\}$ for regression with $\mathbf{x}_i = [x_{i1},....,x_{ip}]^T \in \chi \subset \mathbb{R}^p$, the goal of supervised learning is to learn a function, $y = f(\mathbf{x})$, which not only recalls this information but also generalizes well. Here it is assumed that the function has the structure shown below.

$$y = f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \phi(\mathbf{x}) \qquad (1)$$

where $\phi(\mathbf{x}) \in \mathbb{R}^d$ is a *d*-dimensional feature vector extracted from the input $\mathbf{x}_i$. Assume that the features of the data set are generated by different sensors. Multiple features may be extracted from each sensor. Assume that the *d* features are extracted from *m* sensors such that each feature belongs to one and only one sensor. This is not a strict requirement but makes the notation simpler. Let $\mathbf{s}_k \in \mathbb{N}^{d_k}$ denote the feature set of sensor *k*. Therefore, $\sum_{k=1}^{m} d_k = d$ and $\mathbf{s}_{k1} \cap \mathbf{s}_{k2} = \varnothing \quad \forall k1 \neq k2$. The goal is to find a good model, which uses features from only a small subset of sensors.

Hierarchical Formulation for Simultaneous Group and Feature Selection

Let $\mathbf{w} = [\mathbf{w}_1,....,\mathbf{w}_d]^T \in \mathbb{R}^d$ represent the *d*-dimensional vector of parameters. A Gaussian prior over the parameters is given by $p(\mathbf{w} \mid \lambda) = \prod_{j=1}^{d} \left(\sqrt{\lambda_j} / \sqrt{2\pi}\right) \exp\left(-\lambda_j \mathbf{w}_j^2 / 2\right)$ where λ_j is the inverse variance for the *j*th parameter with a higher value of λ_j placing a higher emphasis on sparseness. As λ_j tends to infinity the probability mass of $\mathbf{w}_j$ concentrates around zero. In order to introduce grouping information based on the sources, **w** can be divided into *m* groups, $\mathbf{w} = \left[\mathbf{w}^1,...\mathbf{w}^k,...\mathbf{w}^m\right], \mathbf{w}^k = \left\{\mathbf{w}_j : j \in \mathbf{s}_k\right\}$ and each group could have a Gaussian prior controlled by a different parameter from the set $\lambda = \left\{\lambda_k : k = 1,...,m\right\}$. However, this does not entirely reflect our requirements for simultaneous group and feature selection. Specifically, it does not allow features belonging to the same group to have different prior variances, which is essential for the framework to perform feature selection within the group. In order to achieve this, a flexible prior structure with a larger number of hyper-parameters is considered. Let λ_{1j} and λ_{2k} be two hyper-parameters that determine the inverse variance of a normal prior over parameter $\mathbf{w}_j$, which belongs to the *k*th group, as $\mathbf{w}_j \sim N\left(\mathbf{w}_j \mid 0, \left(\lambda_{1j} + \lambda_{2k}\right)^{-1}\right)$. Here, a separate hyper-parameter, λ_{1j} , is assigned to each feature while λ_{2k} is constrained to be the same for all features belonging to the *k*th group. Therefore, $\lambda_1 \in \mathbb{R}^d$ and $\lambda_2 \in \mathbb{R}^m$. This structure suitably reflects the belief that the selection of a feature, which is equivalent to having a non-zero value for the corresponding parameter, is dependent on the relevance of the feature to the prediction task (as estimated by λ_{1j}) and the relevance of the sensor to which the feature belongs (as estimated by λ_{2k}). This is because the actual variance of the prior is given by $\left(\lambda_{1j} + \lambda_{2k}\right)^{-1}$ and as either λ_{1j} or λ_{2k} tends to infinity this variance tends to zero and the parameter weight is fixed at zero. An infinite value for λ_{1j} (λ_{2k}) indicates that the corresponding feature (sensor) is irrelevant. Moreover, when λ_{2k} tends to zero, indicating that the sensor is almost definitely relevant and selected, we can see that the variance of the parameter prior tends to λ_{1j}^{-1} , which results in a prior structure for pure feature selection as in the RVM [22]. Assuming suitable conjugate hyper-priors for λ_{1j} and λ_{2k} , the overall prior and hyper-prior structure is as shown below.

$$\begin{aligned} Prior: \ p(\mathbf{w} \mid \lambda) &= \prod_{k=1}^{m} \prod_{j \in \mathbf{s}_k} N\left(\mathbf{w}_j \mid 0, \left(\lambda_{1j} + \lambda_{2k}\right)^{-1}\right) \\ &= \prod_{k=1}^{m} \prod_{j \in \mathbf{s}_k} \frac{\sqrt{\left(\lambda_{1j} + \lambda_{2k}\right)}}{\sqrt{2\pi}} \exp\left(-\frac{\left(\lambda_{1j} + \lambda_{2k}\right)\mathbf{w}_j^2}{2}\right) \end{aligned} \qquad (2)$$

$$\begin{aligned} Hyper\ Prior: \ p(\lambda_{1j} \mid a_1, b) &= \Gamma\left(\lambda_{1j} \mid a_1, b\right) \\ &= \frac{b^{\,a_1} \lambda_{1j}^{a_1 - 1} \exp\left(-b\lambda_{1j}\right)}{\Gamma\left(a_1\right)}, \ \lambda_{1j} > 0 \\ p(\lambda_{2k} \mid a_2, b) &= \Gamma\left(\lambda_{2k} \mid a_2, b\right) \\ &= \frac{b^{\,a_2} \lambda_{2k}^{a_2 - 1} \exp\left(-b\lambda_{2k}\right)}{\Gamma\left(a_2\right)}, \ \lambda_{2k} > 0 \end{aligned} \qquad (3)$$

III. Relevant Group Selector (RGS)

The algorithm for regression is presented first. The modifications required to be made to apply the algorithm for classification will be pointed out later. Assuming that observations are generated by the model and corrupted by independent identically distributed Gaussian noise with variance τ^{-1} , the conditional distribution of the target variables, $\mathbf{t} = \left\{\mathbf{t}_1,...,\mathbf{t}_n\right\}$, given the input, is

$$P(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \tau) = \prod_{i=1}^{n} N\left(\mathbf{t}_i \mid \mathbf{w}^T \phi(\mathbf{x}_i), \tau^{-1}\right) \qquad (4)$$

A conjugate hyper-prior is also assumed for the inverse noise variance.

$$P(\tau) = \Gamma\left(\tau \mid c, d\right) \qquad (5)$$

Combining the likelihood function (4) and (5) with the weight prior (2) and hyper-prior (3) the complete model specification is obtained. All the hyper-priors can be effectively made non-informative by choosing low values for the parameters of the gamma distribution (a_1 = a_2 = b = c = d = 10^{-6}). In an ideal Bayesian framework, we are interested in predicting the distribution for a new input given the available data.

$$P(\mathbf{t}_{new}\,|\,\mathbf{t}) = \int P(\mathbf{t}_{new}\,|\,\mathbf{w},\lambda_1,\lambda_2,\tau)P(\mathbf{w},\lambda_1,\lambda_2,\tau\,|\,\mathbf{t})d\mathbf{w}d\lambda_1 d\lambda_2 d\tau \tag{6}$$

It is not possible to obtain an exact solution for $P(\mathbf{w},\lambda_1,\lambda_2,\tau\,|\,\mathbf{t}) = \dfrac{P(\mathbf{t}\,|\,\mathbf{w},\lambda_1,\lambda_2,\tau)P(\mathbf{w},\lambda_1,\lambda_2,\tau)}{P(\mathbf{t})}$ analytically because of the intractability of computing the normalizing integral to obtain $P(\mathbf{t})$. Hence a decomposition of the posterior probability is made as shown below.

$$P(\mathbf{w},\lambda_1,\lambda_2,\tau\,|\,\mathbf{t}) = P(\mathbf{w}\,|\,\lambda_1,\lambda_2,\tau,\mathbf{t})P(\lambda_1,\lambda_2,\tau\,|\,\mathbf{t}) \tag{7}$$

$P(\mathbf{w}\,|\,\lambda_1,\lambda_2,\tau,\mathbf{t})$ can be calculated easily because of the Gaussian nature of the prior and the likelihood.

$$P(\mathbf{w}\,|\,\lambda_1,\lambda_2,\tau,\mathbf{t}) = (2\pi)^{-(d+1)/2}\,|\Sigma|^{-1/2}\exp\left\{-\frac{1}{2}\left(\mathbf{w}-\boldsymbol{\mu}\right)^T\Sigma^{-1}\left(\mathbf{w}-\boldsymbol{\mu}\right)\right\} \tag{8}$$

where
$$\boldsymbol{\Sigma} = \left(\tau\boldsymbol{\Phi}^T\boldsymbol{\Phi}+\mathbf{A}\right)^{-1} \tag{9}$$

$$\boldsymbol{\mu} = \tau\Sigma\boldsymbol{\Phi}^T\mathbf{t} \tag{10}$$

with $\boldsymbol{\Phi} = \left[\phi(\mathbf{x}_1)\ \ \phi(\mathbf{x}_2)\ \ \ldots\ \ \phi(\mathbf{x}_n)\right]^T$ and $\mathbf{A} = diag(\lambda_1+\lambda_2)$, i.e. , it is a diagonal matrix with the elements of the vector $\lambda_1+\lambda_2$ along its diagonal. The term $\lambda_1+\lambda_2$ is used here to denote a vector where λ_{2k} is added to each λ_{1j} for which $j\in\mathbf{s}_k$. $P(\lambda_1,\lambda_2,\tau\,|\,\mathbf{t})$ on the other hand is not as easy to evaluate and in the empirical Bayes procedure, it is replaced by a delta function at its mode, i.e., it is approximated by $\delta(\lambda_1^*,\lambda_2^*,\tau^*\,|\,\mathbf{t})$. With this approximation, we hope that

$$\begin{aligned}
P(\mathbf{t}_{new}\,|\,\mathbf{t}) &= \int P(\mathbf{t}_{new}\,|\,\mathbf{w},\lambda_1,\lambda_2,\tau)P(\mathbf{w},\lambda_1,\lambda_2,\tau\,|\,\mathbf{t})d\mathbf{w}d\lambda_1 d\lambda_2 d\tau \\
&= \int P(\mathbf{t}_{new}\,|\,\mathbf{w},\lambda_1,\lambda_2,\tau)P(\mathbf{w}\,|\,\lambda_1,\lambda_2,\tau,\mathbf{t})P(\lambda_1,\lambda_2,\tau\,|\,\mathbf{t})d\mathbf{w}d\lambda_1 d\lambda_2 d\tau \\
&= \int\left(\int P(\mathbf{t}_{new}\,|\,\mathbf{w},\lambda_1,\lambda_2,\tau)P(\mathbf{w}\,|\,\lambda_1,\lambda_2,\tau,\mathbf{t})d\mathbf{w}\right)P(\lambda_1,\lambda_2,\tau\,|\,\mathbf{t})d\lambda_1 d\lambda_2 d\tau \\
&\approx \int\left(\int P(\mathbf{t}_{new}\,|\,\mathbf{w},\lambda_1,\lambda_2,\tau)P(\mathbf{w}\,|\,\lambda_1,\lambda_2,\tau,\mathbf{t})d\mathbf{w}\right)\delta(\lambda_1^*,\lambda_2^*,\tau^*\,|\,\mathbf{t})d\lambda_1 d\lambda_2 d\tau
\end{aligned}$$

The inner integral, which marginalizes over $\mathbf{w}$ can still be computed analytically since it is the convolution of two Gaussians. Therefore, the main goal now is to find the mode of $P(\lambda_1,\lambda_2,\tau\,|\,\mathbf{t})$. This can be done using the expression below.

$$P(\lambda_1,\lambda_2,\tau\,|\,\mathbf{t}) \propto P(\mathbf{t}\,|\,\lambda_1,\lambda_2,\tau)P(\lambda_1)P(\lambda_2)P(\tau) \tag{11}$$

where

$$\begin{aligned}
P(\mathbf{t}\,|\,\lambda_1,\lambda_2,\tau) &= \int P(\mathbf{t}\,|\,\mathbf{w},\tau)P(\mathbf{w}\,|\,\lambda_1,\lambda_2)d\mathbf{w} \\
&= (2\pi)^{-n/2}\left|\tau\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T\right|^{-1/2}\exp\left\{-\frac{1}{2}\mathbf{t}^T\left(\tau\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T\right)^{-1}\mathbf{t}\right\}
\end{aligned} \tag{12}$$

In order to find the mode of $P(\lambda_1,\lambda_2,\tau\,|\,\mathbf{t})$ we maximize the log of this quantity which is given by,

$$\begin{aligned}
L &= \log P(\lambda_1,\lambda_2,\tau\,|\,\mathbf{t}) \\
&= -\frac{1}{2}\log\left|\tau\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T\right| - \frac{1}{2}\mathbf{t}^T\left(\tau\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T\right)^{-1}\mathbf{t} \\
&\quad + \sum_{j=1}^{n}\left(a_1\log\lambda_{1j} - b\lambda_{1j}\right) + \sum_{k=1}^{m}\left(a_2\log\lambda_{2k} - b\lambda_{2k}\right) + c\log\tau - d\tau \\
&\quad + \text{Constant}
\end{aligned} \tag{13}$$

The above quantity can be rewritten in terms of λ_1,λ_2,τ and $\boldsymbol{\mu}$ as

$$\begin{aligned}
L(\lambda_1,\lambda_2,\tau,\boldsymbol{\mu}) &= -\frac{1}{2}\log\left|\tau\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T\right| - \tau\left\|\mathbf{t}-\boldsymbol{\Phi}\boldsymbol{\mu}\right\|^2 \\
&\quad -\boldsymbol{\mu}^T\mathbf{A}\boldsymbol{\mu} + \sum_{j=1}^{n}\left(a_1\log\lambda_{1j} - b\lambda_{1j}\right) \\
&\quad + \sum_{k=1}^{m}\left(a_2\log\lambda_{2k} - b\lambda_{2k}\right) + c\log\tau - d\tau + \text{Constant}
\end{aligned} \tag{14}$$

Setting the partial derivative of $L(\lambda_1,\lambda_2,\tau,\boldsymbol{\mu})$ with respect to each of $(\lambda_1,\lambda_2,\tau,\boldsymbol{\mu})$ to zero and solving each set of equations in an iterative fashion gives the required method for finding the mode of $P(\lambda_1,\lambda_2,\tau\,|\,\mathbf{t})$. Once the mode is obtained, $P(\mathbf{w}\,|\,\lambda_1^*,\lambda_2^*,\tau^*,\mathbf{t})$ can then be evaluated using (8), (9) and (10) and let Σ^* and $\boldsymbol{\mu}^*$ be the covariance and mean weight matrices obtained using $\lambda_1^*,\lambda_2^*,\tau^*$. The predictions for new inputs are then made as

$$\begin{aligned}
P(\mathbf{t}_{new}\,|\,\mathbf{t}) &\approx \int P\left(\mathbf{t}_{new}\,|\,\mathbf{w},\tau\right)N\left(\mathbf{w}\,|\,\boldsymbol{\mu}^*,\Sigma^*\right)d\mathbf{w} \\
&= N\left(\mathbf{t}_{new}\,|\,\boldsymbol{\mu}^{*T}\phi(\mathbf{x}_{new}),\sigma^2\right)
\end{aligned} \tag{15}$$

$$\sigma^2 = \frac{1}{\tau^*} + \phi\left(\mathbf{x}_{new}\right)^T\Sigma^*\phi\left(\mathbf{x}_{new}\right) \tag{16}$$

For classification, the conditional distribution of the targets is given by

$$\begin{aligned}
P(\mathbf{t}\,|\,\mathbf{w}) &= \prod_{i=1}^{n}\left(1+e^{-\mathbf{t}_i\mathbf{w}^T\phi(\mathbf{x}_i)}\right)^{-1} = \prod_{i=1}^{n}\sigma\left(\mathbf{t}_i\mathbf{w}^T\phi(\mathbf{x}_i)\right) \\
&= \prod_{i=1}^{n}\sigma\left(\mathbf{z}_i\right)
\end{aligned} \tag{17}$$

where $\sigma(z) = \left(1+e^{-z}\right)^{-1}$ and $\mathbf{z}_i = \mathbf{t}_i\mathbf{w}^T\phi(\mathbf{x}_i)$. Since the prior over **w** is not conjugate to this likelihood function, it is no longer possible to evaluate $P(\mathbf{w}\,|\,\lambda_1,\lambda_2,\mathbf{t})$ analytically. In order to overcome this problem, Tipping [22] makes use of a local Gaussian approximation based on Laplace's Method to the posterior distribution of weights. This is done by numerically optimizing for $P(\mathbf{w}\,|\,\lambda_1,\lambda_2,\mathbf{t}) \propto P(\mathbf{t}\,|\,\mathbf{w})P(\mathbf{w}\,|\,\lambda_1,\lambda_2)$ to find $\boldsymbol{\mu}^*$ and then calculating $\boldsymbol{\Sigma}^*$ as

$$\boldsymbol{\Sigma}^* = \left(\nabla_{\mathbf{w}}\nabla_{\mathbf{w}}\log P(\mathbf{w}\,|\,\lambda_1,\lambda_2,\mathbf{t})\Big|_{\boldsymbol{\mu}^*}\right)^{-1} = \left(\boldsymbol{\Phi}^T\mathbf{B}\boldsymbol{\Phi}+\mathbf{A}\right)^{-1} \tag{18}$$

where $\mathbf{B} = diag\{\mathbf{B}_1, \mathbf{B}_2, ..., \mathbf{B}_n\}$ and $\mathbf{B}_i = \sigma\left(\boldsymbol{\mu}^{*T}\phi(\mathbf{x}_i)\right)\left[1-\sigma\left(\boldsymbol{\mu}^{*T}\phi(\mathbf{x}_i)\right)\right]$.

The prediction for new data can be obtained by using the posterior mean weights in (17). Therefore, in order to complete the algorithm, it is necessary to obtain the expressions for the partial derivative of $L(\lambda_1, \lambda_2, \tau, \boldsymbol{\mu})$ with respect to each of $(\lambda_1, \lambda_2, \tau, \boldsymbol{\mu})$. It is generally more convenient to obtain the partial derivative of $L(\lambda_1, \lambda_2, \tau, \boldsymbol{\mu})$ with respect to the log of the hyper-parameters $(\lambda_1, \lambda_2, \tau)$ and setting this to zero is equivalent from the perspective of maximizing L. The partial derivatives are presented below without derivations, which have been skipped to conserve space.

$$\frac{\partial L}{\partial \log \lambda_{1j}} = \frac{\lambda_{1j}}{2(\lambda_{1j}+\lambda_{2k})} - \frac{\lambda_{1j}\Sigma^*_{jj}}{2} - \frac{\lambda_{1j}\mu^{*2}_j}{2} + a_1 - b\lambda_{1j} \tag{19}$$

$$\frac{\partial L}{\partial \log \lambda_{2k}} = \sum_{j\in \mathbf{s}_k}\left(\frac{\lambda_{2k}}{2(\lambda_{1j}+\lambda_{2k})} - \frac{\lambda_{2k}\Sigma^*_{jj}}{2} - \frac{\lambda_{2k}\mu^{*2}_j}{2}\right) + a_2 - b\lambda_{2k} \tag{20}$$

$$\frac{\partial L}{\partial \log \tau} = \frac{1}{2}\left(n - \tau(\mathbf{t}-\boldsymbol{\Phi\mu})^T(\mathbf{t}-\boldsymbol{\Phi\mu}) - trace\left(\boldsymbol{\Sigma}^*\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)\right) + c - d\tau \tag{21}$$

$$\frac{\partial L}{\partial \boldsymbol{\mu}} = -2\tau\boldsymbol{\Phi}^T(\mathbf{t}-\boldsymbol{\Phi\mu}) + 2\mathbf{A}\boldsymbol{\mu} \tag{22}$$

Each of the above expressions can be set to zero and solved for the corresponding variables except for (20), which has to be solved numerically (a one-dimensional binary search technique is used to locate the root). From an implementation perspective, it is essential to solve (20) first as given in the algorithm below. The final algorithm which makes use of these solutions is given below.

Algorithm 1. *Relevant Group Selection*

```
Initialization: Set $\lambda_{1j}, \lambda_{2k}$ values to 0.5. Set $\boldsymbol{\mu}$ to maximum likelihood solution.
Set $\tau$ for regression.
repeat (until parameters converge)
    Solve (20) numerically using binary search and update $\lambda_{2k}$ for $k = 1, 2..., m$.
    Update $\lambda_{1j}$ values to equal the positive root of quadratic expression (19) for $j = 1, 2..., d$.
    Update $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ using (9) and (10)
    $\tau = \frac{n - trace\left(\boldsymbol{\Sigma\Phi}^T\boldsymbol{\Phi}\right) + 2c}{(\mathbf{t}-\boldsymbol{\Phi\mu})^T(\mathbf{t}-\boldsymbol{\Phi\mu}) + 2d}$
end
return $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \tau, \lambda_1$ and $\lambda_2$
```

Finally, although the entire explanation has been provided for the case of a single output, it is easy to consider multi-class classification by using a one-versus-all [26] approach and grouping all the parameters corresponding to features from a single sensor together.

IV. Experimental Results

The above algorithm is applied on a real world data set acquired from a set of 16 cylinder diesel engines used in mining applications. The data includes 30000 samples acquired from forty three temperature and pressure sensors connected to engine control units and sampled at 1 Hz at various operating points. The list of variables monitored on the engine is given in Table 1. The engine is assumed to be in eight different states, the first one being healthy and classes two to eight denoting various kinds of faults such as high crankcase pressure, low oil pressure, high intake manifold temperature and so on. More details about the sensors and data acquisition procedure may be found in [27].

Table 1 List of Variables Monitored On Diesel Engine

Variable	Description	Units
Y	Class Number. 1 – Healthy 2 to 8 – Faulty	-
u_1	Brake Horse Power	Bhp
u_2	ECM Temperature	C (F)
u_3	% Acceleration Pedal	%
u_4	Instantaneous Engine Load	%
u_5	Oil Filter Differential Pressure	kPa (psig)
u_6	Post Filter Oil Pressure	kPa (psig)
u_7	Pre Filter Oil Pressure	kPa (psig)
u_8	Oil Rifle Pressure	kPa (psig)
$u_9 - u_{10}$	LB/RB Boost Pressures	kPa (psig)
$u_{11} - u_{14}$	All Banks IMT	C (F)
u_{15}	Coolant Pressure	kPa (psig)
u_{16}	Coolant Temperature	C (F)
u_{17}	Rail Pressure	kPa (psig)
u_{18}	Battery Voltage	V
u_{19}	LBR Compressor Inlet Temperature	C (F)
$u_{20} - u_{35}$	EGT for 16-Cylinders	C (F)
u_{36}	Avg. Exhaust Temperature	C (F)
u_{37}	Engine Oil Temperature	C (F)
u_{38}	Engine Speed	Rpm
u_{39}	Timing Pressure	kPa (psig)
u_{40}	Fuel Temperature	C (F)
$u_{41} - u_{42}$	LB/RB Avg. Exhaust Temperatures	C (F)
u_{43}	Crankcase Pressure	in of H_20

Y – Output, u – Inputs, LB – Left Bank, RB – Right Bank, F – Front, R – Rear, IMT – Intake Manifold Temperature, EGT – Exhaust Gas Temperature

The performance of RGS is compared to two existing methods: the group lasso [16] and the RVM. The group lasso has been used in many recent works and requires the tuning of a tradeoff parameter between a penalty term of the form $\sum_{k=1}^{m}\sqrt{\sum_{j\in\mathbf{s}_k}\mathbf{w}_j^2}$ and the training error. The RVM is used as a baseline measure as it only considers feature selection without grouping information. In addition to this the results from [28] based on using an information theoretic criterion are also presented for comparison purposes.

Four features were extracted from each sensor signal namely the signal itself, the square of the signal, the cube of the signal and the logarithm of the signal. This allows the classification boundary to be slightly non-linear without imposing a significant computational burden for this large dataset. The data set was standardized so that each feature has zero mean and unit standard deviation and a bootstrap sample size of 167, with 180 data points being held out in each sample, was used to estimate the performance of the different algorithms (this is the procedure recommended and followed in [28] based on the data collection procedure),. The results are given in the table below. The RGS outperforms the Group Lasso both in terms of classification accuracy and the number of sensors selected. The difference in the prediction accuracy is not very high since very few features were extracted from each sensor and the total number of features is anyhow small compared to the large number of training data points available. The RVM has a slightly better prediction accuracy but at the cost of using a significantly larger number of sensors. Therefore considering the overall tradeoff between the cost of sensor installation and maintenance and the prediction accuracy, the results from RGS could be considered “optimal”.

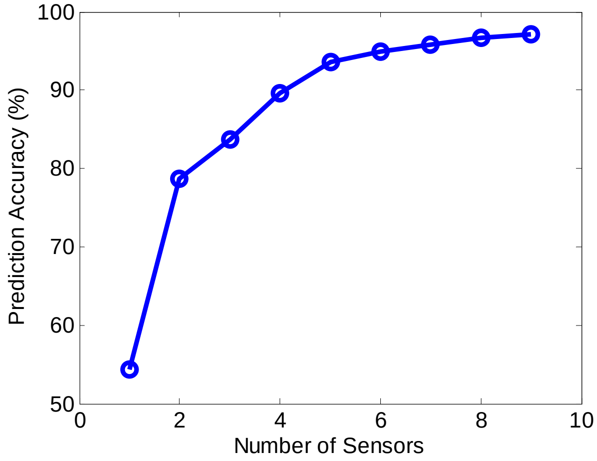


Figure 2. Prediction accuracy versus number of sensors selected according to the ranking returned by RGS

TABLE 2 GROUP SELECTION RESULTS FOR DIESEL ENGINE FAULT DETECTION (THE TABLE PRESENTS RESULTS OBTAINED USING 167 BOOTSTRAP SAMPLES)

RGS		Group Lasso		RVM	
Accuracy (%)	Groups	Accuracy (%)	Groups	Accuracy (%)	Groups
97.11	**9**	96.79	15	98.32	21

In the next part of the analysis, the rank-order of sensors is determined using the sum of the weights of the features belonging to each sensor, i.e., by sorting the sensor according to $Weight(k)=\sum_{c=1}^{f}\sum_{j\in\mathbf{s}_k}\left|\mathbf{w}_j^c\right|$, where f is the number of outputs. From Table 2 it can be seen that the RGS has already narrowed down the number of sensors to 9. The ranking among these sensors, in decreasing order of importance, was found to be – 19, 2, 16, 11, 29, 40, 27, 18 and 10 (these correspond to the input number from Table 1). It is now possible to check the performance of the diagnostic system using only the desired top sensors by running the RGS again on this restricted data set. The results obtained are plotted in Fig 2. A prediction accuracy of 89.66% was obtained with the top 4 sensors and a prediction accuracy of 96.68% was obtained with the top 8 sensors. Comparing this to the best accuracy of 89.7% with 8 sensors reported in [28], it can be seen that performing simultaneous sensor selection and model development is indeed beneficial to improving the performance of the system. The simpler linear in parameter model chosen here should also be easier to implement and update for on-board diagnostics. The top 8 sensors selected in [28] are almost entirely different (except for sensor 19), which could also explain the significant difference in prediction accuracy with 9 sensors obtained by the RGS.

V. DISCUSSION ON THE ISSUES RELATED TO CIML VIRTUAL ORGANIZATION

Although the paper has so far focused on the application of the proposed algorithm to engine diagnostics, it is easy to see that the generic nature of the algorithm itself makes it applicable to a broad range of problems where grouping of parameters is of interest. For example, the algorithm can be directly applied to other important problems such as sensor selection in wireless sensor networks, pathway selection in microarray data analysis (where groups of genes have predefined roles), multi-task compressive sensing, region (group of pixels) selection in imaging applications, bandwidth selection in spectroscopic applications and so on. An EVO would be an ideal place for experts from different application communities to come together and discuss the potential uses of this tool to solve problems in their specific areas.

In order to create such a tool, which is readily applicable to different domains, the software development aspects should be given significant consideration, especially for modules designed around the proposed algorithm, which have to be tailored to take into account the idiosyncrasies of individual application domains. Specifically, it is important to address the following issues.

- *Standardization of input data format.* This should include input regarding grouping of parameters.
- *Creation of a centralized database.* It should be possible for different users of a specific domain to upload data to this data-based and also exploit knowledge gained from data already existing in the data-base.
- *Security and validation of input data.* Only authorized users should be allowed access to the data-base and even then the submitted data has to be validated.
- *Real time communication.* The system should also allow the direct uploading of data from units in the field and in turn provide updated models in real time.
- *Interface design.* The Bayesian paradigm allows the extraction of a large amount of information such as the parameter estimates and uncertainty, prediction uncertainty, sufficiency of currently available data, trade-off between using additional features/groups and model accuracy and so on. It is essential to display all the above information in a nice graphical format without overwhelming the non-epxert user.
- *Algorithm implementation.* For large scale systems, it might be necessary to consider efficient implementations on the server using parallel programming techniques.
- *Knowledge validation.* While the proposed tool can be used to extract significant correlations and relationships from the database, it is important to have the semantics of these relationships validated by experts in the field before accepting them as process knowledge.

Discussion among the members of the EVO could provide significant inputs regarding the above issues as well as some new ones.

VI. Conclusions

This work presents a simple algorithm based on the empirical Bayes method for estimating the hyper-parameters in a parameter-free, fully Bayesian framework for simultaneous sensor and feature selection. The algorithm presented here is simpler to implement than the Variational Bayesian algorithm presented in [24]. It is also applicable in various other fields like bio-informatics and spectroscopic analysis where grouping of inputs during data mining is useful. The experimental results presented in this paper show that the proposed framework effectively manages to achieve its goal of selecting a sparse number of feature groups while simultaneously learning a model that generalizes well. Moreover, additional information regarding the rank-order of the sensors can be attained by ranking the sensors according to the sum of the weights of the features belonging to each sensor. This information can then be used to further reduce the number of sensors while sacrificing performance accuracy only slightly. The reduction in the number of sensors and features will prove to be useful for remotely monitoring systems as it reduces the data transfer and on-line processing required to convey information about the health of the system. It will also help in the creation of compact databases and allow the timely updating of models using a small set of parameters. The performance of the algorithm with only 4 sensors selected this way is still comparable to the results reported in [28].

References

[1] S. Y. Liang, R. L. Hecker, and R. G. Landers, “Machining Process Monitoring and Control: The State-of-the-Art,” *Journal of Manufacturing Science and Engineering*, vol. 126, no. 2, pp. 297-310, 2004.

[2] K. Worden, and J. M. Dulieu-Barton, “An Overview of Intelligent Fault Detection in Systems and Structures,” *Structural Health Monitoring*, vol. 3, no. 1, pp. 85-98, March 1, 2004, 2004.

[3] Y. Peng, “Intelligent condition monitoring using fuzzy inductive learning,” *Journal of Intelligent Manufacturing*, vol. 15, no. 3, pp. 373-380, 2004.

[4] M. Azam, K. Pattipati, and A. Patterson-Hine, "Optimal sensor allocation for fault detection and isolation," *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics.* pp. 1309-1314.

[5] L. B. Jack, and A. K. Nandi, “Genetic algorithms for feature selection in machine condition monitoring with vibration signals,” *IEE Proceedings: Vision, Image and Signal Processing*, vol. 147, no. 3, pp. 205-212, 2000.

[6] Q. Liu, X. Chen, and N. Gindy, “Fuzzy pattern recognition of AE signals for grinding burn,” *International Journal of Machine Tools and Manufacture*, vol. 45, no. 7-8, pp. 811-818, 2005.

[7] C. Giraud, and B. Jouvencel, "Sensor selection in a fusion process: A fuzzy approach," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems.* pp. 599-606.

[8] H. R. Berenji, J. Ametha, and D. Vengerov, "Inductive learning for fault diagnosis," *IEEE International Conference on Fuzzy Systems.* pp. 726-731.

[9] L. Wang, E. Kannatey-Asibu Jr., and M. G. Mehrabi, “A method for sensor selection in reconfigurable process monitoring,” *Journal of Manufacturing*

Science and Engineering, Transactions of the ASME, vol. 125, no. 1, pp. 95-99, 2003.

[10] I. Guyon, S. Gunn, M. Nikravesh *et al.*, *Feature Extraction, Foundations and Applications*, New York: Physica-Verlag, Springer, 2006.

[11] I. Guyon, and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of Machine Learning Research*, vol. 3, pp. 1157-1182, March 2003, 2003.

[12] T. N. Lal, M. Schroder, T. Hinterberger *et al.*, “Support vector channel selection in BCI,” *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1003-1010, 2004.

[13] Y. Kim, J. Kim, and Y. Kim, “Blockwise Sparse Regression,” *Statistica Sinica*, vol. 16, pp. 375-390, 2006.

[14] T. Similä, and J. Tikka, “Input selection and shrinkage in multiresponse linear regression,” *Computational Statistics and Data Analysis*, vol. 52, pp. 406-422, 2007.

[15] L. Wang, G. Chen, and H. Li, “Group SCAD regression analysis for microarray time course gene expression data,” *Bioinformatics*, vol. 23, no. 12, pp. 1486-1494, 2007.

[16] M. Yuan, and Y. B. Lin, “Model selection and estimation in regression with grouped variables,” *Journal of Royal Statistical Society*, vol. 68, pp. 49–67, 2006.

[17] P. Zhao, G. Rocha, and B. Yu, “Grouped and hierarchical model selection through composite absolute penalties,” *Technical Report, University of California.*, 2006.

[18] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B*, vol. 58, no. 1, pp. 267-288, 1996.

[19] S. Ma, X. Song, and J. Huang, “Supervised group Lasso with applications to microarray data analysis,” *Bioinformatics*, vol. 8, no. 60, 2007.

[20] L. Meier, S. van de Geer, and P. Buhlmann, “The group lasso for logistic regression,” *Technical Report, Eidgenössische Technische Hochschule.*, 2006.

[21] J. Stoeckel, and G. Fung, "SVM feature selection for classification of SPECT images of Alzheimer's disease using spatial information." p. 8 pp.

[22] M. E. Tipping, “Sparse Bayesian Learning and the Relevance Vector Machine,” *Journal of Machine Learning Research*, vol. 1, no. 3, pp. 211-244, 2001.

[23] C. M. Bishop, and M. E. Tipping, "Variational Relevance Vector Machines," *Uncertainty in Artificial Intelligence*, C. Boutilier and M. Goldzmidt, eds., pp. 46-53: Morgan Kaufmann, 2000.

[24] N. Subrahmanya, and Y. C. Shin, “A Variational Bayesian Framework for Group Feature Selection,” *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. (submitted), 2008.

[25] D. J. C. McKay, “Bayesian Interpolation,” *Neural Computation*, vol. 4, no. 3, pp. 415-447, 1992.

[26] B. Scholkopf, and A. J. Smola, *Learning with Kernels*, Cambridge, MA: MIT Press, 2002.

[27] A. A. Joshi, P. H. Meckl, G. B. King *et al.*, “Information-Theoretic Sensor Subset Selection: Application to Signal-Based Fault Isolation in Diesel Engines,” in Proceedings of IMECE2006, Chicago, Illinois, USA, 2006.

[28] A. A. Joshi, S. M. James, P. H. Meckl *et al.*, “Information-Theoretic Feature Selection for Classification,” in Proceedings of the 2007 American Control Conference, New York City, USA, 2007.

Computational Intelligence Software for Interval Type-2 Fuzzy Logic

Oscar Castillo, Patricia Melin, Juan R. Castro

Abstract— **A software tool for interval type-2 fuzzy logic is presented in this paper. The software tool includes a graphical user interface for construction, edition and observation of the fuzzy systems. The Interval Type-2 Fuzzy Logic System Toolbox (IT2FLS), is an environment for interval type-2 fuzzy logic inference system development. Tools that cover the different phases of the fuzzy system design process, from the initial description phase, to the final implementation phase, are part of the Toolbox. The Toolbox's best qualities are the capacity to develop complex systems and the flexibility that enables the user to extend the availability of functions for working with the use of type-2 fuzzy operators, linguistic variables, interval type-2 membership functions, defuzzification methods and the evaluation of Interval Type-2 Fuzzy Inference Systems.**

I. INTRODUCTION

The fuzzy sets were presented by L.A. Zadeh in 1965 [1, 2] to process / manipulate data and information affected by unprobabilistic uncertainty / imprecision. These were designed to mathematically represent the vagueness and uncertainty of linguistic problems; thereby obtaining formal tools to work with intrinsic imprecision in different type of problems; it is considered a generalization of the classic set theory.

Intelligent Systems based on fuzzy logic are fundamental tools for nonlinear complex system modeling. Fuzzy sets (of type-1) and fuzzy logic are the basis for fuzzy systems, where their objective has been to model how the brain manipulates inexact information. Type-2 fuzzy sets are used for modeling uncertainty and imprecision in a better way. These type-2 fuzzy sets were originally presented by Zadeh in 1975 and are essentially "fuzzy fuzzy" sets where the fuzzy degree of membership is a type-1 fuzzy set [4, 6]. The new concepts were introduced by Mendel and Liang [8, 10] allowing the characterization of a type-2 fuzzy set with a superior membership function and an inferior membership function; these two functions can be represented each one by a type-1 fuzzy set membership function. The interval between these two functions represents the footprint of uncertainty (FOU), which is used to characterize a type-2 fuzzy set. Recently, there has been a lot of applications of type-2 fuzzy systems and for this reason we considered the need of building a software tool that could ease the development of type-2 fuzzy systems for real-world problems.

II. INTERVAL TYPE-2 FUZZY SET THEORY

A. Type-2 Fuzzy Sets Concept

A type-2 fuzzy set [6,7] expresses the non-deterministic truth degree with imprecision and uncertainty for an element that belongs to a set. A type-2 fuzzy set denoted by $\overset{\approx}{A}$, is characterized by a membership function $\mu_{\overset{\approx}{A}}(x,u)$, where x∈X (universe), u $\in J_x^u \subseteq [0,1]$ and $0 \le \mu_{\overset{\approx}{A}}(x,u) \le 1$ defined by either of the equivalent expressions in equation (1).

$$\overset{\approx}{A} = \{(x,\mu_{\overset{\approx}{A}}(x)) \mid x \in X\}$$

$$\overset{\approx}{A} = \{(x,u,\mu_{\overset{\approx}{A}}(x,u)) \mid \forall x \in X, \forall u \in J_x^u \subseteq [0,1]\} \qquad (1)$$

An example of a type-2 membership function constructed with the IT2FLS toolbox is composed by a Pi primary and a Gbell secondary type-1 membership functions, these are depicted in Figure 1.

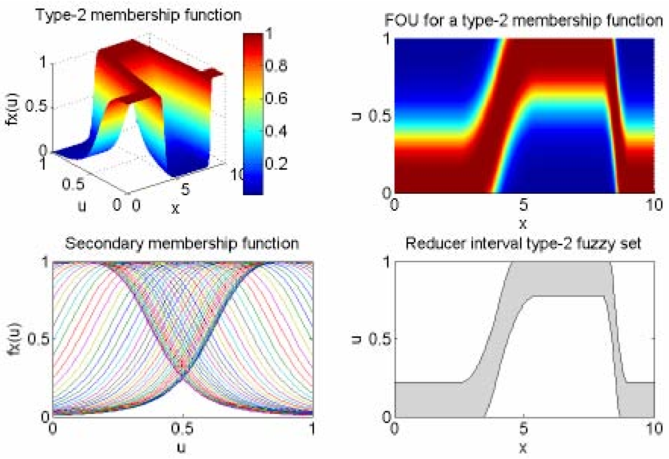


Fig. 1. FOU for Type-2 Membership Functions.

If $\overset{\approx}{A}$ is continuous it is denoted in equation (2).

$$\overset{\approx}{A} = \left\{ \int_{x\in X} \left[\int_{u\in J_x^u \subseteq [0,1]} f_x(u)/u \right] / x \right\} \qquad (2)$$

where $\iint$ denotes the union of x and u. If $\overset{\approx}{A}$ is discrete then it is denoted by equation (3).

Oscar Castillo is with the Division of Graduate Studies and Research in Tijuana Institute of Technology, Mexico (corresponding author phone: 52664-623-6318; fax: 52664-623-6318; e-mail: ocastillo@tectijuana.mx).

Patricia Melin is with the Division of Graduate Studies and Research in Tijuana Institute of Technology, Mexico (corresponding author phone: 52664-623-6318; fax: 52664-623-6318; e-mail: pmelin@tectijuana.mx)

$$\overset{\approx}{A}=\left\{\sum_{x\in X}\mu_{\overset{\approx}{A}}(x)/x\right\}=\left\{\sum_{i=1}^{N}\left[\sum_{k=1}^{M_i}f_{x_i}(u_{ik})/u_{ik}\right]/x_i\right\} \quad (3)$$

where $\sum\sum$ denotes the union of x and u.

If $f_x(u)=1, \forall u\in[\underline{J}_x^u,\overline{J}_x^u]\subseteq[0,1]$, the type-2 membership function $\mu_{\overset{\approx}{A}}(x,u)$ is expressed by one type-1 inferior membership function, $\underline{J}_x^u\equiv\underline{\mu}_A(x)$ and one type-1 superior, $\overline{J}_x^u\equiv\overline{\mu}_A(x)$ (Fig. 2), then it is called an interval type-2 fuzzy set [8] denoted by equations (4) and (5).

$$\overset{\approx}{A}=\begin{cases}(x,u,1)\,|\,\forall x\in X, \\ \qquad \forall u\in[\underline{\mu}_A(x),\overline{\mu}_A(x)]\subseteq[0,1]\end{cases} \quad (4)$$

or

$$\overset{\approx}{A}=\left\{\int_{x\in X}\left[\int_{u\in[\underline{J}_x^u,\overline{J}_x^u]\subseteq[0,1]}1/u\right]/x\right\}$$
$$=\left\{\int_{x\in X}\left[\int_{u\in[\underline{\mu}_A(x),\overline{\mu}_A(x)]\subseteq[0,1]}1/u\right]/x\right\} \quad (5)$$

If $\overset{\approx}{A}$ is a type-2 fuzzy Singleton, the membership function is defined by equation (6).

$$\mu_{\overset{\approx}{A}}(x)=\begin{cases}1/1 & si\ x=x' \\ 1/0 & si\ x\neq x'\end{cases} \quad (6)$$

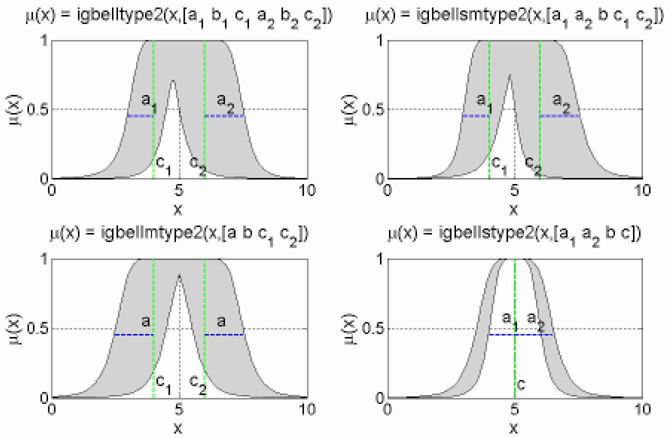


Fig. 2. FOU for Gbell Primary Interval Type-2 Membership Functions.

B. Fuzzy Set Operations

We can apply operators to the fuzzy sets, or we can make some operations between them [4, 10, 11]. When we apply an operator to one fuzzy set we obtain another fuzzy set; by the same manner when we combine an operation with two or more sets we obtain another fuzzy set. If we have two type-2 fuzzy sets identified by the letters $\overset{\approx}{A}$ and $\overset{\approx}{B}$, associated to a linguistic variable, we can define three basic operations: negation or complement, union and intersection (TABLE I).

TABLE I
Interval Type-2 Fuzzy Set Operations.

Name	Operator	Operation
Union	$\int$ = join	$\overset{\approx}{A}\int\overset{\approx}{B}=\{\int_{x\in X}\mu_{\overset{\approx}{A}}(x)\int\mu_{\overset{\approx}{B}}(x)\}$ $=\left\{\int_{x\in X}\left[\int_{\alpha\in[\underline{\mu}_{\tilde{A}}(x)\vee\underline{\mu}_{\tilde{B}}(x),\overline{\mu}_{\tilde{A}}(x)\vee\overline{\mu}_{\tilde{B}}(x)]}1/\alpha\right]/x\right\}$
Intersection	$\sqcap$ = meet	$\overset{\approx}{A}\sqcap\overset{\approx}{B}=\{\int_{x\in X}\mu_{\overset{\approx}{A}}(x)\sqcap\mu_{\overset{\approx}{B}}(x)\}$ $=$ $\left\{\int_{x\in X}\left[\int_{\alpha\in[\underline{\mu}_{\tilde{A}}(x)\wedge\underline{\mu}_{\tilde{B}}(x),\overline{\mu}_{\tilde{A}}(x)\wedge\overline{\mu}_{\tilde{B}}(x)]}1/\alpha\right]/x\right\}$
Negation	$\neg$	$\neg\overset{\approx}{A}=\left\{\int_{x\in X}\mu_{\overset{\approx}{A}}(x)/x\right\}$ $=\left\{\int_{x\in X}\left[\int_{\alpha\in[1-\overline{\mu}_{\tilde{A}}(x),1-\underline{\mu}_{\tilde{A}}(x)]}1/\alpha\right]/x\right\}$

C. Fuzzy Inference System

The human knowledge is expressed in fuzzy rules with the following form:

IF <*fuzzy proposition*> **THEN** <*fuzzy proposition*>

The fuzzy propositions are divided in two types, the first one is called **atomic:** for example **x is A**, where x is a linguistic variable and A is a linguistic value; the second one is called **compounded:** for example **x is A AND y is B OR z is NOT C**, this is a compounded atomic fuzzy proposition with the "AND", "OR" and "NOT" connectors, representing fuzzy intersection, union and complement respectively. The compounded fuzzy propositions are fuzzy relationships. The membership function of the rule IF-THEN is a fuzzy relation determined by a fuzzy implication operator. The fuzzy rules combine one or more fuzzy sets of entry, called antecedent, and are associated with one output fuzzy set, called consequents. The Fuzzy Sets of the antecedent are associated by fuzzy operators AND, OR, NOT and linguistic modifiers. The fuzzy rules permit expressing the available knowledge about the relationship between antecedent and consequents. To express this knowledge completely we normally have several rules, grouped to form what it is known a rule base, that is, a set of rules that express the known relationships between antecedent and consequents. The fuzzy rules are basically IF <Antecedent> THEN <Consequent> and expresses a fuzzy relationship or proposition.

In fuzzy logic the reasoning is imprecise or approximated, which means that we can infer from one rule a conclusion even if the antecedent doesn't comply completely. We can count on two basic inference methods between rules and inference laws, Generalized Modus Ponens (GMP) [5, 6, 8, 13] and Generalized Modus Tollens (GMT), that represent the generalizations of classic reasoning. The GMP inference method is known as direct reasoning and is summarized as:

Rule *IF x is A THEN y is B*
Fact *x is A'*

Conclusion *y es B'*

where A, A', B and B' are fuzzy sets of any kind. This relationship is expressed as $B' = A' \circ (A \rightarrow B)$. Figure 3 shows an example of Interval Type-2 direct reasoning with Interval Type-2 Fuzzy Inputs.

An Inference Fuzzy System is a rule base system that uses fuzzy logic, instead of Boolean logic utilized in data analysis [4, 10, 20]. Its basic structure includes 4 components (Fig. 4):

- ***Fuzzifier.*** Translates inputs (real values) to fuzzy values.
- ***Inference System.*** Applies a fuzzy reasoning mechanism to obtain a fuzzy output.
- ***Type Defuzzifier/Reducer.*** The defuzzifier translates one output to precise values; the type reducer transforms a Type-2 Fuzzy Set into a Type-1 Fuzzy Set.
- ***Knowledge Base.*** Contains a set of fuzzy rules, and a membership functions set known as the database.

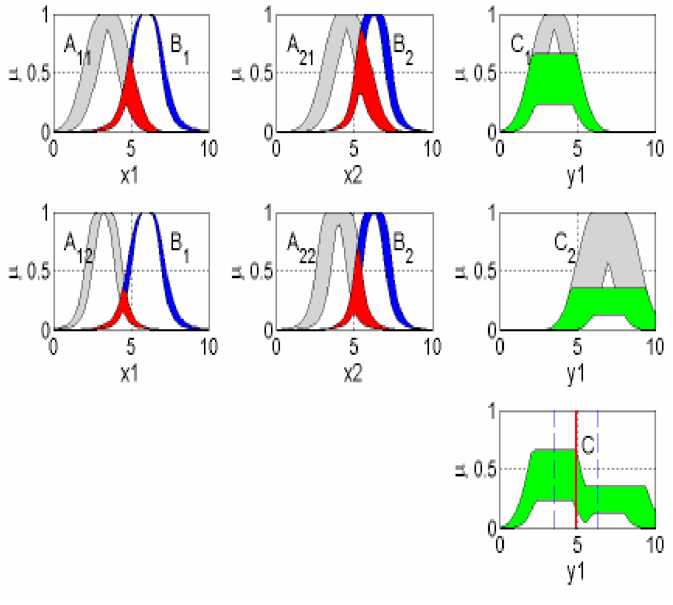


Fig. 3. Interval Type-2 Fuzzy Reasoning.

The decision process is a task that identifies parameters by the inference system using the rules of the rule base data. These fuzzy rules define the connection between the input and output fuzzy variables. A fuzzy rule has the form: IF <Antecedent> THEN <Consequent>, where antecedent is a compound fuzzy logic expression of one or more simple fuzzy expressions connected with fuzzy operators; and the consequent is an expression that assigns fuzzy values to output variables.

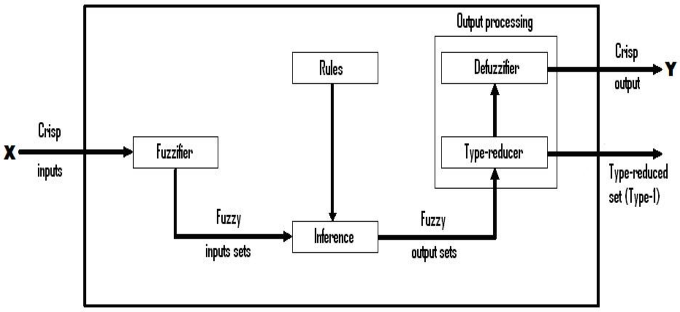


Fig. 4. Structure of the Type-2 Fuzzy Inference System.

III. INTERVAL TYPE-2 FUZZY SYSTEM DESIGN

The Mamdani and Takagi-Sugeno-Kang (TSK) Interval Type-2 Fuzzy Inference Models [10] and the design of Interval Type-2 membership functions and operators are implemented in the IT2FLS Toolbox (Interval Type-2 Fuzzy Logic Systems) reused from the Matlab® commercial Fuzzy Logic Toolbox.

The IT2FLS Toolbox includes a series of folders called dit2mf, it2fis, it2mf and it2op (Fig. 5). This folders contain the functions to create Mamdani and TSK Interval Type-2 Fuzzy Inference Systems (newfistype2.m), functions to add input-output variables and their ranges (addvartype2.m), it has functions to add 22 types of Interval Type-2 Membership functions for input-outputs (addmftype2.m), functions to add the rule matrix (addruletype2.m), it can evaluate the Interval Type-2 Fuzzy Inference Systems (evalifistype2.m), evaluate Interval Type-2 Membership functions (evalimftype2.m), it can generate the initial parameters of the Interval Type-2 Membership functions (igenparamtype2.m), it can plot the Interval Type-2 Membership functions with the input-output variables (plotimftype2.m), it can generate the solution surface of the Fuzzy Inference System (gensurftype2.m), it plots the Interval type-2 membership functions (plot2dtype2.m, plot2dctype2.m), a folder to evaluate the derivatives of the Interval type-2 Membership Functions (dit2mf) and a folder with different and generalized Type-2 Fuzzy operators (it2op, t2op).

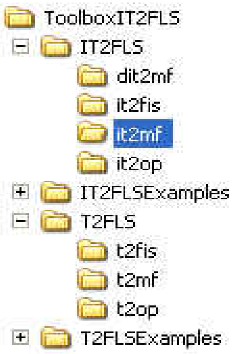

ToolboxIT2FLS
⊟ IT2FLS
dit2mf
it2fis
it2mf
it2op
⊞ IT2FLSExamples
⊟ T2FLS
t2fis
t2mf
t2op
⊞ T2FLSExamples

Fig. 5. Toolbox Folder.

The Interval Type-2 Fuzzy Inference Systems (IT2FIS) structure is the MATLAB object that contains all the interval type-2 fuzzy inference system information. This structure is stored inside each GUI tool. Access functions such as getifistype2 and setifistype2 make it easy to examine this structure. All the information for a given fuzzy inference system is contained in the IT2FIS structure, including variable names, membership function definitions, and so on. This structure can itself be thought of as a hierarchy of structures, as shown in the following diagram (Fig. 6).

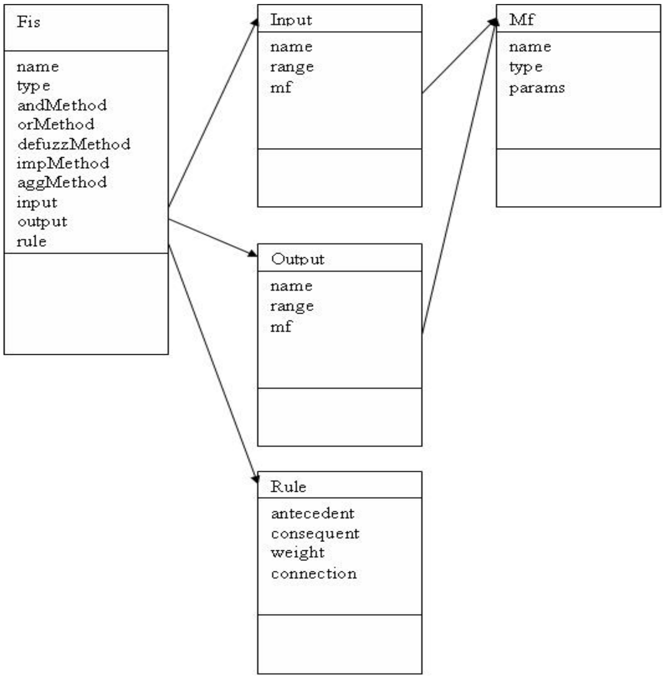
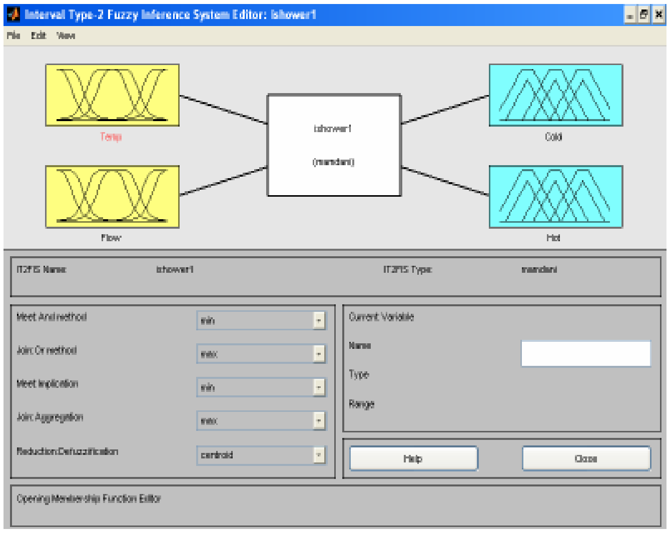


Fig. 6. Hierarchy of IT2FIS structures diagram.

The implementation of the IT2FLS GUI is analogous to the GUI used for Type-1 FLS in the Matlab® Fuzzy Logic Toolbox, thus permitting the experienced user to adapt easily to the use of IT2FLS GUI. Figures 7 and 8 show the main view of the Interval Type-2 Fuzzy Systems Structure Editor called IT2FIS (Interval Type-2 Fuzzy Inference Systems).



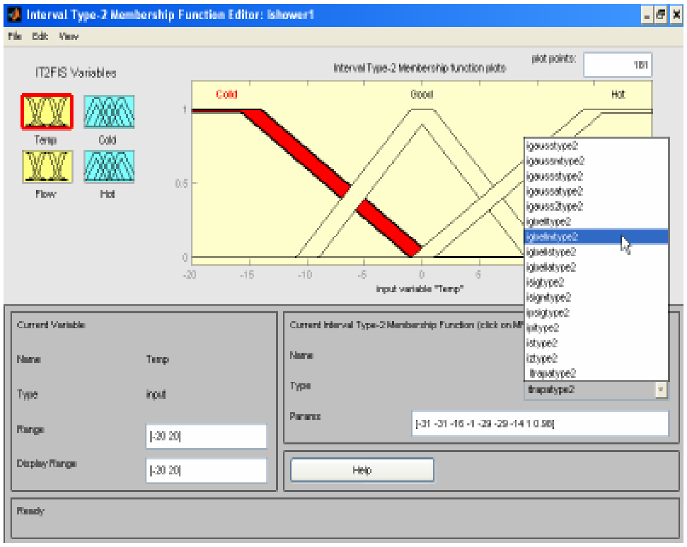
IT2FIS Name: ishower1
IT2FIS Type: mamdani

Meet And method: min
Join Or method: max
Meet Implication: min
Join Aggregation: max
Reduction Defuzzification: centroid

Current Variable
Name
Type
Range

Opening Membership Function Editor

Fig. 7. IT2FIS Editor.



Current Variable
Name: Temp
Type: input
Range: [-20 20]
Display Range: [-20 20]

Current Interval Type-2 Membership Function (click on MF
Name
Type: itrapatype2
Params: [-31 -31 -16 -1 -29 -29 -14 1 0.98]

Ready

Fig. 8. Interval Type-2 MF's Editor.

IV. SIMULATION RESULTS WITH THE INTERVAL TYPE-2 TOOLBOX

We present a shower simulation and a truck backer-upper simulation with interval type-2 fuzzy logic systems using the IT2FLS Toolbox. These examples are well described in the standard Toolbox of type-1 fuzzy logic.

A. *Shower Control Simulation.*

The application of the interval type-2 fuzzy control scheme to the shower gives good control results, which can be appreciated in Figure 9.

B. *Truck backer-upper control simulation.*

The application of the interval type-2 fuzzy control scheme to the truck backer-upper problem gives good control results and in Figures 10 and 11 we show the control results of the car trajectories for different initial conditions.

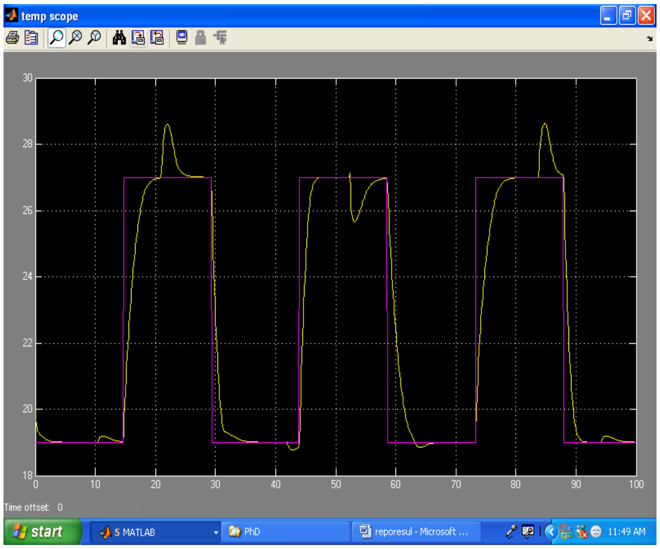
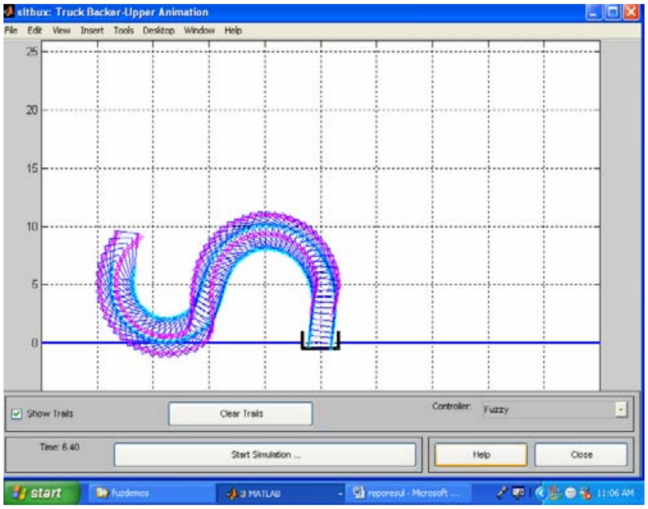


Fig. 9. Temperature interval type-2 fuzzy control.

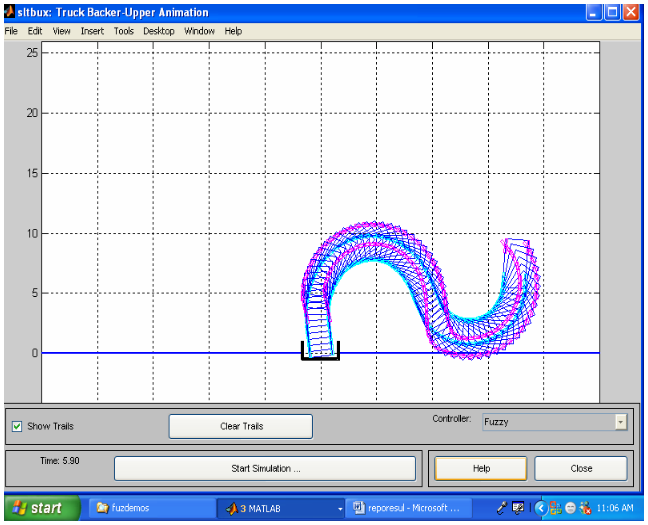

sltbux: Truck Backer-Upper Animation
File Edit View Insert Tools Desktop Window Help
25
20
15
10
5
0
Clear Trails
Start Simulation ...
Help
Close
start
fuzdemos
3 MATLAB
reporesul - Microsoft ...
11:06 AM

☑ Show Trails

Controller: Fuzzy

Time: 6.40

Fig. 10. Trajectory 1 Interval type-2 fuzzy control.


sltbux: Truck Backer-Upper Animation
File Edit View Insert Tools Desktop Window Help
25
20
15
10
5
0
Clear Trails
Start Simulation ...
Help
Close
start
fuzdemos
3 MATLAB
reporesul - Microsoft ...
11:06 AM

☑ Show Trails

Controller: Fuzzy

Time: 5.90

Fig. 11. Trajectory 2 interval type-2 fuzzy control.

V. Conclusions

The results in the interval type-2 fuzzy control case studies of the shower and the truck backer-upper have similar results to the type-1 fuzzy control with moderate uncertain footprints. To better characterize the interval type-2 fuzzy models we need to generate more case studies with better knowledge bases for the proposed problems, therefore classify the interval type-2 fuzzy model application strengths.

The design and implementation done in the IT2FLS Toolbox is potentially important for research in the interval type-2 fuzzy logic area, thus solving complex problems on the different applied areas.

Our future work is to improve the IT2FLS Toolbox with a better graphics user interface (GUI), to have compiled code, and integrate a learning technique Toolbox to optimize the knowledge base parameters of the interval type-2 fuzzy inference system, and design interval type-2 fuzzy neural network hybrid models.

References

[1] Zadeh, L.A., “Fuzzy sets,” *Information and Control*, Vol. 8, pp. 338-353, 1965.

[2] Zadeh, L.A., “Outline of a new approach to the analysis of complex systems and decision processes,” *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 3, No. 1, pp. 28-44, Jan. 1973.

[3] Zadeh, L.A., “The concept of a linguistic variable and its application to approximate reasoning, Parts 1, 2, and 3,” *Information Sciences*, 1975, 8:199-249, 8:301-357, 9:43-80.

[4] Zadeh, L.A., “Fuzzy Logic,” *Computer*, Vol. 1, No. 4, pp. 83-93, 1988.

[5] Zadeh, L.A., “Knowledge representation in fuzzy logic,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 1, pp. 89-100, 1989.

[6] Karnik, N.N. and J.M. Mendel. An Introduction to Type-2 Fuzzy Logic Systems, Univ. of Southern Calif., Los Angeles, CA., June 1998b.

[7] L. Zadeh, “Fuzzy logic = computing with words,” *IEEE Transactions on Fuzzy Systems*, vol. 2, pp. 103–111, 1996.

[8] Q. Liang and J. Mendel, “Interval type-2 fuzzy logic systems: Theory and design,” *IEEE Transactions Fuzzy Systems*, vol. 8, pp. 535–550, 2000.

[9] Dubois, D. and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York, 1980.

[10] J. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. NJ: Prentice-Hall, 2001.

[11] Mizumoto, M. and K. Tanaka, “Some Properties of Fuzzy Sets of Type-2,” Information and Control, vol. 31, pp. 312-340, 1976.

[12] Wang, L.-X., *Adaptive fuzzy systems and control: design and stability analysis*, Prentice Hall, 1994.

[13] R. Yager, “On the Implication Operator in Fuzzy logic,” Information Sciences, vol. 31, pp. 141-164, 1983.

[14] Yager, R.R., “A Caracterization of the Fuzzy Extension Principle,” J. Fuzzy Sets and Systems, vol. 18, pp. 205-217, 1986.

[15] E. H. Mamdani, and S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller,” *International Journal of Man-Machine Studies*, Vol. 7, No. 1, pp. 1-13, 1975.

[16] E. H. Mamdani, “Applications of fuzzy logic to approximate reasoning using linguistic synthesis,” *IEEE Transactions on Computers*, Vol. 26, No. 12, pp. 1182-1191, 1977.

[17] M. Sugeno and G. T. Kang. Structure identification of fuzzy model. Fuzzy Sets and Systems, 28:15-33, 1988.

[18] M. Sugeno, *Industrial applications of fuzzy control*, Elsevier Science Pub. Co., 1985.

[19] Y. Tsukamoto. An approach to fuzzy reasoning method. In Madan M. Gupta, Rammohan K. Ragade, and Ronald R. Yager, editors, Advances in fuzzy set theory and applications, pages 137-149. North-Holland, Amsterdam, 1979.

[20] R. Yager, “On a general class of fuzzy connectives,” *Fuzzy Sets and Systems*, 4:235-242, 1980.

[21] E. H. Mamdani, “Advances in the linguistic synthesis of fuzzy controllers,” *International Journal of Man-Machine Studies*, Vol. 8, pp. 669-678, 1976.

[22] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics, 15:116-132, 1985.

[23] J. R. Castro. Hybrid Intelligent Architecture Development for Time Series Forecasting. Masters Degree Thesis. Tijuana Institute of Technology. December 2005.

Computational Intelligence Software for Pattern Recognition and Time Series Prediction using Modular Neural Networks

Patricia Melin, Oscar Castillo, Olivia Mendoza

Abstract -In this paper we present a software tool to experiment with new neural multi-net structures, incuding the ensemble and modular approaches. This tool allow us to draw models, set parameters, save as project and generate files with results, always in a user friendly graphic enviroment. Another feature of the tool is the implementation of the Sugeno Integral formulas. this program was developed to allow the combination of any number of elements in the neural networks.

I. Introduction

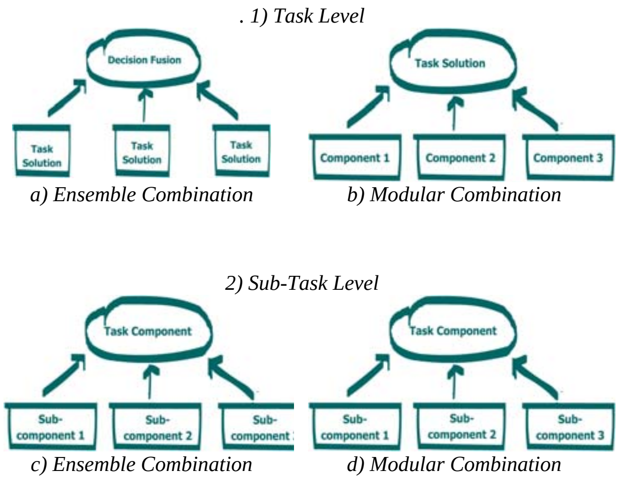
Neural computing has moved beyond simple demostrations to more significant applications. There is a growing realization that such applications can be facilitated by the development of multi-net systems.

Multi-net systems can result in the solution to tasks which either cannot be solved by a single net, or which can be more effectively solved by a system of modular net components. Similary, better performance can be achieved when ANNs, as predictors, are redundantly combined.[1].

To the moment, there are tools to create, train and test monolitic neural networks, but these tools do not allow us to create multi-net systems, for example, the "Neural Network Toolbox" included in Matlab does not allow the design of modular or ensemble neural networks [2]. For this reason, we decided to develop a tool for design and application of modular and ensemble neural networks models.

II. MULTI-NET SYSTEMS AND COMBINATION METHODS

It is important to make a distinction between the ensemble and modular combinations. The term "ensemble" is the one commonly used for combining a set of redundant nets, although the term "committee" or "committee machine" has also been used for the same purpose [3]. By contrast, under a modular approach, the task or problem is descomposed into a number of subtasks, and the complete task solution requires the contribution of all the several modules. Both ensemble and modular combinations can exist at either a task or a sub-task level, as shown in Fig. 1.



. 1) Task Level

a) Ensemble Combination   b) Modular Combination

2) Sub-Task Level

c) Ensemble Combination   d) Modular Combination

Fig. 1 Ensemble and modular multi-net systems, at task and sub-task levels.

A. Methods for Combining Nets in Ensembles.

Once a set of nets has been created, an effective way of combining their several outputs must be found. Some common methods in the current neural networks literature are:

1) Averaging and weighted averaging.
2) Non-linear combinations methods.
3) Supra Bayesian.
4) Stacked generalization.

B. Methods for Combininig Modular Components

It is possible to identify four diferent modes of combining component nets. A distinctions is made here between cooperative combination (fig. 2) and competitive combination (fig. 3). The main difference is that in cooperative combination it is assumed that all the elements to be combined will make some contribution to the decision, even this contribution may be weighted in some way; whereas in competitive combination, it is assumed that for each input the most apropiate element will be selected.

In sequential combination (fig. 4), the processing is successive; the computation of one module depending on the output of a preceding module. In a supervisory relationship (fig. 5), one module is used to supervise the performance of another module. [4]

This research work was supported in part by CONACYT and SAGARPA of Mexico under Grant 2003-002. The principal author (Patricia Melin) was also supported under the System of National Researchers of Mexico (SNI level II).

Patricia Melin is with the Division of Graduate Studies of Tijuana Institute of Technology, Tijuana, Mexico (corresponding author: e-mail: pmelin@tectijuana.mx).
Olivia Mendoza is with the School of Enginnering of UABC University of Tijuana, Mexico (e-mail: omendoza@tectijuana.mx).
Oscar Castillo is with the Division of Graduate Studies and Research of Tijuana Institute of Technology, Tijuana, Mexico (e-mail: ocastillo@tectijuana.mx).

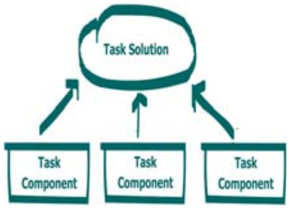


Fig. 2 Cooperative

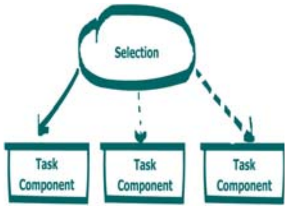


Fig. 3 Competitive

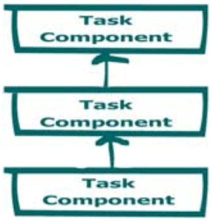


Fig. 4 Sequential

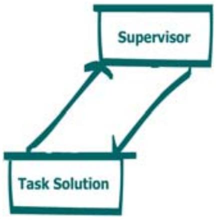


Fig. 5 Supervisory

III. Fuzzy sugeno integral

The Fuzzy Integral is an operator introduced in 1974 by Sugeno [5]. This operator is used to solve problems of multicriteria decision making, where the information that is combined is based in fuzzy measures determined by an expert. The goal is the simulation of the human process for the integration of different source of information.[6].

Fuzzy measures are functions applied to fuzzy sets and they consist of different coefficients call fuzzy densities. Each fuzzy density rates the relevance of the different sets and their combinations, in order to satisfy certain hypothesis.

There are two types of Fuzzy Integral: Choquet Fuzzy Integral (1) and Sugeno Fuzzy Integral (2).[7]

$$h(\sigma_1,\cdots,\sigma_n)=\max_{i=1}^{n}(\min(\sigma_i,\mu(x_i,\cdots,x_n))) \qquad (1)$$

$$h(\sigma_1,\cdots,\sigma_n)=\sum_{i=1}^{n}(\sigma_i-\sigma_{i-1})\mu(\{x_i,\cdots,x_n \qquad (2)$$

Where $\sigma_i = \sigma(x_i)$ and $0 \le \sigma_1 \le \ldots \le \sigma_n \le 1$

A. Fuzzy measures

A fuzzy measure μ, with respect to the data set X, it must satisfy the following conditions [8][9].

1) $\mu(X)=1$, $\mu(\varnothing)=0$
2) If $S \subseteq T$, then $\mu(S) \le \mu(T)$

Where S y T are subsets of X.
One fuzzy measure is a Sugeno Measure or λ-fuzzy, if it satisfies the following condition of addition for some $\lambda > -1$.

$$\mu(S \cup T)= \mu(S)+\mu(T)+ \lambda\mu(S)\mu(T) \qquad (3)$$

λ can be calculated of the following way:

$$\mu(S)=\left[\prod_{x\in S}(1+\lambda\mu(\{x\}))\right]\Big/\lambda \qquad (4)$$

$$\lambda+1=\prod_{i=1}^{n}(1+\lambda\mu(\{x_i\})) \qquad (5)$$

One form of calculate Sugeno measures, it is carrying out the calculation of recursive way [10][11] using (6),(7).

$$\mu(A_1)=\mu(x_1) \qquad (6)$$
$$\mu(A_i)=\mu(x_i)+\mu(A_{i-1})+\lambda\mu(x_i)\mu(A_{i-1}) \qquad (7)$$

Where $1<i\le n$, and the values to μ(xi) corresponds to the fuzzy densities determined by an expert.

.

B. Example for calculation of Sugeno Measures.

Consider the set $X=\{x_1,x_2,\ldots x_n\}$, the fuzzy density values are given as follows:
$\mu(x_1)=0.3$, $\mu(x_2)=0.4$, $\mu(x_3)=0.1$
The value of λ can be calculated by (5), solving the following equation, using some numeric method to found the root of f(λ)([12]:

$$1+\lambda=(1+0.3\lambda)(1+0.4\lambda)+(1+0.1\lambda)$$

The solutions are λ= -16.8 y λ=0.9906, if λ>-1, then λ=0.9906
The Sugeno measures (3) can be constructed as follows:

$\mu(x_1)=0.3$, $\mu(x_2)=0.4$, $\mu(x_3)=0.1$
$\mu(x_1,x_2)=\mu(x_1)+\mu(x_2)+\lambda(\mu(x_1)\mu(x_2))=0.8189$
$\mu(x_1,x_3)=\mu(x_1)+\mu(x_3)+\lambda(\mu(x_1)\mu(x_3))=0.4297$
$\mu(x_2,x_3)=\mu(x_2)+\mu(x_3)+\lambda(\mu(x_2)\mu(x_3))=0.5396$
$\mu(x_1,x_2,x_3)=1$

or using (6) y (7)

$\mu(A_1)=0.3$
$\mu(A_2)=0.4+0.3+(0.9906)(0.4)(0.3)= 0.8189$
$\mu(A_3)=0.1+0.8189+(0.9906)(0.1)(0.8189)= 1$

C. Example for the calculation of Sugeno Integral

The values of the data set $\sigma(x_i)$, that correspond to the data to clasify, must be ordered ascending.

Another condition for the values of the data set σ(xi), it is that they must belong to the rank of values [0,1].
σ(x1)=0.9, σ(x2)=0.6, σ(x3)=0.3
by (1):
h(0.9,0.6,0.3)=max(min(0.9,0.3),min(0.6,0.8189),min(0.3,1))
h(0.9,0.6,0.3)=max(0.3,0.6,0.3)=0.6

IV. Operation and testing of the system

Next, a brief demonstration of the operation of the ***hmr*** system will be given, beginning by describing the types of objects available to draw a multi-net system model.

Tool	Name	Description
□	Ensemble	Solution to Combination of Member Nodes.
🟩	Member.	Neural Net and Member Node of an Ensemble.
▦	Fusion Member.	Ensembre Member and Solution to Combination of Modules.
▦	Fusion.	Solution to Combination of Modules.
🟩	Module.	Neural Net and Module.
□	Commite Module.	Module and Solution to Combination of Member Nodes.
▦	Fusion Mudule	Module and Solution to Combination of Modules .
🟡	Input Data.	Input data for a Neural Net.
○	Target Data.	Target data for a Neural Net.
╱	Union.	Union between nodes.

A. Example: A project for time series prediction.

This model (Fig. 6) has only one solution node. It is an Ensemble with 4 Neural Net members. The problem is to predict the prices of tomato in Mexico.

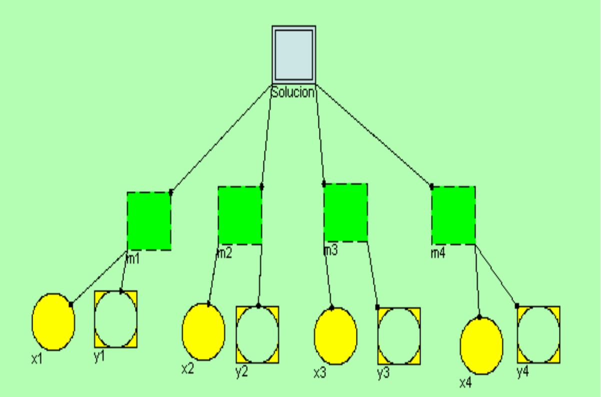


Fig.6 Model: Tomates1

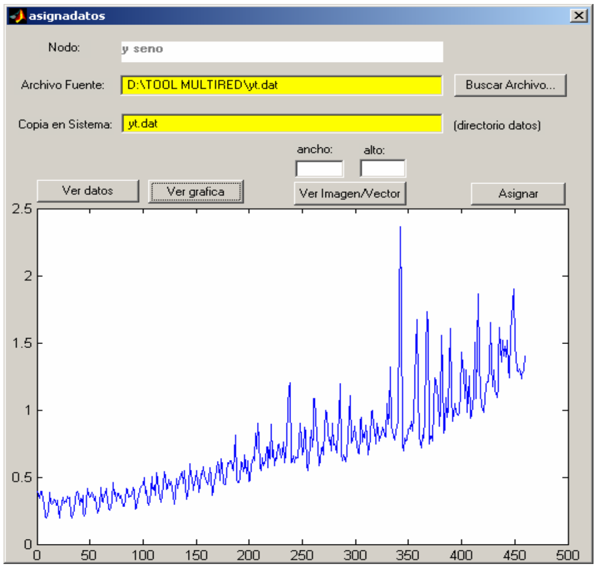
B. Preparing the project to work.

Once the model was drawn, it is time to assign parameters and data, the optimal order of the procedure is described next:

Assign data to nodes: Input and Target.
Assign parameters and train the Neural Networks nodes.
Assign parameters and test the solution nodes.
With these steps the project will be prepared to work in the system.

C. Assigning data to nodes: Input and Target.

Select the node with a double click. In the window shown in Fig. 7 select the text file where the data set is stored.


asignadatos

Nodo: y seno

Archivo Fuente: D:\TOOL MULTIRED\yt.dat

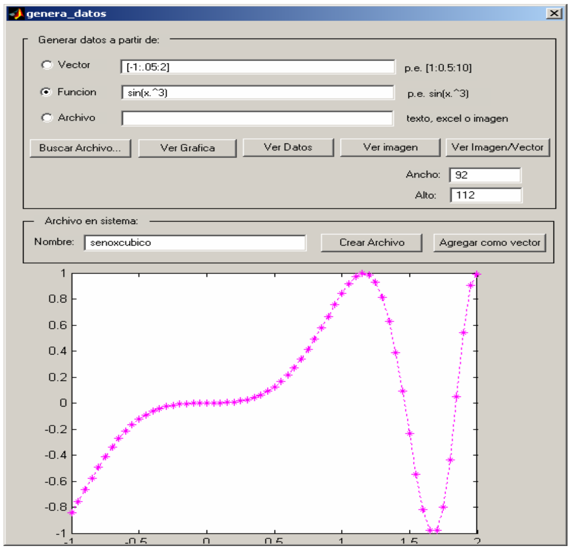
Copia en Sistema: yt.dat (directorio datos)

ancho: alto:

Fig.7 Dialalog box to Assign Data Sets.

D. Creating data files from manual input or functions.

In the window shown in fig. 8, it is possible to create data files from vectors, functions, image files, text files and Excel worksheets.


genera_datos

Generar datos a partir de:

○ Vector [-1:.05:2] p.e. [1:0.5:10]
● Funcion sin(x.^3) p.e. sin(x.^3)
○ Archivo texto, excel o imagen

Ancho: 92
Alto: 112

Archivo en sistema:
Nombre: senoxcubico

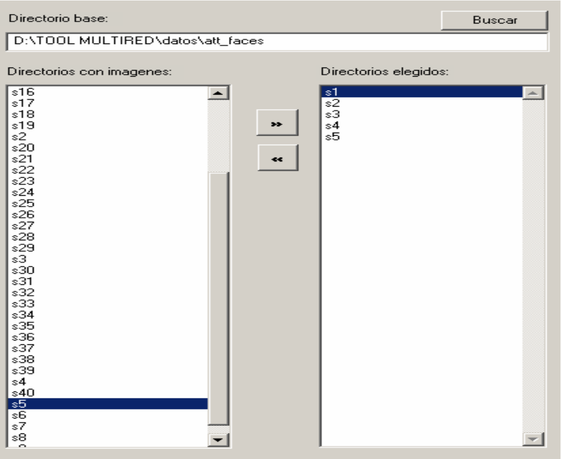
Fig. 8 Dialog box to create data files

E. Creating image data sets.

Create image data sets means to create a set of data files from image files stored of the folowing form.

F. The file layout

The system takes like standard the file layout of the database: ORL [14], (fig. 9).



Directorio base: Buscar

D:\TOOL MULTIRED\datos\att_faces

Directorios con imagenes:

s16
s17
s18
s19
s2
s20
s21
s22
s23
s24
s25
s26
s27
s28
s29
s3
s30
s31
s32
s33
s34
s35
s36
s37
s38
s39
s4
s40
s5
s6
s7
s8

Directorios elegidos:

s1
s2
s3
s4
s5

Fig.9 Selection of image files to create an image data set

G. Example

One image data set for the subjects s1, s2, s3,s4 and s5 will be created with three samples, with images divided in 3 parts and horizontal direction, the results are in the figures: 10, 11 and 12.

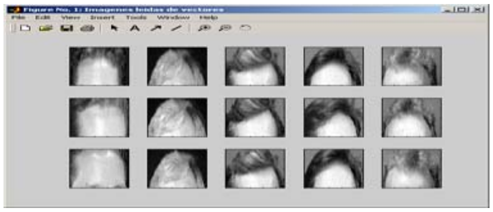


Figure No. 1: Imagenes leidas de vectores
File Edit View Insert Tools Window Help

Fig. 10: Juego_pruebaDiv_1.dat

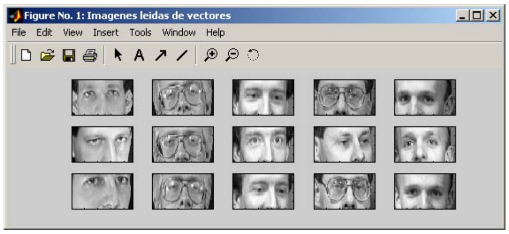


Figure No. 1: Imagenes leidas de vectores
File Edit View Insert Tools Window Help

Fig. 11 Juego_pruebaDiv_2.dat

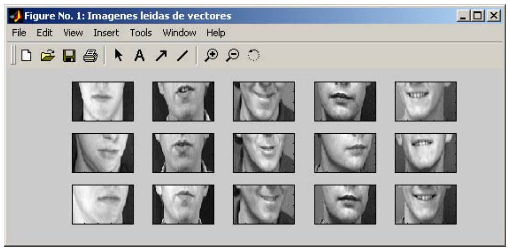
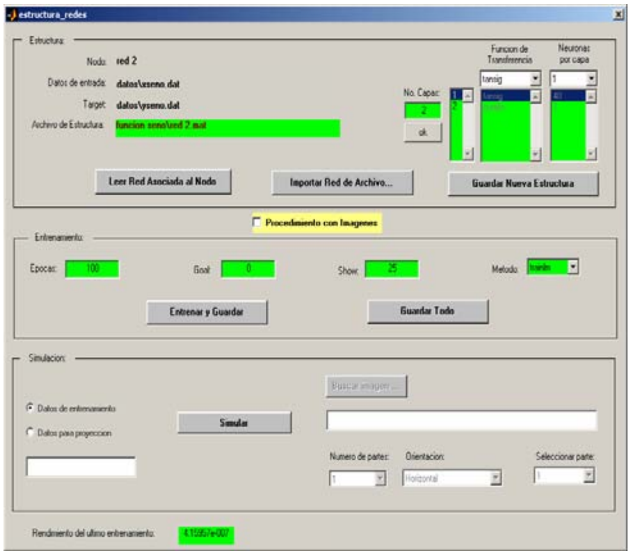


Figure No. 1: Imagenes leidas de vectores
File Edit View Insert Tools Window Help

Fig. 12 Juego_pruebaDiv_3.dat

H. Structure, Training and Simulation of monolitic neural nets.

To create the multi-net model, first we must test each monolitic net. In order to do this, is necessary to give parameters, train and test the learning level obtained.
(fig. 13)



estructura_redes

Estructura:

Nodo: red 2

Datos de entrada: datos\xseno.dat

Target: datos\yseno.dat

Archivo de Estructura: funcion seno\red 2.mat

No. Capas: 2

Funcion de Transferencia: tansig

Neuronas por capa: 1

□ Procedimiento con Imagenes

Entrenamiento:

Epocas: 100

Goal: 0

Show: 25

Metodo: trainlm

Simulacion:

● Datos de entrenamiento

○ Datos para proyeccion

Numero de partes: 1

Orientacion: Horizontal

Seleccionar parte: 1

Rendimiento del ultimo entrenamiento: 4.15957e-007

Fig. 13 Dialog box for configuration, train and test each neural net.

I. Defining the net structure.

The system has three forms to load the structure of the neural network:

Leer Red Asociada al Nodo: Load a .mat file, that contains the structure of a neural net asociated to de selected net node.
Guardar Nueva Estructura**:** Take the values especified by the user for the selected net node, and stored it in a .mat file.
Importar Red de Archivo: Load a .mat file, that contains the structure of a neural net, even that is not associated to the selected net node.

J. Procedures for discreet data and images

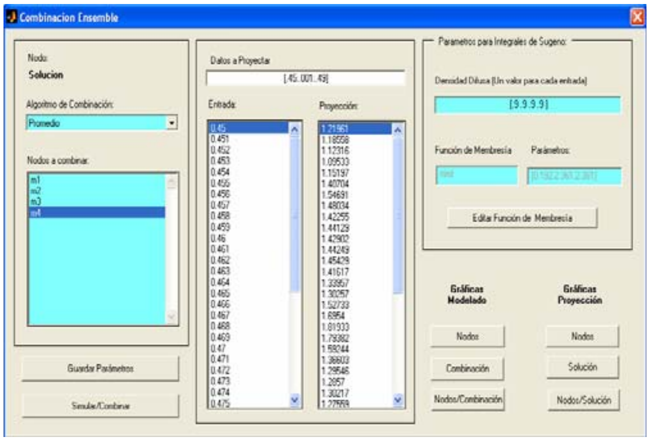
The **hmr** system when creating the model, accepts any node combination, but the training, simulation and final solution are specifically developed to solve two classic problems:

Prediction of discrete data: Tests with ensemble combination. In this problem the train and simulation only calls the functions of the neural net tool box.

Image recognition: Tests with modular combination. In this problem the train and simulation procedures are based in the matlab demo “appr1.mat” [2], for the recongnition of the alphabet with noise.

K. Ensemble Combination

The window shown in the Fig. 14 allows us to test a combination of nodes in ensembles. In order to do this, all the nodes to be combined must be trained and tested before.



Nodo: Solucion

Algoritmo de Combinación: Promedio

Nodos a combinar: m1 m2 m3 m4

Datos a Proyectar: [.45:.001:.49]

Entrada:	Proyección:
0.45	1.21561
0.451	1.18558
0.452	1.12316
0.453	1.09533
0.454	1.15197
0.455	1.40704
0.456	1.54691
0.457	1.48034
0.458	1.42255
0.459	1.44129
0.46	1.42902
0.461	1.44249
0.462	1.45429
0.463	1.41617
0.464	1.33957
0.465	1.30257
0.466	1.52733
0.467	1.6954
0.468	1.81933
0.469	1.79382
0.47	1.59244
0.471	1.36603
0.472	1.29546
0.473	1.2857
0.474	1.30217
0.475	1.27559

Parametros para Integrales de Sugeno:

Densidad Difusa [Un valor para cada entrada]: [.9 .9 .9 .9]

Función de Membresía: [illegible]

Parámetros: [0.1922.361 2.361]

Fig. 14 Dialog box to combine nodes in ensembles.

Gráficas para modelado: Presents the result of simulation for each node using the trainig data (fig. 15, 16 and 17).

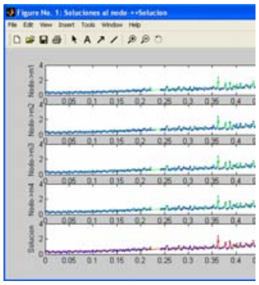


Fig. 15 Nodes graphic.

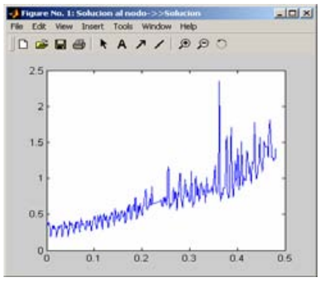


Fig. 4.16 Combinaction graphic.

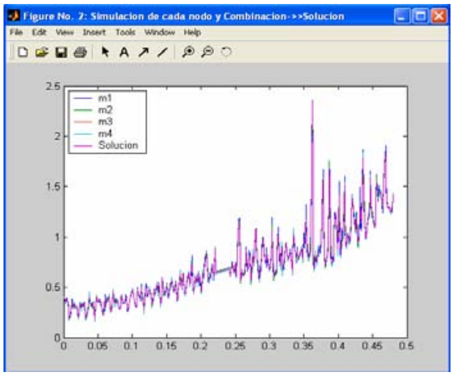


Fig. 17 Nodes/Combination graphic

Gráficas de Proyección: Presents the result of simulation for each node using the proyection data set (fig. 18, 19 and 20).

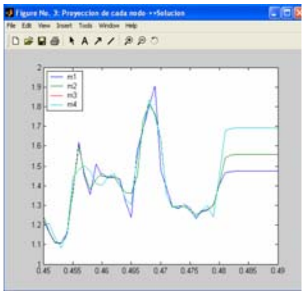


Figure 18 Nodes graphic

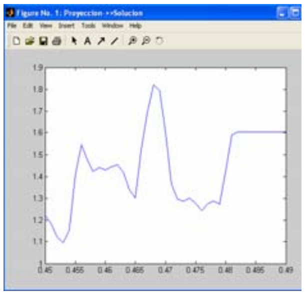


Figure 19 Solution graphic

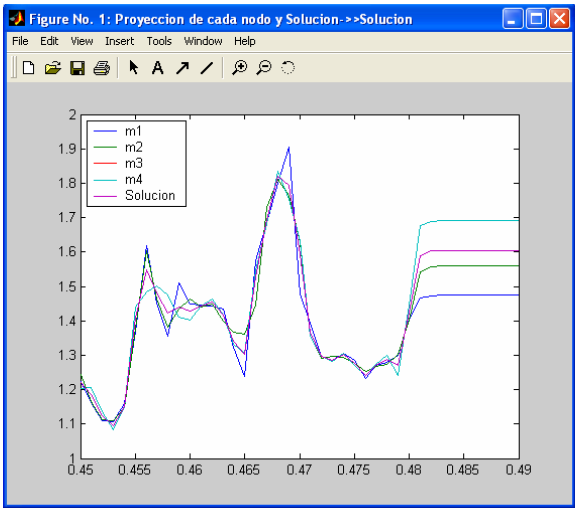


Figura 20 Nodes/Solution graphic.

L. Module Combination (Cooperative Method).

The cooperative method focused to the combinations of modules for the recognition of images divided in several parts. In the model shown in fig. 21, each net node was trained with one of 4 parts of 5 differents subjects, with 7 samples.

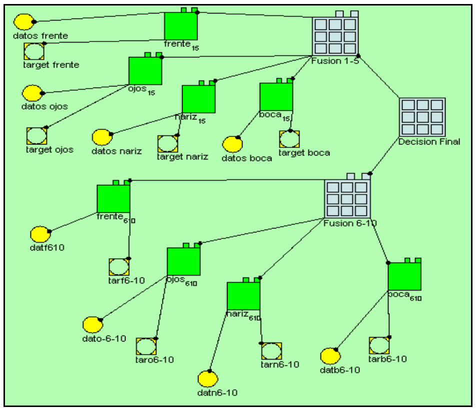


Fig. 21 Model "Fotos 4 partes"

For example, the module "frente" was trained with 7 samples of the superior part of the face of 5 differents subjects, as shown in fig. 22.

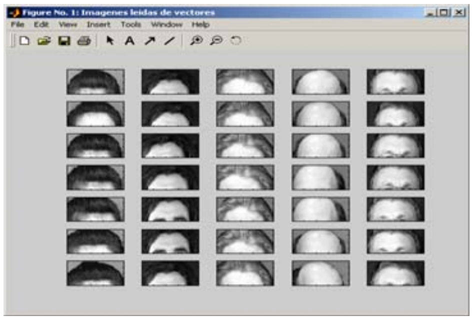


Figure No. 1: Imagenes leidas de vectores
File Edit View Insert Tools Window Help

Fig. 22 Images assigned to the input data node: "datos frente".

We chose the method of combination by Cooperation and ordered the nodes in the list, the results are in the figures 23, 24 and 25.

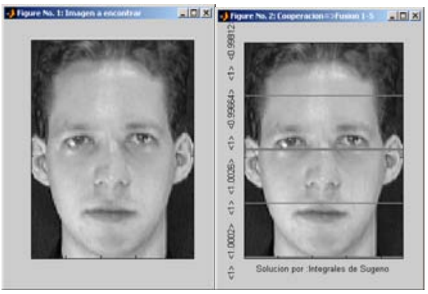
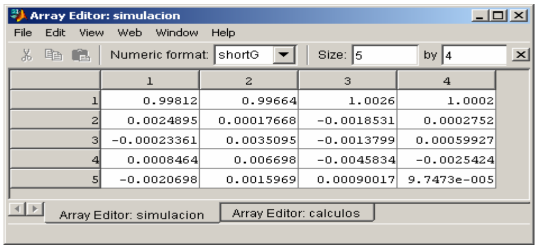


Figure No. 1: Imagen a encontrar
Figure No. 2: Cooperacion=>Fusion 1-5
<1> <0.99812> <1> <0.99664> <1> <1.0026> <1> <1.0002>
Solucion por :Integrales de Sugeno

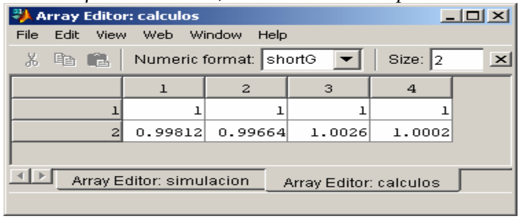
Fig. 23 Results of search of the subjet number 1, in the cooperative module "Fusion 1-5".



Numeric format: shortG Size: 5 by 4

	1	2	3	4
1	0.99812	0.99664	1.0026	1.0002
2	0.0024895	0.00017668	-0.0018531	0.0002752
3	-0.00023361	0.0035095	-0.0013799	0.00059927
4	0.0008464	0.006698	-0.0045834	-0.0025424
5	-0.0020698	0.0015969	0.00090017	9.7473e-005

Fig. 24 Results of the simulation, each column corresponds to one module.



Numeric format: shortG Size: 2

	1	2	3	4
1	1	1	1	1
2	0.99812	0.99664	1.0026	1.0002

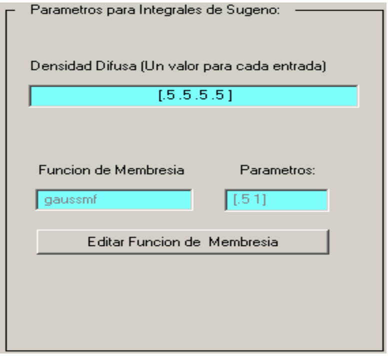
Fig. 25 The value of the Sugeno Integral in each module.

M. Competitive method

In the same model, to obtain the final decision, all the cooperative modules are combined in a competitive module, that make the competence and show the image with the best values.

N. Decision algorithms

The decision algorithms that support the system are: Promedio(Average), Votación(Voting), Valor Máximo (Maximum value) and Integral de Sugeno (Sugeno Integral).


Parametros para Integrales de Sugeno:
Editar Funcion de Membresia

Densidad Difusa (Un valor para cada entrada): [.5 .5 .5 .5]

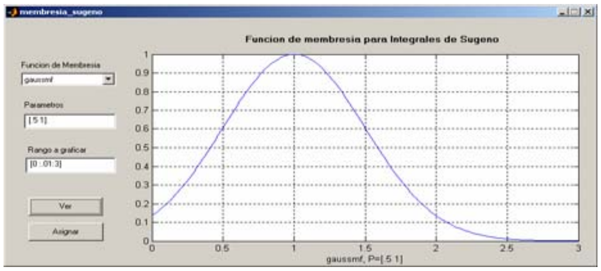
Funcion de Membresia: gaussmf

Parametros: [.5 1]

Fig. 26 Data necessary to calculate the Sugeno Integral

In both types of combination it is possible to choose the Sugeno Integral method and set parameters by the window shown in figures 26 and 27.

Fuzzy density: One value for each combined node. This values must be in the rank [0,1], and represents the spected value for the optimal solution.
Membership function: Function available in the "Matlab Fuzzy Logic Tool box", and it is necesary to obtain fuzzy values for the solutions of de nodes to combine.



Funcion de Membresia: gaussmf

Parametros: [.5 1]

Rango a graficar: [0 :.01:3]

Fig. 27 Window to configure the membership function.

V. Conclusions

This paper was centered in the different forms to combine the results of neural networks of the backpropagation type. These are useful in known applications like prediction and image recognition. The algorithm for the "Sugeno Integral" was developed, which presents a greater variety of options thanks to its input parameters, which helps so that the combination is formed of the most advisable way for each application. All the calculations within the system are designed to obtain results for any number of nodes, the limits are imposed by the memory and capacity of the computer where it is executed, not for the structures within the system.

References

[1] Amanda J.C. Sharkey, United Kingdom, 1999. "Combining Artificial Neural Nets" (Ensemble and Modular Multinet Systems), Ed. Springer, pp. 2,3,4.

[2] The MathWorks, Inc. ©1994-2004. "Neural Network Toolbox 4.0.4", (Product Description), http://www.mathworks.com/products/neuralnet/description1.html, (may, 2004).

[3] Sharkey A, United Kingdom, 2001. "On Combining Artificial Neural Nets", http://dcs.shef.ac.uk/~amanda/Papers/comb.ps, (march, 2003).

[4] Amanda J.C. Sharkey, 1999. "Combining Artificial Neural Nets (Ensemble and Modular Multinet Systems)", Ed. Springer, pp. 15-27.

[5] Didier Dubois, Jean-Luc Marichal, Henri Prado,Marc Roubens, R´egis Sabbadin, Francia, 2000. "The use of the discrete Sugeno integral in decision-making : a survey", http://www.worldscinet.com/ijufks/09/0905/S0218488501001058.html, (september, 2004).

[6] Erkan Duman, Turquía, 2003. "A New Fuzzy Integral Model For Control Systems: Adaptive Fuzzy Integral", http://www.ijci.org/product/tainn/E08012.pdf, (september 2004).

[7] Ruiz-del-Solar, A. Soria-Frisch, "Sistemas Multisensoriales de Inspección Industrial: Procesamiento Conjunto de Imágenes de Color e Infrarrojas", (september, 2004).

[8] H. R. Tizhoosh, Waterloo, Inglaterra, 1997. "Fuzzy Measure Theory", http://watfor.uwaterloo.ca/tizhoosh/measure.htm, (may, 2004).

[9] Arunas Lipnickas, Lithuania, 2001. "Classifiers Fusion With Data Dependent Aggregation Schemes", http://www.elen.ktu.lt/~arunas/public/alincfddas.pdf, (september, 2004),

[10] A.Verikas, A. Lipnickas, K. Malmqvist, Korea, 2000. "Fuzzy measures in neural networks fusion", http://www.elen.ktu.lt/~arunas/public/aviconip2000.pdf, (september, 2004).

[11] A.Lipnickas, Bielorrusia, 2001. "Classifiers Fusion with Data Dependent Aggregation Schemess", http://www.elen.ktu.lt/~arunas/public/alincfddas.pdf, (september, 2004).

[12] A. Nieves, México, 1998. "Métodos Numéricos Aplicados a la Ingeniería", Ed. CECSA, p.p 34-57.

[13] The MathWorks, Inc. ©1994-2004. "Fuzzy Logic Tool Box, For use with Matlab"

[14] AT&T Laboratories, Cambridge, 2002, "The ORL Database of Faces", http://www.uk.research.att.com/facedatabase.html, (may, 2004).

Development of the Intelligent Sensor Network Anomaly Detection System: Problems and Solutions

Leon Reznik and Carll Hoffman

Abstract— **The paper describes the development of the Sensor Network Anomaly Detection System (SNADS). SNADS is designated to become a framework to support signal change detection in sensor networks (SN), the backbone application of a crucial importance for design of other SN applications as well as for improving SN performance and security in general. SNADS will provide a cross-platform management of core SN operation. Although SN is built as a synergy of a variety of modules, the system centerpiece is a collection of intelligent agents realizing different computational intelligence and machine learning techniques employed for change detection in signals coming from sensors. The agents may have various levels of intelligence from a simple comparison against the thresholds through the rules system to neural networks structures distributed over the sensor network nodes. The system is written in Java and could be implemented with the JVM technology. The system composition and component design are described. Implementation details are given.**

***Index Terms*—Sensor networks; Signal processing; Java programming**

I. Introduction

This paper describes the software framework integrating computational intelligence and machine learning techniques for signal change detection in signal processing and other related applications. It will explain the development of the Sensor Network Anomaly Detection System (SNADS), which is designated to become a synergetic structure of various intelligent agents, making a change detection decision in signal processing applications. Although the current design focuses on sensor networks (SN) applications, many agents and the framework as a whole are anticipated to be useful in applications far beyond this technology, even for solving problem other than signal processing. An example of using one of its agents for detecting edges in images is given in [1]. The framework is build up as a cross-platform tool with autonomous parts, many of them working independently from each other. The design idea is to make the framework valuable to a great variety of members of the computational intelligence and machine learning virtual community as well as scholars and professionals who want to employ or wish to investigate a feasibility of applying the intelligent methodologies in their projects. Depending on a particular project's specification, users should be able to choose specific modules and embed them directly in their designs. This will allow for a much faster project prototyping that will cause a significant cost and time reduction. It might also facilitate a feasibility study and a comparative analysis of various intelligent agents in different applications on the implementation level much closer to the final product than the one provided by widely used high level design packages such as Matlab.

Sensor networks have been chosen as an initial designation field because it is an emerging technology with a multitude of important applications in various fields ranging from scientific surveillance to military and security. With the majority of SN employed currently for object and environment monitoring, signal change detection has an ultimate importance in sensor networks. Although a change detection in signals has been investigated for a considerable time, over the last decade there have been new important developments. The literature on change detection is rapidly growing mainly due to applications in engineering, financial mathematics and econometrics. The change detection techniques have developed into various models, which may be classified into likelihood ratio tests, nonparametric approaches, linear model approaches and intelligent techniques [2,3]. Csorgo and Horvath [4] provide a concise overview and rigorous mathematical treatment of methods for a change detection and use a number of datasets to illustrate the effectiveness of the various techniques. Applications of intelligent techniques for personal and technical systems monitoring based on SN applications were reported recently [5-7]. Results received by the first author [8-12] indicate that neural networks and fuzzy models are feasible for description of the kind of uncertainty we may anticipate in SN signals.

SNADS provides support for a wide spectrum of intelligent change detection methods. While triggering alarms based on the limited data provided by the sensors can be accomplished using conventional constructs, the implementations could involve various methodologies and more complicated models, which in turn might require different resources. Depending on the information and resources availability, a variety of models and

Authors are with the Department of Computer Science, Rochester Institute of Technology , 102 Lomb Memorial Drive, Rochester, NY 14623 USA (corresponding author phone: 585-475-7210; fax: 585-475-7100; e-mail: lr@cs.rit.edu).

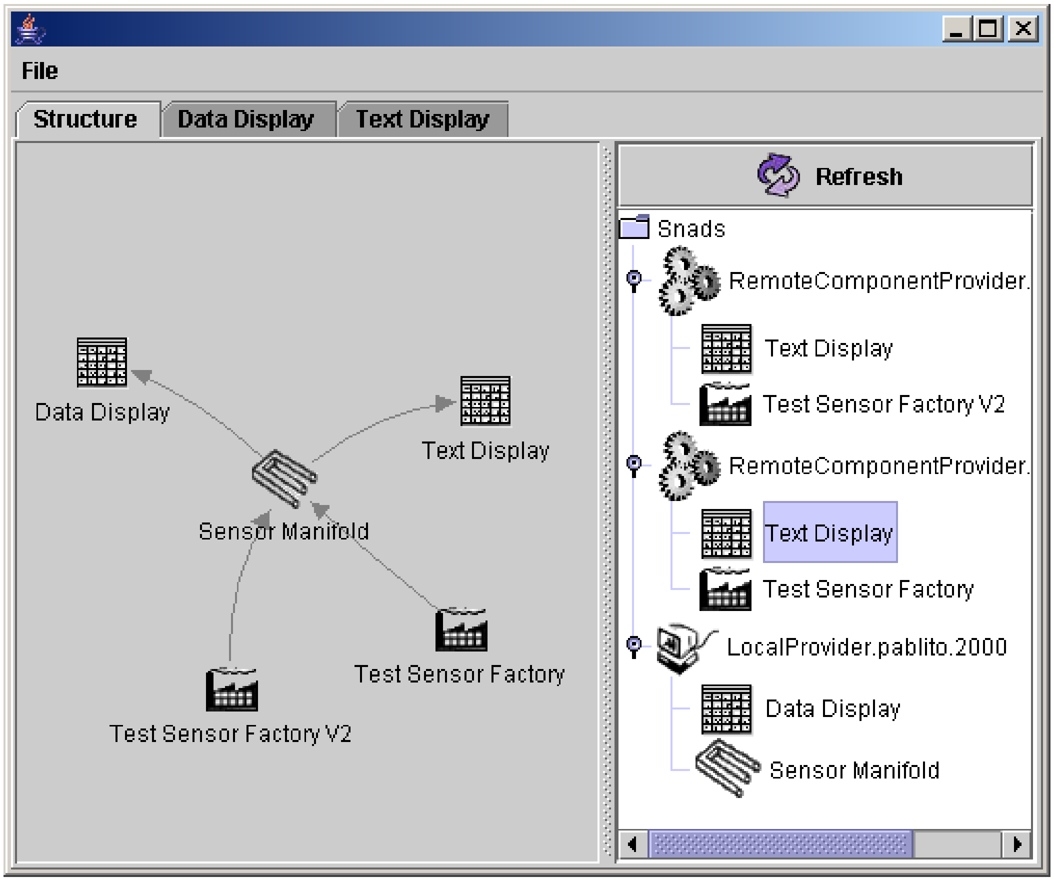


Fig. 1. Graphical system design by component reconfiguration and connection

methodologies could be used in a particular application. In one case, the alarms could be triggered according to the IF/THEN/ELSE rules. For another monitoring application, the system can maintain a history of past values so that it can react according to how quickly measurement results change. Finally, a machine learning technique could be used to derive the model dynamically in order to determine the outcome of the change detection. From the implementation point of view, the application execution will be distributed over the network with simple models used on lower levels and more complicated ones on the upper levels. Depending on the resource availability, however, models and utilities might be redistributed over the network in a given implementation. Simple methods like a comparison against the threshold could be embedded in lower MAC level protocols, while machine learning techniques will be built upon applications.

The paper provides a detail information on SNADS design implementation. SNADS includes the following basic components:

1) Message producing and routing system (MPR),
2) Sensor configuration and management (SCM),
3) Graphical user interface (GUI),
4) Change or anomaly detection (ADS),
5) Database and storage (DB)

Section II provides the design functional specification with a brief description of each component. The basic implementation features including main data structures used to implement sensor querying and processing their signals are given in section III. System configuration and compilation procedures are referred to in section IV with a conclusion drawn down in section V.

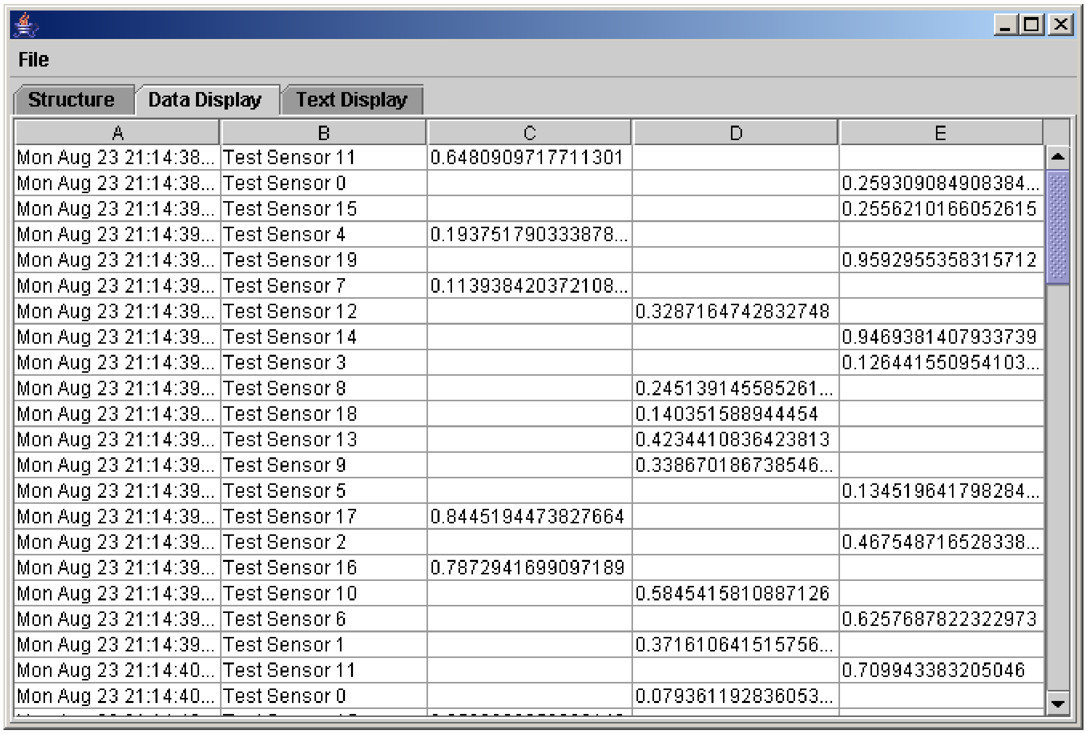
II. SNADS Components and their functionality

A. Design overview

The SNADS system is designed as modular, extensible, robust, and scalable. By providing a generic sensor abstraction and sensor-definable configuration mechanisms, SNADS allows for simple, secure management of arbitrary sensor networks. By supporting network nodes with different hardware and software configurations, SNADS will be a versatile cross platform tool. Components will be added and removed from the system during runtime, and components can be upgraded to a newer version on the fly. The architecture is scalable allowing for components to be spread across multiple computing devices. SNADS is designed as an event based system that would allow it to function in a real time mode given enough resources. In this sense some architectural design decisions are similar to those taken in TinyOS design [13].

1) Modularity

Modular design functionality is achieved via implementing a central messaging system, which allows components to work



A	B	C	D	E
Mon Aug 23 21:14:38...	Test Sensor 11	0.6480909717711301		
Mon Aug 23 21:14:38...	Test Sensor 0			0.259309084908384...
Mon Aug 23 21:14:39...	Test Sensor 15			0.2556210166052615
Mon Aug 23 21:14:39...	Test Sensor 4	0.193751790333878...		
Mon Aug 23 21:14:39...	Test Sensor 19			0.9592955358315712
Mon Aug 23 21:14:39...	Test Sensor 7	0.113938420372108...		
Mon Aug 23 21:14:39...	Test Sensor 12		0.3287164742832748	
Mon Aug 23 21:14:39...	Test Sensor 14			0.9469381407933739
Mon Aug 23 21:14:39...	Test Sensor 3			0.126441550954103...
Mon Aug 23 21:14:39...	Test Sensor 8		0.245139145585261...	
Mon Aug 23 21:14:39...	Test Sensor 18		0.140351588944454	
Mon Aug 23 21:14:39...	Test Sensor 13		0.4234410836423813	
Mon Aug 23 21:14:39...	Test Sensor 9		0.338670186738546...	
Mon Aug 23 21:14:39...	Test Sensor 5			0.134519641798284...
Mon Aug 23 21:14:39...	Test Sensor 17	0.8445194473827664		
Mon Aug 23 21:14:39...	Test Sensor 2			0.467548716528338...
Mon Aug 23 21:14:39...	Test Sensor 16	0.7872941699097189		
Mon Aug 23 21:14:39...	Test Sensor 10		0.5845415810887126	
Mon Aug 23 21:14:39...	Test Sensor 6			0.6257687822322973
Mon Aug 23 21:14:39...	Test Sensor 1		0.371610641515756...	
Mon Aug 23 21:14:40...	Test Sensor 11			0.709943383205046
Mon Aug 23 21:14:40...	Test Sensor 0		0.079361192836053...	

Fig. 2 Sensor measurement tabular display

together as a number of black-box entries. This setup allows for simple runtime reconfiguration of the SNADS system and minimizes the damage a malfunctioning component can cause.

2) Extensibility and upgrading

The messaging system is also designed to be dynamic and highly extensible. Components can be added, removed, or replaced on the fly by the administrator. Updating to a new version of a component can be done without shutting down SNADS or the network. Self-upgrading and self-modifying option is planned but has not been implemented yet.

3) Scalability

A SNADS system implemented over a sensor network platform should be scalable to an arbitrary degree in relation to the number of agents included and modules working together within the limitations given by the hardware platforms in use. From a simple test network to a thousand node perimeter monitoring system, the SNADS architecture can handle it. Because of the message passing abstraction, various SNADS components could easily be located on separate machines, and the workload of a single component could be spread across multiple machines.

B. Component design basics

1) Message Producing and Routing

The message routing subsystem supports communication between modules in the SNADS architecture and by this way provides the functionality of the whole system in a real time mode. By registering with this system, a component will be able to send events to and receive events from other SNADS components. It will also support workload distribution over networked processors.

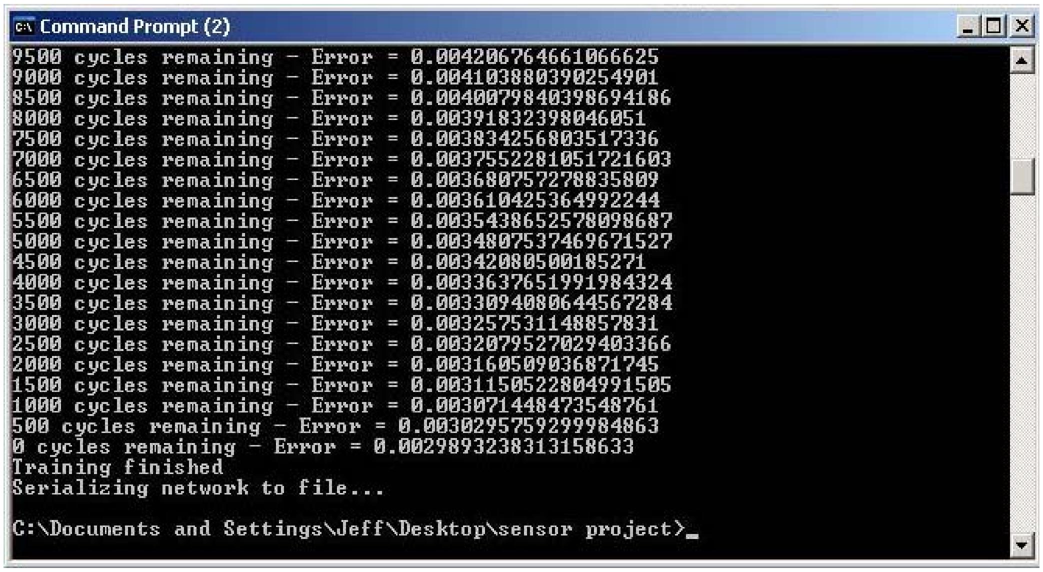
2) Sensor Configuration and Management

The sensor management subsystem generates and maintains a collection of JSensor objects, which provide the basic abstraction for sensors in the network. The sensor manifold object directly handles all incoming sensor notifications and prepares them for use by other components such as the ADS or GUI. This is the basic unit responsible for getting signals from the monitored objects and the environment and their possible preprocessing.

3) Graphical User Interface

The SNADS interface allows for simple central monitoring and management of the system itself as well as of the monitored object and the environment. It works in collaboration with other components and first of all, SCM. Sensors can be dropped from the network or individually configured. Sensor specific configuration dialogs integrate seamlessly into the rest of the GUI. Alerts from the ADS are reported the GUI, which notifies administrators of potential problems. Additionally, the GUI supports database, session, and ADS configuration.

The GUI allows SnadsComponents to be “wired” together in a way which captures the manner in which data flows through the system intuitively. Components are dragged from their component providers, on the right hand side (see fig. 1), into the working area on the left. There they may be dragged around and connections made by right clicking on a component and choosing, for example “Send Events To” on a Sensor Manifold and then clicking on a Data Display component to create an Event Source / Event Listener relationship between the two. These “wires” may only be connected in ways allowed by the system. A trick that is currently being implemented but not yet finished is creating an



```
9500 cycles remaining - Error = 0.004206764661066625
9000 cycles remaining - Error = 0.004103880390254901
8500 cycles remaining - Error = 0.004007984039869418 6
8000 cycles remaining - Error = 0.00391832398046051
7500 cycles remaining - Error = 0.003834256803517336
7000 cycles remaining - Error = 0.003755228105172160 3
6500 cycles remaining - Error = 0.003680757278835809
6000 cycles remaining - Error = 0.003610425364992244
5500 cycles remaining - Error = 0.003543865257809868 7
5000 cycles remaining - Error = 0.003480753746967152 7
4500 cycles remaining - Error = 0.00342080500185271
4000 cycles remaining - Error = 0.003363765199198432 4
3500 cycles remaining - Error = 0.003309408064456728 4
3000 cycles remaining - Error = 0.003257531148857831
2500 cycles remaining - Error = 0.003207952702940336 6
2000 cycles remaining - Error = 0.003160509036871745
1500 cycles remaining - Error = 0.003115052280499150 5
1000 cycles remaining - Error = 0.003071448473548761
500 cycles remaining - Error = 0.003029575929998486 3
0 cycles remaining - Error = 0.002989323831315863 3
Training finished
Serializing network to file...

C:\Documents and Settings\Jeff\Desktop\sensor project>_
```

Fig. 3. Neural network training display

"attraction" between the newly created, but not yet connected, end of one of these wires and a component to which it may be connected, and a repulsion away from any component that cannot be connected. The system would in this way provide a subtle cue to the user as to what connections would be wise.

While the GUI elements in figure 1 are standard to the distributed SNADS management, figure 2 shows a simple tabular data display implemented by an external component. In this particular example the Data Display tab was sent from a component on a remote machine, and its content is determined entirely remotely. This was not simple to achieve in Java's RMI scheme. It was not possible to render entire swing or remote objects because of the restrictions on the creation of dynamic proxy classes underlying the RMI system.

If the GUI component is an instantiated remote object it cannot be marshaled across the network through the RMI system without being replaced by a proxy, but if it is not then the SNADS component behind it cannot communicate to it. The current solution was to send a reference to the GUI component class across the network, creating a new instance (with the class loading handled by RMI automatically) and then calling methods on both the SNADS component and the GUI component to pass the remote references to one another. There may be a way to force a remote object to be marshaled in the normal way.

4) *Change or Anomaly Detection*

At the heart of SNADS lies the ADS. While a default component will be supplied, arbitrary user-defined ADS implementations can easily be used instead of the default. Based on sample data, the ADS looks for changes in the coming data streams, marks such data, and sends notification to the messaging center. These changes might originate from the actual signal novelty as well as erroneous or maliciously altered incoming sensor readings, ADS is designed as a collection of different agents of various intelligence levels and complexity. Examples of the implemented procedures include a comparison against fixed thresholds, an If/Then rules system, and a multilayer perceptron (MLP) neural network.

The Neural Network implementation used for this project utilizes an open source Java Object-Oriented Neural Engine (JOONE). JOONE appeared to be the best offering among the freely available neural network development tools for the Java language. JOONE also provides a graphical framework for testing neural network architectures [14]. The classes provided by JOONE can be utilized to implement complex and highly scalable neural networks. In the test cases a three layer MLP feed forward neural network has been implemented. The MyNeuralNet class builds this neural network, trains it according to a provided training set, and serializes it to a file for use by SNADS. MyNeuralNet implements a JOONE defined interface which enables an event triggered methodology that is helpful for displaying the status of the training phase.

The MyNeuralNet constructor creates an instance of the JOONE NeuralNet class. MyNeuralNet.Build defines the network's makeup (e.g. number of inputs, number of outputs, number of layers, number of nodes per layer, type of transfer functions, etc.), reads a training set from an external file, trains the network according to this training set, outputs training progress to the console and serializes the network to another file. The process needs be performed only once as it is this serialized neural network that is needed by the server program, although there is no harm in re-executing the program (and regenerating the serialized network) and in fact this is exactly what should be done if there is a need to change the behavior of the network. Figure 3 presents the neural network training process.

5) *Database*

The database subsystem provides a simple interface for data logging and searching. Each session is stored separately in the database and the format for sensor readings can be different across sessions. Data cataloged by the SNADS database system is easily accessible for later analysis if so desired. Depending on the specific SNADS application, however, the database may not be used at all.

- **SnadsEvent**
    - **SensorEvent**
        - **DataEvent**
        - **StatusEvent**
            - **Connect**
            - **Disconnect**
                - **SystemEvent**
                - **SnadsShutdown**

Two different types of EventListeners are available, for receiving SensorEvents or SystemEvents; they follow the normal Java EventListener pattern.

Fig. 4. Sensor event classes hierarchy

Database access is currently provided via the SnadsDatabase class. This will probably become an interface in a future revision so that various underlying database architectures can be more easily supported. At the moment, a MySQL database is used as the backing repository. When the database object is created, a network location for the database is specified. To connect, a username and password must be supplied along with a desired session name. The session name will name a table in the database to hold all readings in the session.

The MySQL database needs a pre-configuration in order to be used by SNADS. Eventually, this will be handled in the GUI database setup dialog. The program expects to use a database called "sensor". Sensor must contain a (probably empty) table called "templateA" from which it instantiates the session tables. Once the entire GUI is in place, it should be relatively easy to create tables that support different kinds of sensor readings. The templates will be fully editable in the GUI. Furthermore, a table called metadata will be added to the database setup to correlate specific session tables with the template they used.

III. SNADS Implementation features

A. *Current state and future development*

1) Java RMI implementation

The SNADS distributed framework is implemented using Java built in RMI methods, in java.rmi package. Interfaces were extracted from the original SNADS classes and modified to fit Java's RMI [15], which uses dynamic proxy classes in its own implementation, limiting it to the methods of interfaces, rather than both interface methods and class methods.

SNADS class hierarchy (see fig. 4 for events classes hierarchy) has been designed to better fit the distributed system and to allow for more generalized object management. Every class that would be accessed over the network was made a subclass of SNADS component, allowing for generalized object management and user interaction, both through a standardized configuration method and through standardized methods for the retrieval of GUI objects. The basic SNADS component class also implements (via the Named interface) methods for the identification and retrieval of components.

The ComponentProvider forms the basis for both the administrative console and the remote units. They are registered with Java's RMIRegistry, allowing the client to find them automatically. The client retrieves a set of components made available by a given provider to present to the user. Some components are only available once, such as a specific SensorFactory on a specific remote system, while others, such as BasicSensorManifold, may be instantiated multiple times. The current MultiComponent implementation leaves something to be desired, as it must create an instance of the object in order to offer it to the user, despite the fact that it may never be selected by the user for inclusion in a sensor system. If the creation of the object were to involve threads on a remote server, database connections or a large memory utilization would waste resources.

2) Partially Implemented Features

The message processing system is still in an infantile state. All message routings are currently hard coded, which defeats many of the system goals. Unfortunately, this system was not implemented first, as it should have been, but instead it was added after a minimally functional system had been created. The next milestone for SNADS involves properly implementing the message system to support dynamic message routing and type addition. A couple of database features are also missing at present. Specifically, the ability to select a sensor reading format template is not present, but the underlying code is in place. The change detection by the ADS is not yet indicated in the GUI. However, it is properly stored in the database. Almost everything is in place for the GUI to handle these events, but a last minute rethink of the setup prevented inclusion in this release.

3) Known Bugs

For some yet to be determined reason, the SNADS system (or at least the GUI) appears to hang when around 30 sensors are added to the network. Obviously, this is unacceptable and will be fixed in the next milestone.

4) The Next Step

The next major milestone for SNADS will include full GUI support for configuration of all major subsystems, a couple of bug fixes, and finalization of the messaging system. Also slated is the dynamic creation/destruction of sensor factories and a number of extensions in the database interface. Once anomaly reporting is fully integrated into the GUI, the first public release will be offered as v1.0.

B. *The JSensor Abstraction*

In order to support diverse sensor platforms, SNADS has a generic sensor abstraction. JSensor is an abstract class definition which provides functionality to disseminate incoming data and sensor status events. The basic JSensor also provides support for unique per session identification. Classes derived from JSensor must also support queries for sensor name, type, and capabilities as well as provide a configuration mechanism. Optionally, such classes may support sensor shutdown and renaming operations. By default, attempts to issue these commands result in an exception.

1) Session IDs

A session ID is assigned to each JSensor object via its constructor. Each ID issued must be unique within the current session in order to identify each sensor properly. Currently, sensor factories are charged with managing ID generation, however this may change in the future.

2) *Sensor Name and Type*

Each sensor has a name and type associated with it. There are no restrictions placed on these fields and they can be used by extending classes as they see fit. However, some standards may help. The type field should contain information about the actual kind of sensor, which the JSensor represents. For example, a type query for one of the simulated sensors used in testing would return a string like "TestSensor;vers=1.2;". Names do not have to be unique, but should describe the purpose of the sensor.

3) *Sensor Capabilities*

Sensor capabilities specify the non-standard functionality, which a sensor supports. A capability is specified in *Interface*.*method*() format. The standard capabilities are JSensor.shutdown() and JSensor.setName(); these do not have to be supported. As an example of extending the basic functionality set, the test sensors added TestSensor.skew() and TestSensor.unskew(). While these methods were only called from within the TestSensor configuration dialog, a SNADS module could theoretically utilize these methods directly after querying for them dynamically.

4) *Sensor Configuration*

The JSensor class has an abstract method called configure, which must be supported by deriving classes. All the method has to do is allowing for configuration of a specific sensor. Theoretically, if the sensor has no configuration options, this could result in no operation. While not a requirement, standard configuration methods create a GUI interface for a dynamic configuration. Any other approach should be well documented. The configuration procedure may be more strictly defined in a future revision of the SNADS specification.

C. *Sensor Factories*

Because of the varied underlying technologies, there is no single way to instantiate a new JSensor derived class. As such, each type derived from JSensor should have a corresponding sensor factory class, which implements the JSensorFactory interface. Basically, the sensor factory just waits for new sensor connections, creates a JSensor derived object, and adds it to the sensor manifold.

D. *Error Handling*

In the operation of the distributed network it is possible for entire nodes to become disconnected, leaving the proxies connected to Remote objects, talking to so much dead hardware. Originally the system was set to detect these events proactively and at the earliest possible time and propagate the appropriate LostComponent events through the messaging system. This required each component to listen for these events, and check the component identified as lost against all member variables of type SnadsComponent and members of any Collections, Arrays and the like used by the object. This system quickly grew to the size of the rest of the program, and added much complexity, as for example, in handling one lost component the system could come across another lost component, triggering another LostComponent message.

The replacement scheme used instead is not centralized, and does not rely on the messaging system. Every method of a Remote interface must be declared to throw a RemoteException, which will occur when the method is called on a component located on an unreachable node. At that time only the component triggering the exception is expected to remove its own references to the lost component. In this way each object deals with its loss on its own schedule, as it needs to. The code for dealing with the loss of an object is located where the object is used, so the object catching the RemoteException does not have to search through a number of locations to find the references. If the author of a future component wished to locate all lost objects' code in a single method, he could call that method whenever a RemoteException is caught.

IV. System Configuration and Compilation

A. *Network Configuration Persistence*

The network structure created in the work area of the management GUI can be saved to disk and reloaded later. Using the named interface of the components and the providers, they came from, a fully qualified component name is constructed and saved. During loading the system looks for the same component provider (by name) and gets the same component (again by name) from it. In the event that the specific provider is not available the system will search all providers until it finds a matching component.

Currently version information is a part of a component's name. It would be better to store this separately so that the loading process could load a newer version of a component if it has been upgraded. If a component is not available at all, the load should either fail or load the network with placeholders for the missing pieces. This state is not currently handled.

B. *Component Upgrade*

To integrate a new version of a component into the system either a RemoteComponentProvider must be restarted or a new one started. Hitting "refresh" on the console will show the new information.

C. *Compilation*

The SNADS distributed framework was developed and compiled under Eclipse Version 2.1.3 built with the RMI Plug-in for Eclipse version 1.5.1.1 from http://www.genady.net/rmi/. This RMI plug-in is a commercial software, however its only purpose was to add the RMI Stub generation for certain classes to the Eclipse build process without any effort. It is possible to build the SNADS distributed framework using only Java's command line tools.(See for details on Rmic
http://java.sun.com/j2se/1.3/docs/tooldocs/win32/rmic.html)
Java RMI has the following known drawbacks.

- Need for base class and interface, i.e. SNADSComponent and BasicSnadsComponent
- Need for RMI compiler to create subclasses, which must be sent over the network and loaded via a class

loader at run time, even though RMI uses only interfaces.

- Difficult to send a real object to another machine and then obtain a remote interface for it, as in a GUI.

V. Conclusion

SNADS is designed as a synergetic collection of heterogeneous computational intelligence and machine learning agents. Its dynamic structure allows for adding up new modules realizing new intelligent techniques previously unavailable and scaling it up to really big numbers of the modules and agents. It is realized in Java that allows for its application with novel Sun Worldspot sensor kits currently available on the market [16].

While still in its early stages, SNADS demonstrates a great deal of potential for the future developments and applications. By facilitating general sensor network security and management even across heterogeneous networks, such tasks will become more accessible. There is an obvious set of scenarios, in which SNADS would be of great use, especially for objects, processes and environment monitoring. By allowing a user to more quickly and easily create SN configurations, SNADS will be invaluable to scientists using sensor networks to monitor objects and the environment. Additionally, the ADS elements can help to identify results that need to be more closely scrutinized to ensure that the data set is not polluted by errors or attack.

Besides the obvious applications in traditional sensor networks, SNADS has a high potential in other areas as well. For example, traditional computer intrusion detection systems could be augmented/replaced by a network of simulated sensors. Each sensor would sit on a separate processor and report statistics back to the SNADS processing center.

References

[1] Reznik L., Von Pless G., and Al Karim T. “Application testing of novel neural network structures” Workshop on Building Computational Intelligence and Machine Learning Virtual Organizations, Fairfax, VA, October 2008

[2] Xiaolong D. and Khorram S. “Development of a new automated land cover change detection system from remotely sensed imagery based on artificial neural networks”, IEEE International Geoscience and Remote Sensing, 'Remote Sensing - A Scientific Vision for Sustainable Development'., 1997, 3-8 Aug. 1997 Singapore, vol.2, pp.1029 - 1031

[3] Han M., Xi J., Xu S., and Yin F.-L. “Prediction of chaotic time series based on the recurrent predictor neural network”, IEEE Transactions on Signal Processing, vol. 52, Issue 12 , Dec. 2004 pp.:3409 – 3416

[4] Csorgo M. and Horvath L. “Limit Theorems in Change-Point Analysis” New York: John Wiley & Sons, 1997

[5] Majeed B. , Nauck D., Lee B.-S., and Martin T. “Intelligent systems for wellbeing monitoring”, Proceedings of 2nd International IEEE Conference on Intelligent Systems BT Exact Intelligent Syst. Lab. Res. & Venturing, British Telecom plc, Ipswich, UK, 22-24 June 2004, Vol.1, pp. 164 - 168

[6] Shan Q., Liu Y., Prosser G. and Brown D. “Wireless intelligent sensor networks for refrigerated vehicle”, Proceedings of the IEEE 6th Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication, 2004, 31 May-2 June 2004 , Vol.2, pp. 525 – 528

[7] Reznik A.M., Shirshov, Yu.M. Snopok, B.A. Nowicki, D.W. Dekhtyarenko A.K. “Associative memories for chemical sensing” Proceedings of the 9th International Conference on Neural Information Processing, ICONIP '02, vol.5 , 18-22 Nov. 2002, pp. 2630 – 2634

[8] Reznik L. and Dabke K.P. “Evaluation of Uncertainty in Measurement: A Proposal for Application of Intelligent Methods” in H. Imai/Ed. Measurement to Improve Quality of Life in the 21st Century, IMEKO – XV World Congress, June 13-18, 1999, Osaka, Japan, vol. II, p.93-99

[9] L. Reznik and Pham B. “Fuzzy Models in Evaluation of Information Uncertainty in Engineering and Technology Applications”, The 10th IEEE International Conference on Fuzzy Systems, December 2-5, 2001, Melbourne, Australia, vol.3, pp.972-975

[10] Reznik L. and Dabke K.P. “Measurement Models: Application of Intelligent Methods”, Measurement, vol. 35, No.1, pp.47 – 58, 2004

[11] Mari L and Reznik L. “Uncertainty in Measurement: Some Thoughts about its Expressing and Processing”, In L.Reznik and V.Kreinovich/Eds. *Soft Computing in Measurement and Information Acquisition*, Springer, Berlin-Heidelberg-New York, 2003, ISBN 3-540-00246- 4, pp. 1-9

[12] Reznik L. Which models should be applied to measure computer security and information assurance? Proceedings of the FUZZ '03, The 12th IEEE International Conference on Fuzzy Systems, St.Louis, May 25- 28, 2003, IEEE, vol. 2, pp. 1243-1248

[13] TinyOS Community forum at http://www.tinyos.net/, retrieved on September 1, 2008

[14] JOONE – Java object oriented neural engine, http://www.jooneworld.com/, retrieved on September 1, 2008

[15] D. Kurtis “Java, RMI, and Corba”, White paper, available at http://www.omg.org/library/wpjava.html retrieved on September 1, 2008

[16] Project Sun Spot at http://www.sunspotworld.com/products/, retrieved on September 1, 2008.

Application Testing of Novel Neural Network Structures

Leon Reznik, Gregory Von Pless, and Tayeb Al Karim

Abstract— **The paper describes applications of a modified time-based multilayer perceptron (MTBMLP), which is a complex structure composed from a few time-based multilayer perceptrons with a reduced connectivity. The modification reduces connections, isolates information for each function and produces knowledge about the system of functions as a whole. This neural network is applied for change detection in signals delivered by sensor networks and for edge detection in image processing. In both applications a MTBMLP is utilized for function predictions and, after a further structure development is implemented, for an error prediction also. In sensor network applications, a number of experiments with Crossbow sensor kits and the MTBMLP acting as a function predictor have been conducted and analyzed for detecting a significant change in signals of various shapes and nature. A series of experiments with Lena image have been conducted for edge detection applications. The results demonstrate that MTBMLP is more efficient and reliable than other methodologies in sensor network change detection and that its application in in edge detection is also feasible.**

Index Terms—**Sensor networks; Signal processing; Image processing; Java programming**

I. Introduction

Many problems in signal processing applications, such as change or anomaly detection in the objects monitored by sensor networks or edge detection in image processing may be addressed using neural network function predictors. Being a universal function approximator, neural networks have proved their feasibility in predicting a variety of signals in different applications ranging from predicting stock prices to weather forecast. Basic neural network topologies including multilayer perceptron (MLP) and radial basis functions (RBF) have been studied and are currently provided by design packages, of a general nature such as Matlab and Mathematica as well as specialized software tools. However, while achieving a good performance in many instances, these basic topologies might consume significant resources that obstructs their use in real time and embedded system applications. In looking at the problem of predicting many functions at once, and specifically predicting many sinusoidal functions simultaneously, some limitations of the most common neural network function predictor, the time-based Multilayer Perceptron, have come to light. First, this predictor rarely learns to predict a sinusoidal curve in a reasonable amount of time. Second, when predicting many signals with one network, this technology is hardly scalable, as a large number of weights requires a great deal of time to process and adjust them.

In order to extend applicability of neural networks in real life applications and expand the market of intelligent methodologies users, the computational intelligence and machine learning community has to develop and examine novel neural network topologies and structures, which are able to provide a good functional performance under the strict limitations of consumed resources. The availability of novel algorithms and the corresponding libraries on-line may become a key factor for an appearance of a new wave of neural network based applications, in particular in such important areas as networking and image processing technologies. The paper presents and investigates in real time applications a novel neural network structure named modified time based multilayer perceptron (MTBMLP) described in section II. It is included as a novel intelligent agent into a software framework SNADS [1] currently being developed as a symbiotic tool integrating a variety of computational intelligence methodologies for sensor networks and other real time and pervasive computing applications.

In this paper we study a set of modifications to the time-based multilayer perceptron that address the aforementioned concerns. The resulting predictor is able to reliably and accurately predict a wide variety of functions, including sinusoids, and is able to do so after a very short training period due to reduced connectivity. This paper examines predictor quality for sensor network and image processing applications. The later sections discuss two possible applications of this new predictor: sensor network novelty detection (Section III) and image edge map generation (Section V). Some results obtained in each application are presented along with the performance comparison between the new predictor and its predecessor.

Anomaly detection in this paper is understood as an identification of unforeseen change in general characteristics and parameters of the signal observed [2,3]. The case involving time series is one of the most interesting applications of the problem, attracting much attention [3-5], while being a very challenging research topic. A number of methodologies for outlier detection have been developed over the years. Almost all of them so far have the following weaknesses [6]:

Authors are with the Department of Computer Science, Rochester Institute of Technology , 102 Lomb Memorial Drive, Rochester, NY 14623 USA (corresponding author phone: 585-475-7210; fax: 585-475-7100; e-mail: lr@cs.rit.edu).

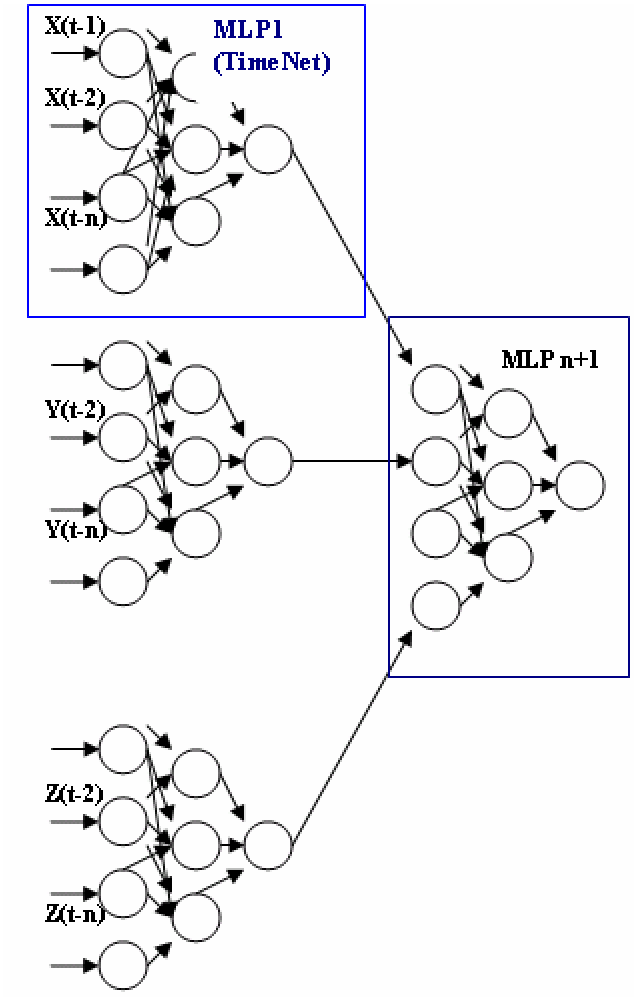


Fig. 1. Distributed time-delay neural network architecture based on MLP. Inputs to the first layer are delayed samples of sensor signals X(t), Y(t), Z(t)

1. difficulty in automated threshold determination,

2. failure to specify a generic methodology that is applicable across applications without a priori knowledge of known data distributions, and

3. failure to specify an effective incremental classifier retraining procedure.

As well as anomaly detection, the system goal could be to detect violations of confidentiality, integrity and availability of sensor data in complex network systems by comparison of currently acquired results against an association model derived during execution. The detector operation should minimize probabilities of missing attacks and/or false alarms (recognizing a “normal” situation, which is due to the inaccuracy of measurement results and association information, as an attack). To solve this problem one has to take into account that no data, which originate from measurements, are absolutely precise. Normally measurement results fluctuate with some degree, which does not constitute any malicious action or malfunctioning. Also, the association model may not be accurate, or may include some vagueness and uncertainty. The degree of its uncertainty may vary depending on the methods applied for its derivation.

The goal in edge map generation is to determine where object outlines exist in images, so that objects themselves may be distinguished from one another. Edges can be detected from color changes, shadow and luminance effects, and the interpretation of other effects implying the separation between objects. Some classical approaches to edge detection are discussed in [7]. These have been implemented to get the results shown below. Some other interesting solutions that are not used for comparison in this paper, but do demonstrate other methods for applying neural network technologies to this domain are discussed in [8-10].

II. The Modified Time-Based Multilayer Perceptron

A. *Multilayer Perceptron*

Multilayer Perceptrons (MLPs) are neural networks that have nodes arranged in multiple layers, with connections between neurons from one layer to the next. Data are propagated forward through the network to produce an output. An error is determined for the output and distributed backwards through the layers. The errors are used to adjust the connection weights between nodes. The backpropagation algorithm allows for reaching the minimal error over the course of the training process. Each time the weights are changed, the direction and magnitude of their change is determined so as to make a move towards the minimal error.

B. *Time-Based Multilayer Perceptron*

A time-based MLP is simply a backpropagation-trained MLP whose inputs are time-delayed values. Time-based MLPs are widely used as function predictors.

C. *Modified Time-Based Multilayer Perceptron*

The Modified Time-Based Multilayer Perceptron (MTBMLP) consists of multiple time-based MLPs (referred to in this paper as TimeNets) all connected to a single end MLP (the MainNet). Each TimeNet is associated with a single function. The MainNet is responsible for predicting the next value for all functions (see Fig. 1).

D. *Explanation of the Modification*

The purpose for altering the standard time-based MLP structure is threefold. The modification reduces connections, isolates knowledge for each function, and produces knowledge about the system of functions as a whole.

First, the number of connections, as compared to a standard MLP structure with an equal number of nodes in the same layers, is reduced significantly. The benefit reaped from this is an increase in propagation speed over the network, as nodes have fewer inputs and fewer weights need to be trained. This increase in speed makes the network more practically applicable in a real-time setting.

Second, by handling some of the time-series prediction in separate networks, before converging all the information in the network, some knowledge of a given function is stored in weights that do not connect to parts of the whole network not associated with the specific function. This separation of knowledge makes the system more robust, as when a single function changes, the composite knowledge in the end MLP

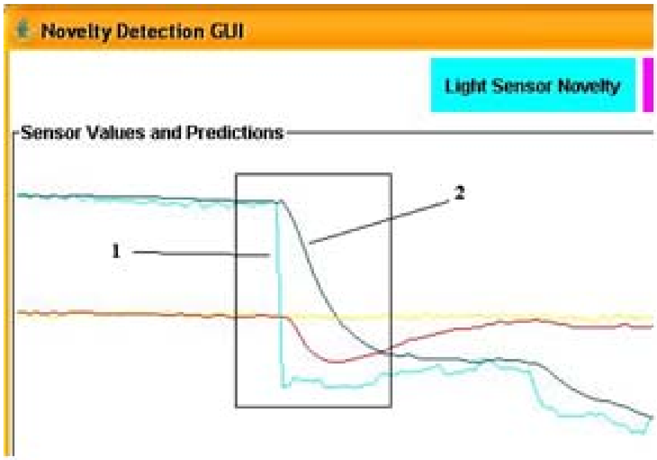


Fig. 2. Change detection screen shot. The area within the black box shows where the light value(1) has changed, due to the lights turning off. The curve(2) that follows this change is the network's prediction. The box at top center that says "Light Sensor Novelty" is a different color because the application changes the color here when change is detected.

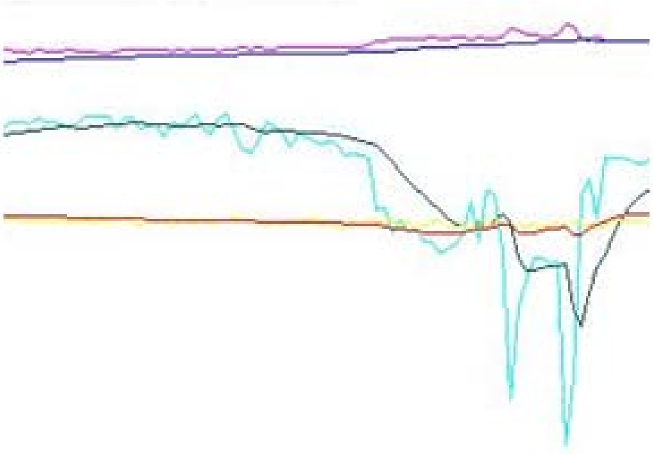


Fig. 3. The topmost curves show a gradual temperature change and the predictor adapting to that change. The experiment was conducted indoors with doors to the outside open, allowing an air current to come in. The results shown above occurred once the doors were closed and the room heated up again.

can be altered to fit the change more quickly than in a standard topology because knowledge of the other functions is not significantly affected.

Finally, converging the separate networks into a single MLP allows the network to keep the knowledge of the entire system of functions that a fully connected MLP would have. Thus, prediction is based in part upon the interaction of the various functions, as well as the previous values for each individual function.

The selection of threshold value is an important consideration in a MLP architecture. According to Li et al. [11], the best threshold should ideally maximize a function based on the log of the actual output of the network and the threshold output. They show that using 0.5 as a threshold on a given output node is not a good way since test data may lie outside the input data space.

III. CHANGE DETECTION EXPERIMENTS IN SENSOR NETWORKS

A. *Problem Description*

The change detection should be made based on comparison of imprecise data against uncertain models. It is clear that an attack detection problem, as well as a malfunctioning detection problem could be reformulated as the problem of distinguishing between a rather small discrepancy between the measurement results and the prediction model and a large difference between them. The solution could be found by drawing the border lines to separate an acceptable and unacceptable region. The border position will be influenced by the sensor metrological characteristics, which can be retrieved by standard methods of metrological analysis as well as by the characteristics of the association information, which are harder to obtain and analyze.

B. *Procedure Description*

In order to detect novelty, an MTBMLP is utilized for function prediction. The MTBMLP is trained in real time to predict the next outputs for each sensor in the network. A predefined threshold is used to determine if the difference between the prediction and the actual next value is acceptable based on the threshold comparison. Finally, an error goal is set, specifying how low the MTBMLP's mean squared error (MSE) must be in order for it to skip the training process on a specific iteration. This last requirement speeds up processing by skipping the expensive backpropagation algorithm when the network's prediction is below the error goal. At the same time, habituation is not lost because as soon as the MSE jumps, the network is required to train again.

C. *Detecting Novelty in Light Intensity Data*

Figure 2 shows a screenshot from the application developed to report sensor data and network prediction data during experimentation. The experiments utilized Crossbow Inc. Mote kits (MICA type), including a light sensor, an ultrasound sensor, and a temperature sensor. Most of the experiments conducted involved light intensity changes, as of the three aforementioned data types, light data are the most easily manipulated.

The MTBMLP-based system is able to detect a light change from on to off and vice versa 100% of the time, as this problem is relatively simple. Also, the system is able to adapt to gradual changes without detecting novelty, as shown in Figure 3. Depending upon the application of the system, this may or may not be a problem. For instance, for a sensor detecting outdoor light levels, this quick adaptation would be beneficial in that the setting sun would not be considered novel. However, as in the case shown in Figure 3, adaptation can cause significant changes to be overlooked. The slow change to temperature shown in the screenshot was caused by opening two doors, allowing an air current from outside (where it was below freezing) to blow through the room. Most likely, detection of this temperature change, which alerts the user that the doors are open, is desirable.

Table 1. Data from sensor simulations using sine curves. Some tests involved correlated data and some involved uncorrelated data. The network performed better with correlated data. Also, multiple thresholds were explored. The numbers represent false alarm rates and detection rates for various tests at various thresholds. UC stands for uncorrelated, C stands for correlated, and "Sensor Fails" means that in that test one sensor becomes a flat line at 0.0.

Thresholds:	0.1	0.15	0.2	0.25	0.3
	False Alarm Rates, Averaged for all Training Times				
Sensor Fails – UC	2.640%	1.098%	0.569%	0.298%	0.170%
Frequency Changes – UC	3.350%	0.832%	0.725%	0.451%	0.272%
Amplitude Changes – UC	3.075%	1.307%	0.758%	0.359%	0.167%
Frequency Changes – C	1.39%	0.48%	0.29%	0.19%	0.12%
Amplitude Changes – C	0.27%	0.10%	0.06%	0.04%	0.02%
	Detection Rates, Averaged for all Training Times				
Sensor Fails – UC	93.35%	92.52%	85.83%	86.65%	82.48%
Frequency Changes – UC	95.00%	87.50%	85.82%	82.50%	75.82%
Amplitude Changes – UC	94.17%	90.83%	85.00%	75.00%	65.00%
Frequency Changes – C	100.00%	100.00%	100.00%	93.50%	91.16%
Amplitude Changes – C	98.16%	85.67%	74.73%	52.05%	48.03%

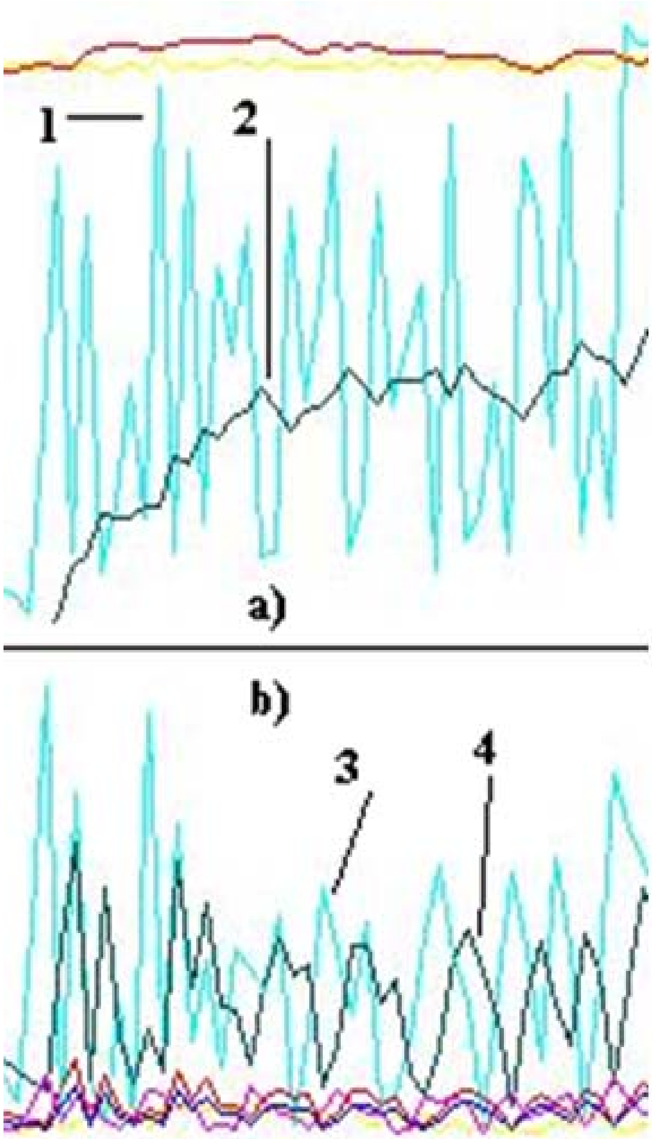


Fig. 4. Light intensity values, predictions, errors, and error predictions from experiments conducted indoors using Crossbow Inc.Telos v.B type sensor kits. a) Light intensity values(1) recorded while lights were flickering along with predictions(2) of those values; b) prediction errors(3) and the predictions of those errors(4). The error predictions are far better than the value predictions.

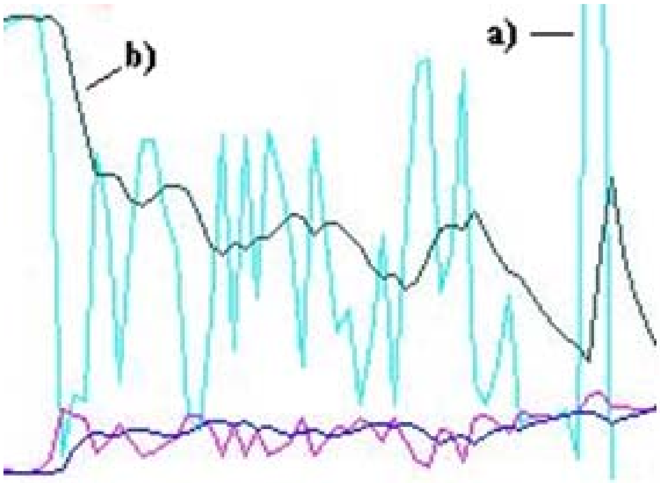


Fig. 5. Results of experiments conducted indoors using Crossbow Inc. Mote sensor kits. a) Light sensor output when the light is flickering; b) MTBMLP's prediction. At high flicker frequencies, the prediction becomes a sort of best fit curve.

D. Performance Versus a Standard MLP

It is expected that an MTBMLP-based system will demonstrate a comparable performance with an MLP-based system, since the MTBMLP is designed to perform a prediction on multiple functions at once, with faster convergence (see Table 1). In order to confirm this expectation, all of the sensor data used to generate the figures in this section and the next were stored and used to run an experiment with systems of each type. The data feature light changes from on to off, from off to on, and flickering. Also, the temperature changes caused by opening and closing doors (letting in very cold air) are included in this data.

A system of each type was run with this saved data being supplied as inputs and the predictions for each value were recorded. The networks involved had the same number of nodes, with the MTBMLP's nodes linked as described in Section II, and the MLP's layers fully connected. Each network was given five past samples to work with for each sensor. Then, the mean error after the removal of outliers was calculated. Outliers were determined using these formulae:

f_s = upper fourth – lower fourth (1)

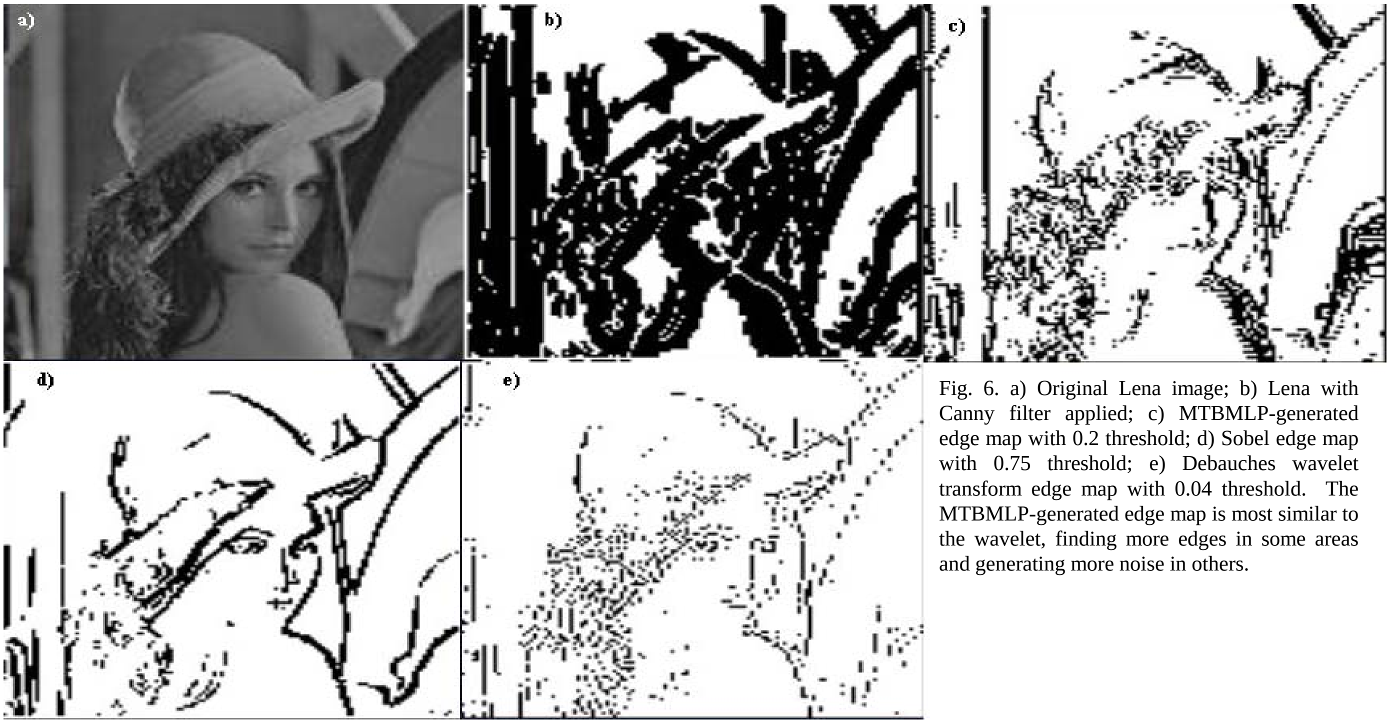


Fig. 6. a) Original Lena image; b) Lena with Canny filter applied; c) MTBMLP-generated edge map with 0.2 threshold; d) Sobel edge map with 0.75 threshold; e) Debauches wavelet transform edge map with 0.04 threshold. The MTBMLP-generated edge map is most similar to the wavelet, finding more edges in some areas and generating more noise in others.

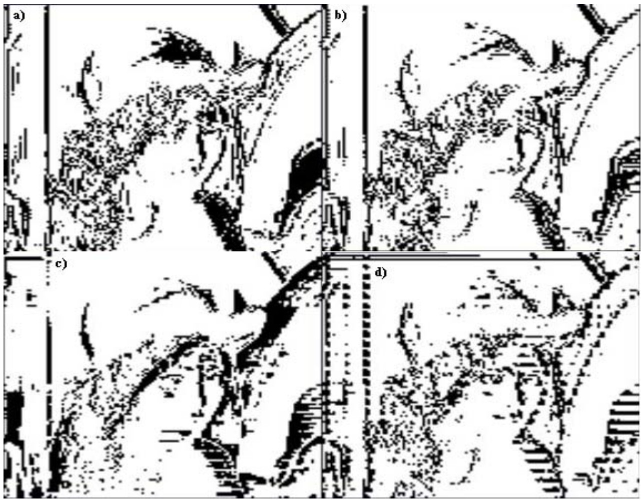


Fig. 7. a) MLP-generated edge map with threshold 0.2 and band width 2; b) MTBMLP-generated edge map with the same parameters; c) MLP-generated edge map with threshold 0.35 and band width 5; d) MTBMLP-generated edge map with threshold 0.2 and band width 5. The threshold for c) had to be increased because at 0.2 it is too noisy.

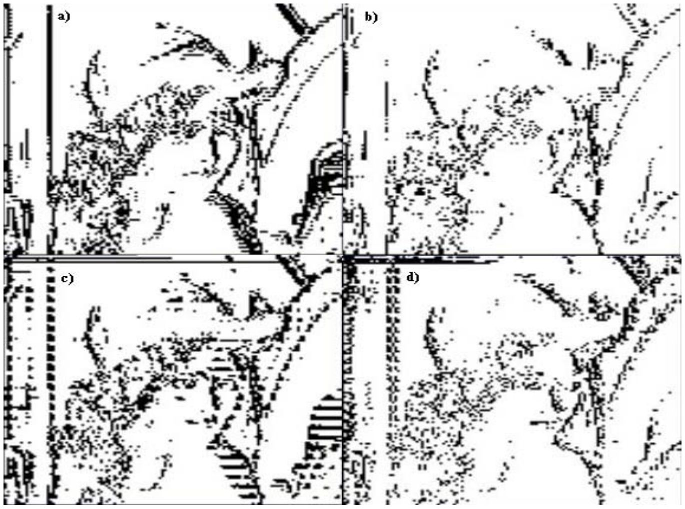


Fig. 8. a) MTBMLP without error predictor and band width 2; b) With error predictor and band width 2; c) Without error predictor and band width 5; d) with error predictor and band width 5. All use threshold 0.2.

upper limit = median + 1.5f_s (2)
lower limit = median – 1.5f_s (3)

The means for MLP and MTBMLP errors between their respective upper and lower limits are, respectively:

MLP: 0.00844
MTBMLP: 0.00696

This shows that the MTBMLP predicts the sensor data even more accurately.

IV. Error Prediction

The procedure described in the previous section suffers from the following limitation: it cannot properly predict discontinuous curves, such as clock signals; a real world example would be a flickering light. However, a solution to this problem has been implemented and is discussed below.

A. Description and Rationalization

Detecting change in a data stream using an MLP-type neural network for a prediction and threshold comparison was more efficient for an input-output mapping representing a continuous function than with a discontinuous one.

If one analyzes the error produced by subtracting the actual signal from the prediction of the signal (given by the neural network), this error curve will be continuous and cyclic, as the network, in attempting to learn each "step" of the function, will lag behind the actual signal. The error function can by modeled by a second neural network in a more reliable manner, because the MTBMLP is designed to predict cyclic, continuous functions. As this error function should only spike when novelty occurs, this methodology might allow for a more accurate change detection in discontinuous functions.

B. *Results With Error Prediction*

Figure 4 shows mostly successful error prediction as the light sensor reports flickering. The first network's value predictions appear as in Figure 5, while here predictions are much more accurate. Thus, change is reported on the basis of a difference between error and the prediction thereof, rather than value/prediction differences, allowing the system to properly adapt to repeated, yet discontinuous, trends.

V. Edge Map Generation

A. *MTBMLP Prediction Model versus Well-Known Methods*

Neural network function prediction can be applied to images to create edge maps. By using pixel intensity values as the output values for functions, the change detection procedure can be applied, with the prediction errors representing the degree of intensity change across an edge. To generate these edge maps, a certain number of rows were taken at a time, with each one being used as an input function to an MTBMLP. The number of rows used at once is termed the "band." The minimum band used is 2, so that correlations can be explored. The same network is used for each set of rows.

In order to illustrate the effectiveness of this edge detection methodology, edge maps were created using the well-known "Lena" image as input. Figure 6 shows the results of edge detection using an MTBMLP predictor, a Sobel detector, a Wavelet detector, and a Canny detector [7]. One can see that the MTBMLP approach does not yield the best possible edge map, however it outperforms the Canny filter and it yields a map comparable to the Wavelet map.

B. *MTBMLP versus MLP*

The MTBMLP also outperforms the same algorithm implemented with a standard MLP, as shown in Figure 7. The MLP generates more noise, especially at higher bands. The MLP also takes longer to run, because it has many more links.

C. *MTBMLP with Error Prediction versus MTBMLP Without*

The above examples all feature MTBMLP-based system with a second error prediction network. Since this method greatly improves the performance of the MTBMLP system in novelty detection, an exploration of its effect on edge map generation was performed. Figure 8 shows edge maps from an MTBMLP and an MTBMLP with an error predictor at band widths two and five. In both cases, the second network removes some amount of noise, creating a more useful edge map.

VI. Conclusion

The Modified Time-Based Multilayer Perceptron is a powerful function predictor that converges quickly and can learn to accurately predict cyclic, sinusoid functions faster and in some sense better than a conventional Time-Based Multilayer Perceptron. This neural network is included as one of the main agent into Sensor Network Anomaly Detection System [1]. Its application allows not only for achieving a high detection performance but also for a significant reduction in resource consumption.

The MTBMLP fits perfectly into a prediction model that allows for an accurate novelty detection in sensor networks. The predictor allows for a habituation, a robust prediction, and the ability to learn correlations between data sources. With the addition of an error predictor, this system can properly habituate to repeated discontinuous functions.

Experimentation has revealed that the MTBMLP prediction model could be applied not only in sensor networks but also in other applications like image processing. Edge maps that are far more useful than Canny edge maps can be obtained, however in our experiments it has not produced the best results in comparison to the Sobel algorithm. Rather than detecting edges, the prediction model could probably be successfully applied to image novelty detection as over time it will learn to perform its job better.

References

[1] Reznik L. and Hoffman C., "Development of the intelligent sensor network anomaly detection system: problems and solutions", Workshop on Building Computational Intelligence and Machine Learning Virtual Organizations, Fairfax, VA, October 2008

[2] Ma J., Perkins S., "Time-series novelty detection using one-class support vector machines", Proceedings of the International Joint Conference on Neural Networks, 20-24 July 2003, pp. 1741 – 1745.

[3] Dasgupta D., Forrest S., "Novelty detection in time series data using ideas from immunology", In Proceedings of the 5th International Conference on Intelligent systems, Reno, Nevada, June 19-21, 1996.

[4] Keogh E., Lonardi S., Chiu W., "Finding surprising patterns in a time series database in linear time and space", Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Canada, July 23-26, 2002, pp. 550-556.

[5] Guralnik V., Srivastave J., "Event detection from time series data", Proceedings of the International Conference on Knowledge Discovery and Data Mining, San Diego, Ca, 1999.

[6] Singh S., Markou, M., "An approach to novelty detection applied to the classification of image regions", IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 4, April 2004, pp. 396 – 407.

[7] Shapiro L.G. and Stockman G. C., "Computer Vision", Princeton Hall, NJ, 2001, pp 141 - 157

[8] Zhengquan H.and Siyal, M.Y., "Edge detection with BP neural networks", Proceedings of the 1998 Fourth International Conference on Signal Processing, Vol. 2, October 1998, pp. 1382 – 1384.

[9] De Silva C.R., De Silva L.C., Ranganath S., "An edge detection scheme using radial basis function networks", Proceedings of the 2000 IEEE Signal Processing Society Workshop, Vol. 2, December 2000, pp. 604 – 613.

[10] Uchiyama Y., Haseyama, M., Kitajima, H., "Hopfield neural networks for edge detection", IEEE International Symposium on Circuits and Systems 2001, Vol. 2, May 2001, pp. 608 – 611

[11] Li Y., Pont M.J., and Jones N.B., "Improving the Performance of the Radial Basis Function Classifiers in Condition Monitoring and Fault Diagnosis Applications where 'Unknown' Faults May Occur", Pattern Recognition Letters, vol. 23, 2002, pp. 569-577.

Collaborative Experimentation Using Agent-based Simulation

Jan D. Gehrke
Center for Computing Technologies – TZI
Universität Bremen
Bremen, Germany
jgehrke@tzi.de

Abstract— **This paper presents the PlaSMA multiagent-based simulation system and its application for collaborative experimentation. PlaSMA enables large-scale parallel and distributed simulations and relies on Java for platform independence. The applied agent-based simulation approach particularly suits virtual organizations in research because it enables shared computing resources as well as collaborative experimentation with common simulation scenarios.**

Multiagent-based simulation, intelligent agents, distributed computing, collaborative experimentation

I. Introduction

Simulation is a widely used means to evaluate the performance and adaptation abilities of systems in dynamic environments. For many domains, such as logistics or communication networks, discrete event simulation (DES) has been the predominant simulation technique. But for complex models, sequential simulation will exhibit poor runtime performance. Thus, parallel distributed simulation systems [6] have been developed that enable the integration of simulation hardware as well as different simulation systems. The latter is the focus of the *High Level Architecture* (HLA) [9] that defines services and interface to couple multiple simulation systems with potentially different simulation purposes.

Another approach to distributed simulation of complex systems is provided by *multiagent-based simulation* (MABS). In MABS, the simulation model is composed of multiple autonomous but interacting software agents [19] that may run concurrently and distributed over multiple computers. It is the general idea of MABS to model agents as a direct and natural mapping of the simulated real-world entities acting in the simulation. This is particularly useful if the simulated world consists of multiple technical systems, organizations, and/or humans. Consequently, a major application domain of MABS has been social simulation [4, 15].

Beyond simulation distribution, the model decomposition to agents in multiagent-based simulation allows for easy extension, integration, and substitution of agents. Researchers or other system users that agree on an environment and interaction model for the domain of interest can add their own agents in a common scenario. The respective agents could be evaluated in separate simulation experiments or even concurrently in a competitive manner. The latter case, for instance, is applied in the RoboCup soccer simulation league [13], while the USARSim system of RoboCup rescue league [3] rather pertains to the first case. However, the RoboCup simulations do not focus on distributed simulations and are limited to a specific simulation domain.

In this paper we will present the PlaSMA simulation system which enables distributed multiagent-based simulation based on established technologies for implementing software agents. PlaSMA has been used particularly in the logistics domain to evaluate different approaches of self-organized decision-making as well as robustness and adaptation of transportation processes in dynamic environments. The PlaSMA system architecture facilitates collaborative experimentation in terms of large-scale simulations with shared computing resources and experiments consisting of agents deployed by different users. The different agents can compete in a common reference simulation environment model. Users can monitor running simulations with multiple remote clients.

The remainder of this paper is organized as follows: Sect. 2 introduces the concept of multiagent-based simulation. Sect. 3 presents the PlaSMA simulation system and its applications. The collaborative experimentation process with PlaSMA is discussed in Sect. 4. Sect. 5 concludes the paper.

II. Multiagent-based Simulation

Multiagent-based simulation combines distributed discrete event or time-stepped simulation with decision-making encapsulated in agents as separate and concurrent logical processes [15, 12]. In classical parallel and distributed simulation systems, the logical processes involved as well as interaction links have to be known in advance and must not change during simulation [6]. By contrast, this does not hold for MABS because each agent may interact with all other agents [12]. Agents may join or leave simulation during execution, e.g., depending on a stochastic simulation model or human user interaction. On the other hand, this flexibility complicates simulation time management.

A. *Notions and Models of Time*

It is necessary to distinguish different notions of time related to MABS. Generally, *physical time* refers to the time at

DFG grant “Collaborative Research Centre 637” (SFB 637)

which simulated events happen in the real world. *Simulation time* models physical time in simulation. *Wall-clock time* refers to the time that is consumed by the simulation program executing the simulation. As simulation speed might differ between agents, each of them has its own *local virtual (simulation) time* or in short LVT [11].

In contrast to equation-based modeling (cf. [15]), time progression is discrete in MABS. Equidistant steps, as applied in time-stepped simulation, are hardly efficient since time steps are even processed if no events occur [6] and inadequate if time steps are too long because events are considered simultaneous although there would not be in physical time [8]. By contrast, in discrete event simulation (DES) time steps bridge the gap until the next event occurs. Thus, DES is the dominant approach in most areas. However, some MABS systems still use time-stepped simulation because it is easier to implement and understand.

B. *Synchronization*

Agents in MABS can be simulated on distributed platforms, which may differ regarding their computational power. Hence, simulation time progression depends on the CPU performance of the respective platform. However, even on one platform the local virtual time of agents may diverge, if their computational demands differ. Problems may arise whenever agents interact. For instance, consider an agent passing a message to another agent that is advanced in its local virtual time. The recipient of such a so-called *straggler message* might have taken other decisions if it were aware of that message on time. This is denoted as the *causality problem* [6]. In order to guarantee correct and reproducible simulations, the simulation system has to ensure that agents process events in accordance to their timestamp order.

This problem is addressed by synchronization, which can be either optimistic or conservative. In general, progression of local virtual time is not restricted for agents in optimistic synchronization. This allows executing simulations efficiently since fast processes do not have to wait for slower ones. Whenever an agent receives a straggler message with time stamp *t* a *rollback* is conducted that resets the agent to its former state at LVT = *t*. However, optimistic synchronization is more complicated in implementation and has potentially high requirements regarding space [6]. Conducting a rollback might require returning many steps back in time. Hence, all preceding states of every agent must be stored.

By contrast, conservative synchronization prevents causality problems by ensuring the correct order of event processing at each time. This, in turn, restricts the speedup achievable by parallelism. There are numerous approaches to conservative synchronization which are described in [6]. The synchronization mechanism is an important choice when implementing a MABS system. The PlaSMA system, which is presented in the next section, applies coordinated conservative synchronization with a simulation controller hierarchy.

III. PLASMA SIMULATION PLATFORM

The PlaSMA system[1] provides a distributed multiagent-based simulation and demonstration system based on the FIPA-compliant Java Agent Development Framework JADE [2]. PlaSMA stands for *Platform for Simulations with Multiple Agents*. The system is developed at the University of Bremen, Germany, as part of the Collaborative Research Center 637 “Autonomous Cooperating Logistic Processes”. Within this interdisciplinary center, the PlaSMA system is the joint software platform for autonomous logistics applications and evaluations. Although the primary application domain is logistics, PlaSMA is applicable for other simulation domains as well.

A. *Architecture*

The PlaSMA system consists of the basic components *simulation control*, *world model*, *simulation agents*, *analysis*, and *user interface*. The simulation control handles world model initialization, simulation time management as well as agent lifecycle management. Further-more, it provides an interface for world model access. Simulation control primarily consists of two kinds of instances: one top-level controller and a sub-controller for each processor or computer in distributed settings. Sub-controllers handle the actual software agents (called simulation agents) that are the actors in a simulation scenario. The interaction between top-controller and sub-controllers concerns agent lifecycle management, runtime control, and time events. For time management, the top-controller sends time events to all sub-controllers to indicate progression of simulation time. Sub-controllers propagate these events to all of their simulation agents. This architecture allows for distributed settings where researchers can share computation resources by providing PlaSMA sub-controller services that connect to the top-controller.

The world model component represents the state of the world consisting of its environment and the physical objects acting therein. It is initialized based on a formal ontology description [1]. The simulation agents represent physical objects, abstract services, or legal entities (e.g., organizations) in the simulation model. Simulation agents are able to communicate with each other by message passing in FIPA Agent Communication Language ACL [5] and may act in the environment. The PlaSMA user can create or adapt scenario-specific agents by implementing an extension based on a Java template agent implementation.

The analysis component consists of an object-relational database storing scenario performance metrics (e.g., cargo cycle times or vehicle utilization), a program library for pseudo random number generation and statistical analysis, as well as an interface to log and query these metrics and additional log messages. The simulation agents use this interface to add or update their performance metrics during runtime. The metrics are specified at design time. Each metrics specification consists of a label, the data type, the unit of measurement, optional arithmetic functions, and temporal precision of updates. It is implemented in Java based on template performance metrics for

[1] Available from http://plasma.informatik.uni-bremen.de

each basic data type. It is also possible to specify complex metrics as composites of basic or other complex metrics.

One user or multiple users at the same time can remotely observe the metrics online within the PlaSMA client. When agents provide position data and the scenario contains additional visualization information (i.e. a map) one can also track the simulation environment within the viewer (see Fig. 1). The client/viewer remotely connects to the server and to the relational database of the analysis component. The client allows for selection of scenarios that were deployed previously to the PlaSMA server. A scenario may be paused, resumed, stopped, and visualized while running.

B. *Applications and Experiences*

Although actively used, the PlaSMA system is still considered prototype research software. PlaSMA is applied for comparison and evaluation of algorithms for logistics planning and special sub-processes therein, e.g., coordination mechanisms of logistics objects, information distribution, environment adaptation and prediction [7] as well as routing and cargo clustering algorithms [16]. Furthermore, PlaSMA is part of the "Intelligent Container" platform [10] integrating simulation with real-world hardware in perishable food transport scenarios. In the context of adaptive route planning, PlaSMA was integrated with the AQ21 machine learning system [18] for predictions of expected traffic and speed on potential routes [7].

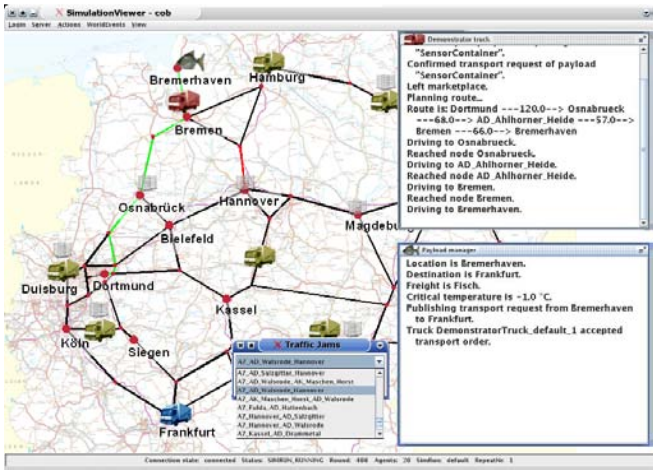
Complexity of simulation surveys ranges from very few agents (4) to a great many agents (approx. 20,000). Few agents will not benefit form a large distributed setting because one single agent cannot be distributed. Large-scale scenarios will work best on multi-CPU machines or clusters. The run-time performance of truly distributed computing resources will heavily depend on amount of agent interaction and network connection. PlaSMA has been executed on two distributed servers with 8 CPUs and 64 GM working memory each. However, even with several thousand agents one machine might not be fully utilized because agents are waiting for each other when interacting.

IV. COLLABORATIVE EXPERIMENTATION WITH MABS

The agent-based experimentation process is structured in the following steps: (a) Experiment design and performance metric specification, (b) agent implementation, (c) simulation infrastructure setup, (d) agent deployment, (e) simulation execution, and finally (f) result analysis. These steps are further described in the following sections and discussed with respect to collaborative experimentation.

A. *Experiment Design and Performance Metric Specification*

The first step addresses the specification of the general design of the simulation experiment. The design determines which types of agent are the simulation subject, how these agents should interact, and in which environment they should be situated. For instance, an experiment in transportation may have agents for trucks, trains, cargo, warehouses, and humans or corporations operating them. It is important to distinguish physical objects, technical decision systems such as software agents, human decision makers, and organizations.



```
  "SensorContainer".
Confirmed transport request of payload
  "SensorContainer".
Left marketplace.
Planning route...
Route is: Dortmund ---120.0--> Osnabrueck
  ---68.0--> AD_Ahlhorner_Heide ---57.0-->
  Bremen ---66.0--> Bremerhaven
Driving to Osnabrueck.
Reached node Osnabrueck.
Driving to AD_Ahlhorner_Heide.
Reached node AD_Ahlhorner_Heide.
Driving to Bremen.
Reached node Bremen.
Driving to Bremerhaven.
```

```
Location is Bremerhaven.
Destination is Frankfurt.
Freight is Fisch.
Critical temperature is -1.0 °C.
Publishing transport request from Bremerhaven
  to Frankfurt.
Truck DemonstratorTruck_default_1 accepted
  transport order.
```

Figure 1. PlaSMA Control and Visualization Client

This process of object and subject identification is supported and formalized by formal ontologies provided by PlaSMA using the W3C standard OWL [1]. The PlaSMA ontologies provide a general domain conceptualization that can be extended for each scenario. In particular, a scenario specification provides instantiations of concepts, i.e., the identified objects and subjects, including their attributes and relationships.

A multiagent-based simulation is constituted by agents as logical simulation processes. Thus, following the identification process, the objects and subjects have to be mapped to actual simulation agents that should be implemented as the simulation model. But not each object or subject needs to be implemented as exactly one agent in bijective mapping. It is up to the designer of the experiment whether, e.g., truck and driver are separate agents or combined in one agent.

Fortunately, not all users participating in collaborative experimentation have to agree on these design decisions. There only has to be an agreement on the possible interactions of subjects (objects only interact via the simulated environment). The general structure and language of these interactions is already defined in the FIPA Agent Communication Language (ACL) standard [5]. This is the great advantage of agent interoperability. Nevertheless, there has to be a common understanding of the content of the ACL messages exchanged between agents, i.e., the semantics. Again, ontologies provide an adequate means here.

Performance metrics specification is about the data to be collected during simulation and to be evaluated after simulation. It is also important to define the data type and physical unit for all metrics. The specification also includes the desired precision in numerical, temporal, and possibly spatial terms. As described in Sect. 3, metrics are finally implemented as Java classes. Thus, the implementation of performance metrics is a common task in collaborative design and considered a part of the design process.

B. *Agent Implementation*

All agents are implemented based on a template simulation agent that provides basic features for simulation time synchronization. Besides these small simulation-specific changes, the template agent is equal to the basic Agent class provided by the underlying JADE software. Nevertheless, each agent participating in the simulation has to be mapped to some subject and/or object specified during experiment design in order to link simulation agents to the world model. This link is specified in a scenario XML configuration but also extensible at run-time.

Collaboration in implementation phase depends on the intended scenario. A competitive approach will imply that different groups will implement agents with the same tasks. These agents will then run in parallel or in sequential competition. Only in the first case the agent of different groups might actually influence each other or cooperate. Another collaborative setting will split the scenario to different subjects and objects that are implemented by a certain group. The simulation controllers will decide on which computer in the distributed infrastructure some agent is actually executed.

C. Simulation Infrastructure Setup

PlaSMA has a distributed hierarchic infrastructure that allows sharing computing resources, e.g., in a virtual organization. The PlaSMA top-controller acts as a master server and sub-controllers as subordinate servers that host the simulation agents. For infrastructure setup, the master server program is started first. Afterwards subordinate servers on potentially remote machines are started and register at the main server. For a well-balanced distribution of agents, each subordinate server states its computing performance at registration. The performance indicator is given by the maximum number of agents that can run on each machine respectively. This value corresponds to a reference performance test agent; the actual agents will have individual computation needs. Based on this information, the master server decides during simulation execution how to distribute the simulation agents.

There are ongoing efforts in the MABS community to apply Grid computing for sharing computing resources in simulation. Corresponding approaches which also rely on Java software for platform independence have been proposed by Timm and Pawlaszczyk [17] as well as Mengistu et al. [14].

In PlaSMA, it is not necessary to re-setup the simulation infrastructure for each scenario. The main and remote server services are started once, but they may restart the underlying JADE agent platform when loading a new scenario. One difficulty here is dynamic code loading because new simulations might have new agents. In general, the Java runtime environment (JRE) needs to know every resource location and name already on startup. But in distributed simulation and particularly in collaborative experimentation the agent code is distributed on multiple systems and possibly not yet present when the PlaSMA server is started.

Thus, PlaSMA separates distributed server programs from the actual simulation controllers and agent platforms. When a new scenario is loaded, the server services will bundle and distribute the necessary Java code to all PlaSMA services in the network. The services will then start new PlaSMA sub-controllers in a separate JRE.

This distribution and start procedure for each simulation experiment does not require any user interaction. The only thing that has to be determined in advance is the location (i.e., IP address and network port) of the master server. However, before a simulation scenario can be loaded the participating users will need to deploy the agent code to a resource location dedicated for scenario code.

D. Agent Deployment

An agent developer for PlaSMA can compile the Java source code as he or she prefers and is used to. Additionally, a deployment process has to be executed when the scenario is ready. All compiled resources, associated libraries, and configuration files must be bundled in a scenario package with a certain directory structure. This package is then copied (or deployed) to a designated location for PlaSMA scenarios similar to a web application deployed to a web server.

PlaSMA provides the necessary program and template configuration for this procedure. The deployment configuration specifies the file system destination of a simulation scenario and the local location of resources to be deployed. The deployment location has to be accessible for the PlaSMA master server that will load and distribute the resources.

Currently, the scenario deployment assumes that all agents and other scenario resources are located at one location already before deployment. This restriction complicates the collaborative and distributed development of scenarios. Thus, we are working on an adapted deployment process that facilitates the cooperation by distributed and partial deployments. That is, each collaborator can deploy his code to a location that is accessible to a (local) subordinate PlaSMA service. The system then retrieves and composes the distributed resources when a scenario is loaded.

E. Simulation Execution

After infrastructure setup and scenario deployment the simulation is ready for execution. All experiment collaborators can connect to the PlaSMA master server via the PlaSMA client. Any user can load and start the scenario and all users are able to monitor the experiment process including object motion and performance metric values. If simulation execution is too fast for online monitoring one is able to adjust the maximum speed of the simulation by setting the maximum ratio between simulation time and wall-clock time. A change of maximum speed will affect the actual simulation server and thus all viewer clients.

The experiment stops if some user explicitly halts it or a configured simulation time length is reached. Other stop criteria under consideration for future releases include goal values for statistic significance (e.g., maximum standard error or confidence interval length).

F. Result Analysis

The analysis phase is not yet fully integrated in the collaborative experimentation process. All data collected during simulation is stored in a single database which is saved after each simulation run. The simulation data is saved in structured text files for each database table. All files are located on the server hosting the database system. PlaSMA provides a limited analysis toolkit to evaluate the experiments. It is also easily possible to import the database data to spreadsheet programs or numerical analysis software.

The preliminary experiment analysis tool of PlaSMA is based on a third-party Java translation of the *R* statistical computing environment. An XML configuration file specifies the performance metrics to be evaluated, what and how data sources should be merged (simulation runs, agents), how performance metrics should be aggregated, and the significance levels to be used. The tool provides a text-based evaluation report including confidence intervals if applicable.

V. CONCLUSION

In this paper we presented the PlaSMA multiagent-based simulation system. PlaSMA enables large-scale distributed simulations as a middleware for the JADE agent platform and relies on Java to maximize platform independence. The system has been applied in several applications including experiments to evaluate the impact of environment condition prediction in transportation based on inductive machine learning.

The agent-based simulation approach particularly suits virtual organizations in research because it enables shared computing resources as well as collaborative experimentation for, e.g., comparative and competitive evaluation of algorithms. PlaSMA can also provide the basis for reference evaluation scenarios without shared resources. The experimentation process is described with particular focus on (remote) collaborative experimentation of researchers sharing simulation scenarios and computing resources.

Some shortcomings in the experimentation process have been detected that complicate collaborative experimentation. Thus, future work will include enhancements in collection and distribution of scenario resources as well as better access to simulation results for instant analysis by all collaborators. PlaSMA was used with the machine learning program AQ21. Additionally, we will need to integrate further tools for computational intelligence and machine learning to make it a valuable platform for virtual organizations in this research area.

REFERENCES

[1] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, L. A. Stein, and F. W. Olin, OWL Web Ontology Language Reference, W3C Recommendation. World Wide Web Consortium, Feb. 2004.

[2] F. Bellifemine, A. Poggi, and G. Rimassa, “Developing multi-agent systems with a FIPA-compliant agent framework,” Software-Practice and Experience, vol. 31 (2), pp. 103–128, Feb. 2001.

[3] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper, “Bridging the gap between simulation and reality in urban search and rescue,” in RoboCup 2006: Robot Soccer World Cup X, in series LNAI, vol. 4434. Berlin: Springer-Verlag, 2008, pp. 1–12.

[4] P. Davidsson, J. Holmgren, H. Kyhlbäck, D. Mengistu, and M. Persson, “Applications of agent based simulation,” in Multi-Agent-Based Simulation VII, International Workshop (MABS 2006), Hakodate, Japan, May 8, 2006, in series LNAI, vol. 4442.Berlin: Springer, 2007, pp. 15–27.

[5] FIPA, Ed., FIPA ACL Message Structure Specification. Document Nr.: SC00061G. Internet: http://www.fipa.org/specs/fipa00061/. Geneva: Foundation for Intelligent Physical Agents, 2002.

[6] R. Fujimoto, Parallel and Distributed Simulation Systems. New York, NY: John Wiley & Sons, 2000.

[7] J. D. Gehrke and J. Wojtusiak, “Traffic prediction for agent route planning,” in International Conference on Computational Science 2008 (vol. 3), Kraków, Poland, June 23-25, 2008, in series LNCS, vol. 5103. Berlin: Springer-Verlag, 2008, pp. 692–701.

[8] J. D. Gehrke, A. Schuldt, and S. Werner, “Quality criteria for multi-agent-based simulations with conservative synchronisation,” in 13th ASIM Dedicated Conference on Simulation in Production and Logistics, Berlin, Germany, Oct. 1–2, 2008, in press.

[9] IEEE; Ed., Standard for Modeling and Simulation High Level Architecture. IEEE Standard 1516-2000. IEEE Computer Society, 2000.

[10] R. Jedermann, C. Behrens, R.Laur, and W. Lang, “Intelligent containers and sensor networks, approaches to apply autonomous cooperation on systems with limited resources,” in Understanding Autonomous Cooperation & Control in Logistics, M. Hülsmann and K. Windt, Eds. Berlin: Springer-Verlag, 2007, pp. 365–392.

[11] D. R. Jefferson. Virtual Time II, “Storage management in conservative and optimistic systems,” in Proceedings of the Ninth Annual ACM Symposium on Principles of Distributed Computing (PODC 1990), August 22–24, 1990, Quebec City, Quebec, Canada. New York, NY: ACM Press, 1990, pp. 75–89.

[12] M. Lees, B. Logan, R. Minson, T. Oguara, and G. Theodoropoulos, “Distributed simulation of MAS,” in Multi-Agent and Multi-Agent-Based Simulation, Joint Workshop (MABS 2004), New York, NY, USA, July 19, 2004, in series LNAI, vol. 3415. Berlin: Springer-Verlag, 2005, pp. 25–36

[13] N. M. Mayer, J. Boedecker, R. da Silva Guerra1, O. Obst, and M. Asada, “3D2Real: Simulation league finals in real robots,” in RoboCup 2006: Robot Soccer World Cup X, in series LNAI, vol. 4434. Berlin: Springer-Verlag, 2008, pp. 25–34.

[14] D. Mengistu, P. Davidsson, and L. Lundberg, “Middleware support for performance improvement of MABS applications in the Grid environment,” in Multi-Agent-Based Simulation VIII, International Workshop (MABS 2007), Honolulu, HI, May 15, 2007, in series LNAI, vol. 5003. Berlin: Springer-Verlag, 2008, pp. 20–35.

[15] H. V. D. Parunak, R. Savit, and R. L. Riolo, “Agent-based modeling vs. equation-based modeling: A case study and users' guide,” in Multi-Agent Systems and Agent-Based Simulation, First International Workshop (MABS ’98), Paris, France, July 4–6, 1998, in series LNAI, vol. 1534. Berlin: Springer-Verlag, 1998, pp. 10–25.

[16] A. Schuldt and S. Werner, “Distributed Clustering of Autonomous Shipping Containers by Concept, Location, and Time,” in 5th German Conference on Multiagent System Technologies, Leipzig, Germany, Sept. 24–26, 2007, in series LNAI, vol. 4687. Berlin: Springer-Verlag, 2007, pp. 121–132.

[17] I. J. Timm and D. Pawlaszczyk, “Large scale multiagent simulation on the Grid,” in Proceedings of the Fifth IEEE International Symposium on Cluster Computing and the Grid (CCGRID’05), Cardiff, UK, May 9–12, 2005. Washington, DC: IEEE Computer Society, 2005, pp. 334–341.

[18] J. Wojtusiak, R. S. Michalski, K. A. Kaufman, and J. Pietrzykowski, "The AQ21 Natural Induction Program for Pattern Discovery: Initial Version and its Novel Features," in Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence, Washington, DC, November 13–15, 2006. Los Alamitos, CA: IEEE Computer Society, 2006, pp. 523–526.

[19] M. Wooldridge and N. R. Jennings, “Intelligent agents: Theory and practice,” The Knowledge Engineering Review, vol. 10 (2), pp. 115–152, 1995.

Demonstration and Application of Rule Discovery Methods Using iAQ

Jaroslaw Pietrzykowski

Abstract—**The paper presents iAQ, an interactive, multimedia-capable system, that exhibits and allows the application of machine learning methods representing the Natural Induction (NI) paradigm. The system is presented in relation to the Virtual Organizations (VO) area. The program's unique set of features is examined and demonstrated with selected examples. It can be downloaded from the Machine Learning and Inference (MLI) laboratory website: http://www.mli.gmu.edu/software.html**

***Index Terms*—Artificial intelligence, Computational intelligence, Computer science education, Learning systems, Machine leaning, Multimedia systems, Virtual Organizations.**

I. Introduction

THIS paper presents iAQ, a program that demonstrates important concepts in the machine learning area, as well as allowing users to apply some of the methods from that field to user's own problems. The aim of developing the program was to create a unique educational experience for a wide spectrum of audiences that goes beyond traditional research and academia community. To this end, it provides an attractive, entertaining environment, which should appeal to people regardless of their background, with various levels of familiarity with presented topics, and that may raise and increase their interest in science and technology.

iAQ can also be viewed not only as a way of disseminating knowledge about the "world of machine learning", but also as a component of virtual organizations being developed in that area. Virtual Organizations (VO) form a very important recent trend, and are identified as a key enabler of system-level science that is concerned with complex, large, and multidisciplinary phenomena [1]. Although it seems that the major focus of the development the VO is on the large-scale, high-performance infrastructure, smaller scale software like iAQ can also play a significant role. Since VOs support multi-disciplinary collaboration, iAQ can be used as an important tool allowing the participants, coming from various fields, to familiarize themselves with the underlying technology and test it with their own problems, all this in a user-friendly, human-oriented manner. Depending on the level of interest, it offers ways for the users to learn more about the featured methodology via various depth descriptions, links to more elaborate content on Internet or to the accompanying materials.

iAQ can also contribute to the democratization of science – one of the major potential benefits of VOs [1], by involving students at all levels of the education system. Program's ability to offer "learning by doing" may also attract new young people to pursue careers in research. And it seems that its design corresponds well to the modern theories of pedagogy, that concentrate more on how to help students internalize new information in personally meaningful and adaptable ways [2]. iAQ is freely available for download from the Machine Learning and Inference Laboratory website, therefore it can be accessed by self-learners. The importance of this group of users is recognized for example by Open Educational Resources Initiative which supports the use of information technology to help equalize access to knowledge and educational opportunities across the world [2].

iAQ is planned to become a part of the CIML portal [3], as one of the educational packages. Access to such tools for a large number of communities of users is one of the most important benefits of the project.

The iAQ system exemplifies the Natural Induction (NI) paradigm to machine learning [4], that is, the ability of a computer program to learn knowledge which is not only accurate, but also have forms natural to people, and is by that easy to understand and interpret. It is intended to encourage users to go beyond its introductory level, and reach for more sophisticated NI tools, such as VINLEN [5], which integrates a number of learning methods developed over many years in the MLI laboratory.

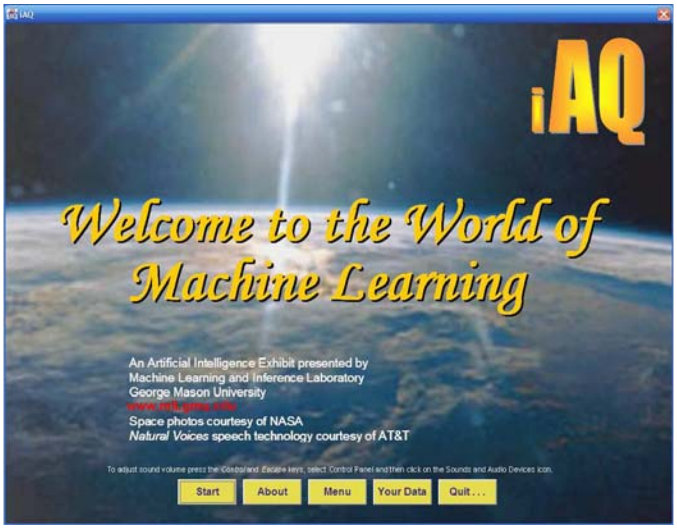
The author believes that the set of features, including such multimedia functions as speech, music, an easy to navigate GUI, the simple, yet appealing problem domain of recognizing robots, an entertaining storyline, the ability to perform a user's own experiments, ready-to-use examples, tutorial pages providing insight into the details of the employed methods, links to the website with more elaborate materials, a book with a comprehensive summary of the MLI Laboratory research, is rather unique among this type of programs. iAQ can serve as a good example of an interactive and multimedia-rich tutorial and learning environment to be offered by virtual organizations.

This work was supported in part by the National Science Foundation Grants No. IIS 9906858, IIS 0097476 and CBET 0742487, and in part by the UMBC/LUCITE #32 grant. The findings and opinions expressed here are those of the authors, and do not necessarily reflect those of the sponsoring organizations.

Jaroslaw Pietrzykowski is with the George Mason University, Fairfax, VA 22030 USA, e-mail: jarek@mli.gmu.edu.

II. iAQ Features

iAQ has been designed to make a user's experience with the program as enjoyable as possible, by including pictures of the outer space and Earth (see Fig.1), visually appealing depiction of the main characters of the presented tale – the robots, easy to navigate graphical user interface with large buttons that have clearly designated functions, represented by easy to understand symbols and text. To enhance the experience all the pages that the program consist of are accompanied by a very clear speech, that helps to follow the presented content by reading the text shown on the screen using female or male voice. The male voice tells the parts of iAQ, an expert program in creating rules for recognizing objects that is the central character of the story, and the female voice tells the narrative parts - in the introduction, in the lead-out, instructions for various parts of the program, and descriptions of the methodology. The up-beat music in the beginning and more relaxed in the end make the whole adventure more complete. The intention of this design was to please audiences at various levels of familiarity with the presented material and regardless of their age.



An Artificial Intelligence Exhibit presented by
Machine Learning and Inference Laboratory
George Mason University
www.mli.gmu.edu
Space photos courtesy of NASA
Natural Voices speech technology courtesy of AT&T

To adjust sound volume press the Control and Escape keys, select Control Panel and then click on the Sounds and Audio Devices icon.

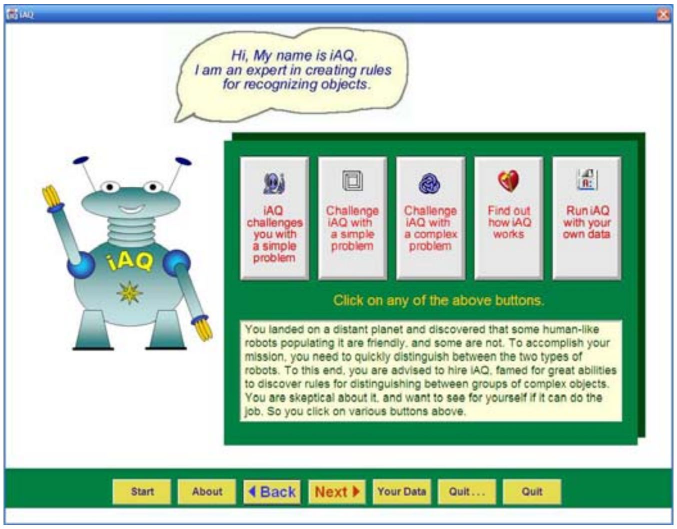
Fig. 1. The welcoming screen of the program.

There has been much emphasis put on the interactive aspect of the program. The user can either follow the pre-defined flow between the pages, or access freely selected parts, once (s)he has become familiar with, for example, introduction or the simple guessing game in the beginning. The "Next" and "Back" buttons allow the user to decide on which part (s)he would like to concentrate the most at the moment, to refresh the context of the task at hand, or simply just to skip to the next part. The buttons are always available at the bottom of the screen, along with other buttons allowing jumping to selected sections: introduction, "goodbye" screens, short description of the used methods and their history, and the application of the methods to user's own problem.

The welcoming screen takes the user to the series of pages, that present a short introduction of how (s)he has become involved in the story, including the references to the machine learning field as a valuable source of help.

The screen with the main menu presents further details of the whole story and options for accessing various modules of the program (see Fig.2).

The first button in the menu features simpler and more complicated versions of a guessing game, that involves the user into solving a problem of discovering rules for distinguishing between "friendly" and "unfriendly" robots.



iAQ
Hi, My name is iAQ.
I am an expert in creating rules
for recognizing objects.
iAQ
iAQ challenges you with a simple problem
Challenge iAQ with a simple problem
Challenge iAQ with a complex problem
Find out how iAQ works
Run iAQ with your own data
Click on any of the above buttons.
Start
About
◀ Back
Next ▶
Your Data
Quit . . .
Quit

You landed on a distant planet and discovered that some human-like robots populating it are friendly, and some are not. To accomplish your mission, you need to quickly distinguish between the two types of robots. To this end, you are advised to hire iAQ, famed for great abilities to discover rules for distinguishing between groups of complex objects. You are skeptical about it, and want to see for yourself if it can do the job. So you click on various buttons above.

Fig. 2. The description of the premise of the lead story and the modules of the program.

In the next step, which is rather more exciting and engaging, the roles are reversed, and now it is the program's turn to guess the user's secret rule. The rule tells which robots should be invited to the user's club, and which should not. There are 16 different robots to choose from, therefore there are many possible ways to devise a rule for grouping them. When the program is demonstrated to a wider audience, this presents a nice opportunity to include more than one participant in the game. The third button demonstrates more complicated version of this game, where the robots can be split into four different groups.

The fourth button called "About" shows a page with a description of how the underlying algorithm works. In order to broaden the educational benefit from the user's experience with the program, this button displays a page with the historical context of the development of the employed methods, including references to the iAQ predecessors like EMERALD [6] that featured wider range of algorithms. For a curious person, who would like to learn more about the research in the Machine Learning and Inference laboratory, on some of the pages there are links that lead to laboratory's web site and also display a brochure with large parts of the website's featured content.

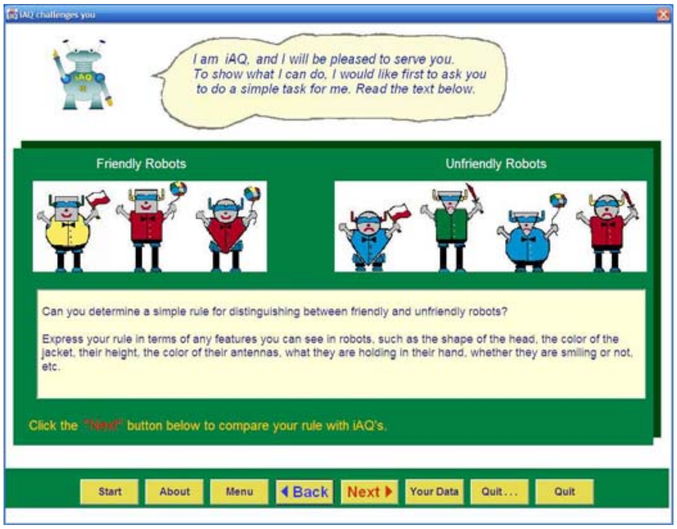
The last item on the menu is designed for more advanced users, who already feel sufficiently familiar with the presented methodology and are ready to apply it to their own problems. This part comes with a built-in example of the input representation for the problem of recognizing various groups of robots. The example can be easily modified to fit to specific tasks that users would like to work on. Modifications can be done by hand, or using the copy and paste mechanism, or a whole new input problem representation can be loaded from a local file. This feature offers a hands-on experience for the user, which is crucial for good understanding of the presented methods. Altogether the modules provide a personal educational adventure.

III. Illustration of using iAQ

In this section the most important parts of the program are presented, that engage the user into the interactive play with the machine learning methods and allow their application to the user's own problems.

A. Simple guessing game

Fig. 3 shows the robots divided into "friendly" and "unfriendly" groups and the description of the task that the user is facing, namely to discover the rule the best distinguishes between these two groups. Although, the rule is quite simple: "friendly robots are smiling and unfriendly are not", it may be not very easy to find for some people, since some of the robots' features may seem more prominent. The "Next" button shows the screen with the correct answer, and subsequently the user is asked to solve a more complicated case, where some of the "unfriendly" robots are smiling (see Fig. 4).


iAQ challenges you
I am iAQ, and I will be pleased to serve you. To show what I can do, I would like first to ask you to do a simple task for me. Read the text below.
Friendly Robots
Unfriendly Robots
Start
About
Menu
Back
Next
Your Data
Quit...
Quit

Can you determine a simple rule for distinguishing between friendly and unfriendly robots?

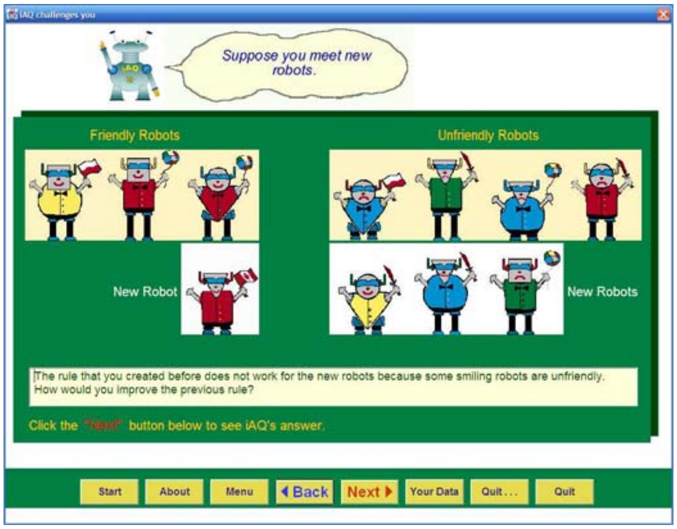
Express your rule in terms of any features you can see in robots, such as the shape of the head, the color of the jacket, their height, the color of their antennas, what they are holding in their hand, whether they are smiling or not, etc.

Click the "Next" button below to compare your rule with iAQ's.

Fig. 3. iAQ challenges the user to guess the rule distinguishing two groups of robots.

B. Challenging iAQ with an easy problem

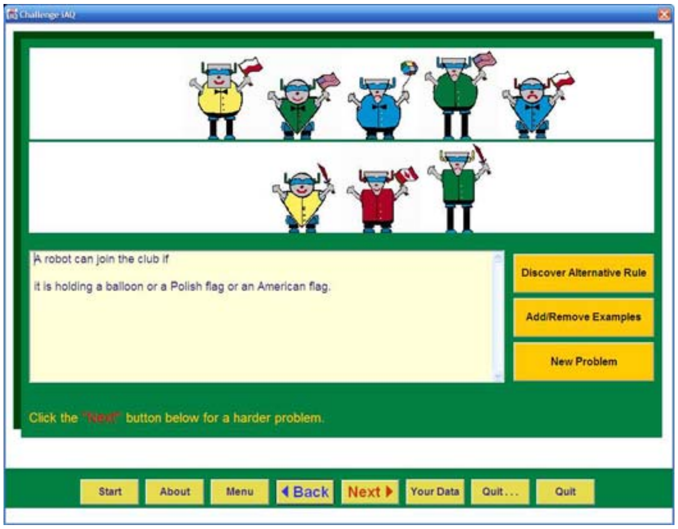
To show how guessing users' hidden rules works, a selection of eight robots is shown, where five of them were invited to the user's club, and the remaining three were not (see Fig. 5). No matter how complicated the hidden rule is, iAQ is able to discover it, trying to find the simplest ones first. The program's subsequent guesses are listed below. This list shows also another feature of the program, namely the ability to generalize knowledge using hierarchies. In the case of the robots, the attributes representing what a robot is holding and the shape of its head are structured in such a way that, for example, different types of flag form one super-type "flag", and various shapes of head are categorized as "polygonal". This allows for human-like inference, which is exemplified with Rule #8 on the list, where four of the invited robots are collectively described as holding a flag (without specifying its kind), apart from other features.



The rule that you created before does not work for the new robots because some smiling robots are unfriendly. How would you improve the previous rule?

Click the "Next" button below to see iAQ's answer.

Fig. 4 The user is asked to guess the rule in a more complicated case.



A robot can join the club if

it is holding a balloon or a Polish flag or an American flag.

Click the "Next" button below for a harder problem.

Fig. 5. An easy problem for iAQ to solve, where there are 2 groups of robots: invited and not invited to the user's club. First found solution is presented.

Rule #2: it is wearing a tie, or if its body is round.

Rule #3: it is not holding a sword and its jacket is not red.

Rule #4: its jacket is blue or green and its body is not square, or if its head is square.

Rule #5: its body is not square and it is not holding a sword.

Rule #6: its antennas are blue or green and it is not holding a sword, or if its jacket is blue.

Rule #7: it is holding a flag and its head is not triangular, or if its antennas are blue.

Rule #8: it is holding a flag and the color of the body and the color of the antennas are different, or if its jacket is blue or green and it is short.

Rule #9: it is holding a flag and the color of the body and the color of the antennas are different, or if its jacket is blue or green and the color of the body and the color of the antennas are the same.

Rule #10: it is holding a flag and the color of the body and the color of the antennas are different, or if the color of the body and the color of the antennas are the same and its body is not square.

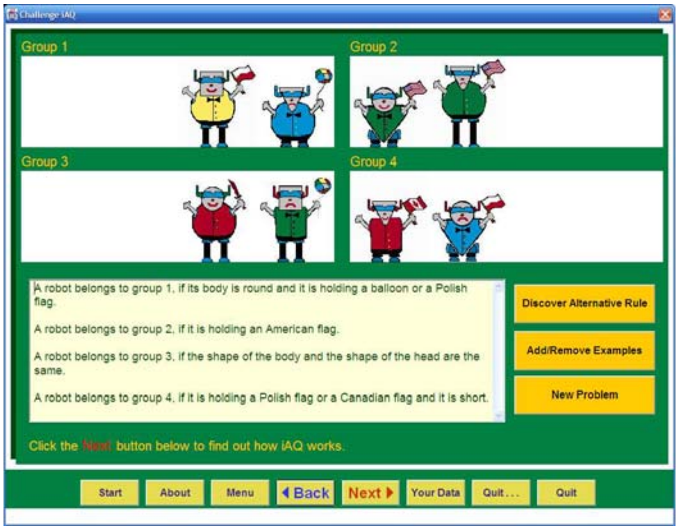
Rule #11: it is holding a flag and the color of the body and the color of the antennas are different, or if its jacket is blue or green and its antennas are blue or green.

Rule #12: its jacket is blue or green and its antennas are not yellow, or if it is holding a flag and the color of the body and the color of the antennas are different.

Rule #13: it is holding a flag and the color of the body and the color of the antennas are different, or if its jacket is blue or green and it is short.

C. *Challenging iAQ with a complex problem*

To illustrate how iAQ can help with distinguishing between more than 2 classes, a sample example has been chosen, where two robots each were selected into four (the maximum number) separate groups, as shown on Fig 6.



A robot belongs to group 1, if its body is round and it is holding a balloon or a Polish flag.

A robot belongs to group 2, if it is holding an American flag.

A robot belongs to group 3, if the shape of the body and the shape of the head are the same.

A robot belongs to group 4, if it is holding a Polish flag or a Canadian flag and it is short.

Click the Next button below to find out how iAQ works.

Fig. 6. A complex problem for iAQ to solve, where there are four different groups of robots to be described. First found solution is presented.

It is easy to see what possible hidden rules the user could have in mind when forming the groups 2 and 4, and somewhat more difficult to see them for the groups 1 and 3. Nevertheless, even more complicated characterizations do not pose a real challenge for the learning module as it may be seen on the list of alternative rules discovered, that is presented below. Discovering alternative descriptions of classes is a powerful feature of the program and it may be inspiring for the user to seek for larger number of possible explanations in her/his investigations.

Rule #2
1, if its jacket is yellow or blue and its body is round.
2, if its jacket is green and it is not of medium height.
3, if it is of medium height and it is not holding a flag.
4, if its jacket is red or blue and it is holding a flag.

Rule #3
1, if its jacket is yellow or blue and its head is polygonal.
2, if its jacket is green and it is holding a flag.
3, if its jacket is red or green and it is not holding a flag.
4, if it is holding a Polish flag or a Canadian flag and its body is polygonal.

Rule #4
1, if its body is round and it is wearing a tie.
2, if its jacket is green and its body is not square.
3, if it is of medium height and its jacket is red or green.
4, if its body is polygonal and its jacket is red or blue.

Rule #5
1, if it is holding a balloon or a Polish flag and its antennas are blue or green.
2, if its jacket is green and its head is not square.
3, if it is not holding a flag and its head is not triangular.
4, if it is holding a Polish flag or a Canadian flag and its head is not square.

Rule #6
1, if its jacket is yellow or blue and its antennas are blue or green.
2, if its jacket is green and the shape of the body and the shape of the head are different.
3, if the color of the body and the color of the antennas are different and it is not holding a flag.
4, if its antennas are red and it is short.

Rule #7
1, if its body is round and it is not holding a sword and it is not tall.
2, if its jacket is green and the shape of the body and the shape of the head are different.
3, if its antennas are red or green and it is not holding a flag.
4, if its antennas are red and it is holding a flag.

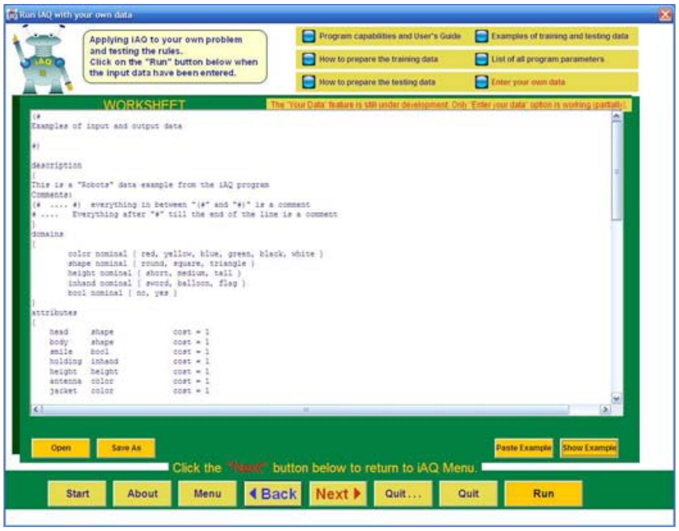
Rule #8:
1, if its body is round and it is not holding a sword and it is not tall.
2, if its jacket is green and the shape of the body and the shape of the head are different.
3, if its antennas are red or green and it is not holding a flag.
4, if its antennas are red and its jacket is red or blue.

Rule #9:
1, if its body is round and it is not holding a sword and it is not tall.
2, if its jacket is green and the shape of the body and the shape of the head are different.
3, if its antennas are red or green and it is not holding a flag.
4, if its antennas are red and the shape of the body and the shape of the head are different.

Rule #10:
1, if its body is round and it is not holding a sword and it is not tall.
2, if its jacket is green and the shape of the body and the shape of the head are different
3, if its antennas are red or green and it is not holding a flag
4, if its antennas are red and its head is not square

D. *Applying iAQ to user's own data*

In order to make it easier for the user to understand better how the underlying method works, iAQ contains a ready to use example which is based on the robot recognition problem used in the other parts of the program. The GUI enables the user to paste the prepared example, run it, analyze the results, then possibly make some modifications and perform a few more iterations. A text editor allows the user to copy some of the data, or attribute descriptions from other programs. The results


Run iAQ with your own data
iAQ
Applying iAQ to your own problem and testing the rules.
Click on the "Run" button below when the input data have been entered.
Program capabilities and User's Guide
Examples of training and testing data
How to prepare the training data
List of all program parameters
How to prepare the testing data
Enter your own data
WORKSHEET
The 'Your Data' feature is still under development. Only 'Enter your data' option is working (partially).
Open
Save As
Paste Example
Show Example
Click the "Next" button below to return to iAQ Menu.
Start
About
Menu
Back
Next
Quit . . .
Quit
Run

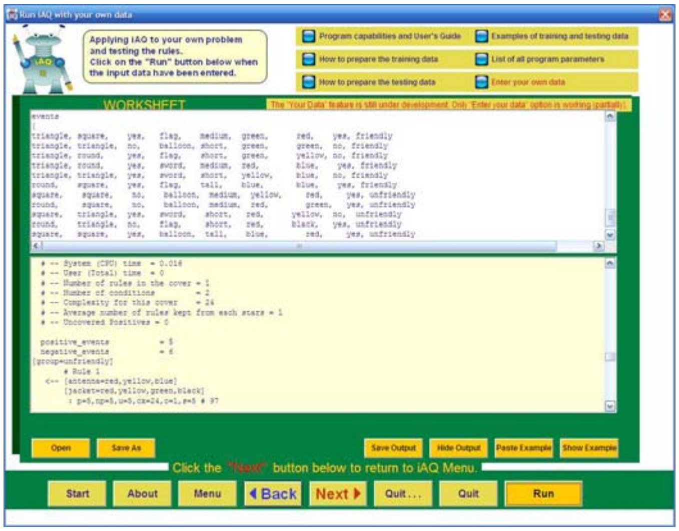
```
(#
Examples of input and output data

#)

description
{
This is a "Robots" data example from the iAQ program
Comments:
(#  .... #)  everything in between "(#" and "#)" is a comment
# ....   Everything after "#" till the end of the line is a comment
}
domains
{
        color nominal { red, yellow, blue, green, black, white }
        shape nominal { round, square, triangle }
        height nominal { short, medium, tall }
        inhand nominal { sword, balloon, flag }
        bool nominal { no, yes }
}
attributes
{
    head     shape            cost = 1
    body     shape            cost = 1
    smile    bool             cost = 1
    holding  inhand           cost = 1
    height   height           cost = 1
    antenna  color            cost = 1
    jacket   color            cost = 1
```

Fig. 7. Running iAQ with the user's own problem. First part of the input file used in the ready to use robot recognition problem.

can be copied to the buffer memory too, and also saved to an external file. If the user has the input file ready, it can be loaded into the editor as well. Fig. 7 shows the text editor window with the input data for the robot example, and Fig. 8 displays the results of the run.


Run iAQ with your own data
iAQ
Applying iAQ to your own problem and testing the rules.
Click on the "Run" button below when the input data have been entered.
Program capabilities and User's Guide
Examples of training and testing data
How to prepare the training data
List of all program parameters
How to prepare the testing data
Enter your own data
WORKSHEET
The 'Your Data' feature is still under development. Only 'Enter your data' option is working (partially).
Open
Save As
Save Output
Hide Output
Paste Example
Show Example
Click the "Next" button below to return to iAQ Menu.
Start
About
Menu
Back
Next
Quit . . .
Quit
Run

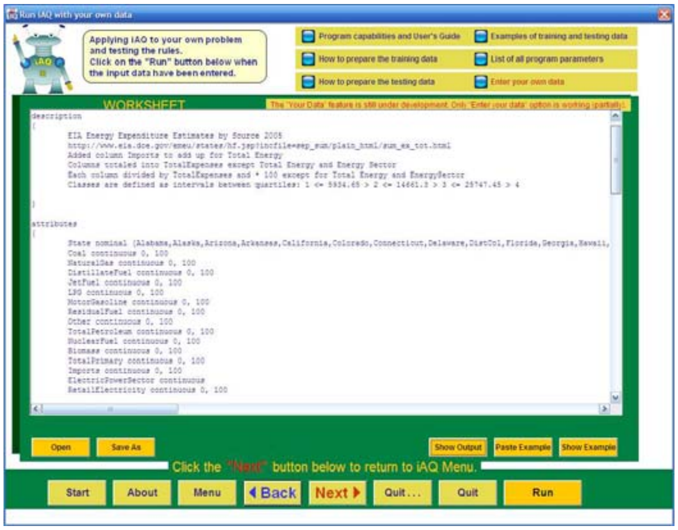
```
events
{
triangle, square,    yes,  flag,    medium,  green,     red,    yes, friendly
triangle, triangle,  no,   balloon, short,   green,     green,  no, friendly
triangle, round,     yes,  flag,    short,   green,     yellow, no, friendly
triangle, round,     yes,  sword,   medium,  red,       blue,    yes, friendly
triangle, triangle,  yes,  sword,   short,   yellow,    blue,   no, friendly
round,    square,    yes,  flag,    tall,    blue,      blue,    yes, friendly
square,    square,    no,   balloon,  medium,  yellow,    red,     yes, unfriendly
round,     square,    no,   balloon,  medium,  red,       green,   yes, unfriendly
square,   triangle,  yes,  sword,    short,   red,      yellow,  no,  unfriendly
round,    triangle,  no,   flag,     short,   red,      black,   yes, unfriendly
square,   square,    yes,  balloon,  tall,    blue,       red,     yes, unfriendly
```

```
 # -- System (CPU) time  = 0.016
 # -- User (Total) time  = 0
 # -- Number of rules in the cover = 1
 # -- Number of conditions         = 2
 # -- Complexity for this cover    = 24
 # -- Average number of rules kept from each stars = 1
 # -- Uncovered Positives = 0

 positive_events          = 5
 negative_events          = 6
[group=unfriendly]
      # Rule 1
  <-- [antenna=red,yellow,blue]
      [jacket=red,yellow,green,black]
       : p=5,np=5,u=5,cx=24,c=1,s=5 # 97
```

Fig. 8. Running iAQ with the user's own problem. The bottom part of the screen shows a part of the output for the robot problem.

As an example to illustrate how iAQ can be employed to help the user analyze the problem at hand, we use a data set from the website of Energy Information Administration (a part of Department of Energy) that concerns state energy expenditure estimates in 2005[1]. The data has been prepared for learning characteristics of the four classes of states in terms of the structure of their expenditures with respect to various energy sources. Classes have been defined based on the quartiles computed for the sum of states' expenditures (including District of Columbia), thus first class contains 13 states with the lowest values of that measure, and the fourth class has 13 states with the highest values. The knowledge sought should allow us to better distinguish between these classes. As opposed to the previous problem, here the attributes contain numerical data.


Run iAQ with your own data
iAQ
Applying iAQ to your own problem and testing the rules.
Click on the "Run" button below when the input data have been entered.
Program capabilities and User's Guide
Examples of training and testing data
How to prepare the training data
List of all program parameters
How to prepare the testing data
Enter your own data
WORKSHEET
The 'Your Data' feature is still under development. Only 'Enter your data' option is working (partially).
Open
Save As
Show Output
Paste Example
Show Example
Click the "Next" button below to return to iAQ Menu.
Start
About
Menu
Back
Next
Quit . . .
Quit
Run

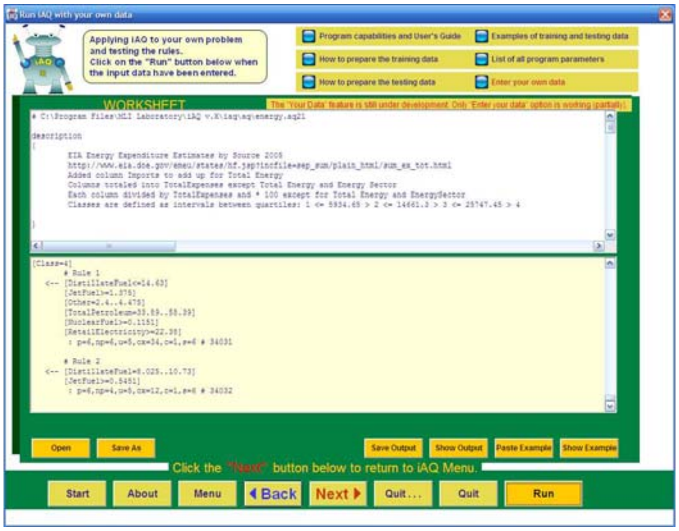
```
description
{
        EIA Energy Expenditure Estimates by Source 2005
        http://www.eia.doe.gov/emeu/states/hf.jsp?incfile=sep_sum/plain_html/sum_ex_tot.html
        Added column Imports to add up for Total Energy
        Columns totaled into TotalExpenses except Total Energy and Energy Sector
        Each column divided by TotalExpenses and * 100 except for Total Energy and EnergySector
        Classes are defined as intervals between quartiles: 1 <= 5934.65 > 2 <= 14661.2 > 3 <= 25747.45 > 4

}

attributes
{
        State nominal (Alabama,Alaska,Arizona,Arkansas,California,Colorado,Connecticut,Delaware,DistCol,Florida,Georgia,Hawaii,
        Coal continuous 0, 100
        NaturalGas continuous 0, 100
        DistillateFuel continuous 0, 100
        JetFuel continuous 0, 100
        LPG continuous 0, 100
        MotorGasoline continuous 0, 100
        ResidualFuel continuous 0, 100
        Other continuous 0, 100
        TotalPetroleum continuous 0, 100
        NuclearFuel continuous 0, 100
        Biomass continuous 0, 100
        TotalPrimary continuous 0, 100
        Imports continuous 0, 100
        ElectricPowerSector continuous
        RetailElectricity continuous 0, 100
```

Fig. 9. Running iAQ with the user's own problem. Fragment of the input file for the energy expenditures characterization problem.

Fig. 9 presents a screen with the loaded input file for the problem, and Fig. 10 shows the result window after running the rule discovering module with some rules characterizing fourth class.


Run iAQ with your own data
iAQ
Applying iAQ to your own problem and testing the rules.
Click on the "Run" button below when the input data have been entered.
Program capabilities and User's Guide
Examples of training and testing data
How to prepare the training data
List of all program parameters
How to prepare the testing data
Enter your own data
WORKSHEET
The 'Your Data' feature is still under development. Only 'Enter your data' option is working (partially).
Open
Save As
Save Output
Show Output
Paste Example
Show Example
Click the "Next" button below to return to iAQ Menu.
Start
About
Menu
Back
Next
Quit . . .
Quit
Run

```
# C:\Program Files\MLI Laboratory\iAQ v.X\iaq\aq\energy.aq21

description
{
        EIA Energy Expenditure Estimates by Source 2005
        http://www.eia.doe.gov/emeu/states/hf.jsp?incfile=sep_sum/plain_html/sum_ex_tot.html
        Added column Imports to add up for Total Energy
        Columns totaled into TotalExpenses except Total Energy and Energy Sector
        Each column divided by TotalExpenses and * 100 except for Total Energy and EnergySector
        Classes are defined as intervals between quartiles: 1 <= 5934.65 > 2 <= 14661.2 > 3 <= 25747.45 > 4

}
```

```
[Class=4]
      # Rule 1
  <-- [DistillateFuel<=14.63]
      [JetFuel>=1.375]
      [Other=2.4..4.479]
      [TotalPetroleum=33.89..58.39]
      [NuclearFuel>=0.1151]
      [RetailElectricity>=22.38]
       : p=6,np=6,u=5,cx=34,c=1,s=6 # 34031

      # Rule 2
  <-- [DistillateFuel=8.025..10.73]
      [JetFuel>=0.5451]
       : p=6,np=4,u=6,cx=12,c=1,s=6 # 34032
```

Fig. 10. Running iAQ with the user's own problem. A fragment of the output file for the energy expenditures characterization problem is shown.

Table I contains summary of the rules learned for this problem. Although this analysis in presented only for illustrative purposes, it may be interesting to see within each class how differently states are grouped based on their expenditure structure. For example the largest group in the first class seems to have more limited financial spending on

[1] Available at : http://www.eia.doe.gov/emeu/states/_seds.html

TABLE I
SUMMARY OF DISCOVERED RULES FOR THE ENERGY EXPENDITURES CHARACTERIZATION PROBLEM

Class 1	Class 2	Class 3	Class 4

NaturalGas <= 14.24
MotorGasoline <= 32.87
TotalPetroleum >= 58.88
RetailElectricity >= 21.14
	NaturalGas = 11.62..22.93
	LPG = 0.6401..3.33
	MotorGasoline = 20.99..32.41
	ResidualFuel <= 0.8399
	TotalPetroleum = 47.65..59.8
		Coal >= 1.17
		NaturalGas >= 10.18
		DistillateFuel = 11.24..16.14
		LPG <= 3.755
		MotorGasoline >= 25.9
		Biomass = 0.1351..0.405
			DistillateFuel <= 14.63
			JetFuel >= 1.375
			Other = 2.4..4.475
			TotalPetroleum = 33.89..58.39
			NuclearFuel >= 0.1151
			RetailElectricity >= 22.38

JetFuel <= 2.135
ResidualFuel >= 0.8051
RetailElectricity <= 27.74
	DistillateFuel = 14.14..18.92
	Other = 2.405..3.285
		DistillateFuel = 13.41..15.18
		LPG <= 1.665
		ResidualFuel <= 1.475
			DistillateFuel = 8.025..10.73
			JetFuel >= 0.5451

JetFuel = 1.16..1.31			RetailElectricity = 26.02..26.36
RetailElectricity <= 17.85			

Each column represents a description of a given class, and each cell in the column represents one rule describing some of the examples belonging to that class. Rules are ordered from the top to the bottom in the decreasing number of covered examples. In most cases there is some overlap between the rules belonging to one class – the same example can be covered by many rules. Rules consist of conjunction of conditions, and one line contains one condition. Numbers denote percentage of the sum of expenditures, and their range is between 0 and 100%.

natural gas than the largest group in the second or third class. On the other hand, the portion of expenses in the "Total Petroleum" category in the largest group of the first class is higher than in the largest groups in classes 2 and 3. Perhaps, the states that spend more money on energy assign larger part to fuels other than petroleum. Of course, further analysis requires involvement of an expert in that field, but such findings may be a strong encouragement for the user to learn more about machine learning and artificial intelligence methods.

IV. CONCLUSION AND FUTURE WORK

This paper presented iAQ, an interactive, multimedia-based tutorial system that enables users to experiment with various Natural Induction methods, and is intended for wide audience with varying level of familiarity with the topic.

The role of the system in the context of virtual organizations and open learning environments was discussed.

The features of the program were reviewed and illustrated with the screenshot examples. The ability of the program to allow the user to experiment with his/her own data was demonstrated with the problem of analyzing energy expenditures at the state level.

Future work may include developing a web based version (preferably with the use of Web 2.0) of the system, providing more ready-to-use, well documented examples from various scientific fields, extending the scope of the demonstration beyond the "robots" domain, modernization of the GUI, implementing support for the XML format for representing problems and results, using better speech-to-text technology, and developing of an animated illustration of how different stages of the presented algorithms work.

IN MEMORIAM

The author dedicates this paper in memoriam Ryszard S. Michalski, the *spiritus movens* behind the creation of iAQ and its predecessors.

ACKNOWLEDGMENT

The author would like to thank Dr. Janusz Wojtusiak for very engaging discussions and Dr. Kenneth Kaufman for his comments which helped to improve this work.

REFERENCES

[1] Cummings, J., Finholt, T., Foster, I., Kesselman, C. and Lawrence, K. , "Beyond Being There: A Blueprint for Advancing the Design, Development, and Evaluation of Virtual Organizations.".

[2] Atkins, D. E., Brown, J. S. & Hammond, A. L. (2007). A Review of the Open Educational Resources (OER) Movement: Achievements, Challenges, and New Opportunities. (online): OERderves. Retrieved July, 2007 from http://www.oerderves.org/wp-content/uploads/2007/03/a-review-of-the-open-educational-resources-oer-movement_final.pdf

[3] Zurada, J. M., Wojtusiak, J., Chowdhury, F., Gentle, J. E., Mazurowski, M.A. and Jeannot, C., "Computational Intelligence Virtual Community: Framework and Implementation Issues," Proceedings of the IEEE World Congress on Computational Intelligence, Hong Kong, June 1-6, 2008.

[4] Michalski, R. S., Kaufman, K., Pietrzykowski, J., Wojtusiak, J., Mitchell, S. and Seeman, W.D., "Natural Induction and Conceptual Clustering: A Review of Applications," Reports of the Machine Learning and Inference Laboratory, MLI 06-3, George Mason University, Fairfax, VA, June, 2006 (Updated: August 23, 2006).

[5] Kaufman, K., Michalski, R. S., Pietrzykowski, J. and Wojtusiak, J., "An Integrated Multi-task Inductive Database and Decision Support System VINLEN: An initial implementation and first results ," Presented at the 5th International Workshop on Knowledge Discovery in Inductive Databases, KDID'06, in conjunction with ECML/PKDD, Berlin, Germany, September 18, 2006.

[6] Kaufman, K. and Michalski, R. S., "EMERALD 2: An Integrated System of Machine Learning and Discovery Programs to Support Education and Experimental Research," Reports of the Machine Learning and Inference Laboratory, MLI 93-10, School of Information Technology and Engineering, George Mason University, Fairfax, VA, September, 1993.

Conservation of Information (COI). A Concept Paper on Virtual Organizations and Communities

W.F. Lawless
Paine College
1235 15th Street
Augusta, GA 30901-3182
706-821-8284 office
706-664-8148 cell
lawlessw@mail.paine.edu

Donald A. Sofge
Natural Computation Group
Navy Center for Applied Research in Artificial Intelligence
Naval Research Laboratory
Washington, DC, USA
donald.sofge@nrl.navy.mil

Abstract

As a work in progress, we have conceptualized virtual organizations and communities based on the conservation of information (COI). The literature indicates that applying social science to human or virtual organizations has been ineffective. Our approach is to unravel the fundamental interdependence between agent-based observation and action (an agent represents a human, machine or robot able to surveil its environment including itself and report on its observations). Tentatively, we propose that social theory parallels the quantum model, with the commonality between interdependence and entanglement, allowing us to borrow from quantum mathematics to develop a computational model.

I. Introduction

Virtual communities combine situational learning and evolution with action skills. However, researchers incorporating agent mobility into models of virtual communities with computational intelligence-machine learning (CI-ML) often draw more from personal experience with human behavior and community than from perspectives based on social theory. Among the reasons to avoid social theory for applications of computational intelligence to mobile agents is the present inability to solve evolving human problems with social theory (e.g., for military applications, see [5][18]; for the lack of a theory-based knowledge of effective decision-making, see [41]; and for our perspective, see the lack of an effective theory of organizations in [34], organizational performance in [26], and organizational metrics in [22].

We have attributed the weak state of social theory for computational modeling to the lack of a fundamental relationship between observation and action information [27]. For example, while it is common to find strong associations between self-reported behavior and self-esteem, arguably one of the most studied phenomenon in psychology, based on a meta-analysis with over 30 years of data, Baumeister and his colleagues [1] were surprised to find only a weak correlation between self-reported self-esteem and actual academic or work performance. Lawless and his colleagues [28] found no association between the knowledge of air-combat maneuvering held by combat fighter pilots and their results in air combat. And, for game theory, the first mathematical model of interdependence, Kelley [19] gave up after a career of troubleshooting the lack of an association between prior stated preferences and actual game choices, a problem that has kept game theory from becoming a predictive science [37]. This poor state linking computational social theory to actual human behavior has opened the way to develop and apply a new theory for human and artificial agent organizations not based solely on researcher or agent observations. Instead, we have concluded that for it to be effective, a computational theory of intelligence must include interdependence between agent observations and behaviors [28].

The general strategy in social science assumes that data to be analyzed statistically is derived from independent sources [20]. The goal for the analyst of social data is to remove or control the effects of statistical interdependence in data. Dawes and his colleagues [10] found similarly that, compared to the estimates by clinical subject matter experts (SMEs), actuarial data was more reliable and valid than expert witnesses of human behavior due to observer dependencies. Providing support, Tetlock [43] concluded that decisions by SMEs are no better than 50 percent of the time. Nonetheless, what makes our goal unusual is the opposite: instead of statistical techniques to remove observer dependencies, we plan to remove observer independencies to establish a science of interdependent systems and social processes (e.g., measuring deindividuation rates among the citizens of Burma imposed by the reigning military dictatorship may indicate geospatially and over time variability in the amount of power expended to oppress Burmese citizens). By extension, measuring the degree of interdependence among computational agents measures virtual community (Eqn. 5 & 6).

We have concluded that interdependence is fragile; measuring it produces only classical information; and it cannot be reproduced [30]. Interdependent beliefs are also conjugate. But the measurement of two conjugate beliefs creates a measurement problem reflected as COI tradeoffs (Eqn. 1).

To make room for a new theory of computational intelligence for mobile agents, the failure with human experience as a resource for building virtual organizations and virtual communities had to addressed. From our research, briefly, we have found that successful decision-making in human organizations was more likely to be practical and based on a competition of ideas (majority rule) using risk determinations by a handful of participants who drove the process to reach hard-fought compromises, but rending any rational perspectives arising from these decisions; in contrast, when a consensus view was purposively sought, it was often reached when decisions were less concrete and more likely to have uncertain effects by incorporating risk perceptions or illusions into the decisions ([29],[45]). Counterintuitively, we have also found for majority rules that after conflict was resolved with a compromise between protagonists, a consensus was quickly reached, which we have labeled an "action consensus".

We propose that these cognitive-action tradeoffs are characteristic of the conservation of information (COI) among four interdependent variables: Situational (localized) knowledge interdependent with plan execution; and energy expenditures from available resources interdependent with the duration of resource expenditures. For multiple interdependent events, we expect a competition to either focus attention and action, or fragment them among these four factors. Aware that the mind creates cognitive-motor maps of physical and social reality, we postulate that coherent thoughts reflect coherent maps of and actions in reality; and that incoherent thoughts reflect fragmented cognitive maps of and actions in reality (for robot generated consensus maps of their environment, see [44]).

For knowledge-execution factors, coordinating one event interdependent with an event sequence in a business or military chain implies a center of gravity (COG) for a system interdependent with the distance between the frequencies of signals sent to coordinate the occurrence of a target event among a series of interdependent events. COG is either the physical centroid of a team [40]; organization; plan participants; or the landscape that is key to a plan. For example, in dealing with terrorist acts, it is "a Gaussian distribution centered … between key features and the event … [where the] terrorists prefer certain spatial features (consciously or not), such as buildings or streets near the target location" [15]. With c as a constant,

$$\Delta x_{COG}\, \Delta(1/\lambda)_{COG} \geq c, \qquad (1)$$

where Δx_{COG} is the uncertainty in locating the center of gravity (COG) of a target (key activity or plan), while uncertainty in the distance between a chain of interdependent events coordinated around a COG's sequence of events, including the target event, is $\Delta_i(1/\lambda)_{COG}$. Here $\Delta_i(1/\lambda)$ equals to $\Delta_i k$, the wave number. Equation (1) measures tradeoffs among the decisions made for the interdependent activities enacted in a virtual community.

In reaching organization decisions, we have found generally that majority rule was about four times faster than consensus rule. Similar to the reduced mass approach [13], we expect that the COG for a group discussion centers around the cognitive-action resistance weights of its protagonists (i.e., reactance or resistance to adopting the target plan or its associated chain of events is higher under cooperation). Let μ be the reduced mass of the COG for both majority and consensus rules. Then,

$$1/\mu = 1/m_1 + 1/m_2 \qquad (2)$$

With μ, we model two citizen groups, advising the Department of Energy on environmental cleanup of nuclear wastes at DOE sites, for which we have had experience. A majority rule group (MR) with 25 members requires at least 13 for a favorable decision, and a consensus rule (CR) group of 31 members requires at least 27 members for a favorable decision [26]. Assuming on average four protagonist in the MR group versus 27 in the CR group, and assuming they are of equal strength (an arbitrary 10 score for protagonists versus 1 for regular participants) produces a contrast resistance of about 3.48. This number is close to our field result of 4 based on an average of 2 hours for CR to make a decision versus ½ hour for MR decisions. Equation (2) measures decision resistance or belief fragmentation in a virtual community. In addition, we explored a variation of Grover's quantum search algorithm to measure decision resistance (i.e., $N_{consensus}/2 \div \sqrt{(N_{majority\ rule})} = 27/2 \div \sqrt{(13)} \approx 3.75$).

Revising Equation 1, with ΔE as resource uncertainty (the entropy associated with resources available to execute the target activity and its associated chain of events) and Δt as time uncertainty, gives [27]:

$$\Delta E \Delta t \geq c. \qquad (3)$$

From Cohen [6] and Rieffel [36], we have revised Equation (3) to form Fourier transform pairs [30]:

$$\sigma_f^2 \sigma_t^2 \geq 4 \qquad (4)$$

with σ_f as the standard deviation of the frequency distribution and σ_t as the standard deviation of the time distribution. Equation (4) assumes that the signals from an agent's motor controller sent to its motor drives can be treated with signal detection theory (SDT). It means that short duration signals are associated with broad frequency distributions or, conversely, that a narrow bandwidth is associated with a long duration signal.

For organizations, we had found that more effort (i.e., power) was expended under competitive than consensus rules. From another direction, human communities are built from multiple mergers of smaller organizations and communities. With multiple regressions and Fourier transform pairs, we have extended our results to mergers between organizations, finding an association between increasing market size and reduced volatility (i.e., beta[1]), implying that one reason for organizations to merge and grow in size is to marginally decrease the resource uncertainty in controlling a market [30]. More stable organizations respond at lower frequencies to market perturbations.

Our results match findings for the brain: the greater expenditures of energy (power) in the brain are associated with higher cognitive functions, leading to an increase in the ability to resolve mental maps of reality [17]; words spoken in anger expend about twice the energy of regular voice [26]; and when performing a complex military exercise, compared to experts, the brains of novices light up like a Christmas tree, indicating the increased energy wasted by novices compared to experts ([23]; see also [32]).

2. Proposed Social Decision Model (SDM):

Using natural computation, we plan to model and study consensus and majority rules in making decisions. In our study, we plan to use recombination operators [11]. For these operators to be able to drive the evolution of machine control algorithms based on Darwin's survival-of-the-fittest [46], we will use binary tournament selection based on a competition between pairs witnessed and evaluated by other machine agents. This is the same basis of political campaigns common to democracies, found to best educate undecided (neutral) voters among the public [7] [however, we have not yet resolved how to employ neutral agents]. Based on Shannon information theory [8], the reason is straightforward: cooperation reduces the information available to observers; competition increases it. The joint uncertainty $I(x_1,x_2)$ between two agents is

$$I(x_1,x_2) = I(x_1) + I_{x1}(x_2) = I(x_2) + I_{x2}(x_1). \quad (5)$$

Equation (5) is the uncertainty in one variable combined with that in the other after removing knowledge of the first. $I(x_1,x_2)$ ranges between $I(x_1)$ or $I(x_2)$ at the minimum when both are equal but one controls the other (enforced cooperation), to $I(x_1)$ plus $I(x_2)$ when both are independent (competition). The information transmitted, or $I_T(x_1{:}x_2)$, between two agents is:

$$I_T(x_1{:}x_2) = I(x_1) + I(x_2) - I(x_1,x_2)$$
$$= I(x_2) - I_{x1}(x_2). \qquad (6)$$

Equation (6) measures the amount of uncertainty that one variable interdependently reduces in the other. The constraint ranges between [0, min$\{I(x_1),I(x_2)\}$] as x_1 and x_2 range between independence to interdependence (i.e., conjugate). With the perspective of Shannon information, interdependence increases when cooperation occurs in a system, business chain, but also under competition when two or more opponents are coordinating their activities around a common objective (e.g., courtroom). Equation (3) will measure pre-decision information among virtual agents; Equation (4) will measure decision interdependence.

Measurement. We speculate that better decisions occur when the self-interests of expert agents (e.g., defense attorneys and prosecutors) are maximized [12]. If we assume that dialectics are composed of polarized views (180 degrees apart; i.e., A is true, $\neg A$ is false), producing a random outcome and a greater chance of conflict, orthogonal views are composed of independent (alternative) views, implying value associated with orthogonal belief systems (modeled with the dot product between two beliefs, A and B; from [14]). Maximally orthogonal beliefs offer several advantages [4]. From our perspective, we postulate that the primary

[1] Beta is the covariance between a target organization and the average of all organizations in a virtual community divided by the variance of the target.

reason is the ability to model interdependence, which we have asserted in the past is similar to the entanglement between qubits (where a qubit is a linear superposition of 0 and 1 bits of information). As noted earlier, interdependent beliefs are conjugate, and the measurement of one of two conjugate beliefs creates a measurement problem for the remaining belief. Measuring one belief conjugate to another produces Von Neumann entropy (the Von Neumann entropy becomes the Shannon entropy only for orthogonal states) in the measured belief and the maximum Shannon entropy in the other; i.e., Von Neumann entropy goes to zero as one belief becomes fully known, the other becomes random, approaching one [8].

In classical science, a system's state is specified by its observable properties at any one point in time (statics) or evolution over time (dynamics). Measurement copies a system's properties. In classical science, there are no entangled states. In general, *n*-bit systems require *n* times as much information as single bit systems. For interdependent systems, however, full descriptions are not possible, only the measured observation of results from interactions constrained by the probabilities inferred for future outcomes. Interdependent state spaces are Hilbert spaces with 2^n dimensions, such that a superposition occurs for 2^n *n*-qubit states. Measurement disturbs the association between a system's conjugate interdependent variables, forming a tradeoff between information gain and disturbance. Our plan is to monitor entropy at the individual, organizational and community levels.

Neutrals. In our social decision model (e.g., juries, citizen advisory boards, etc.), neutrals decide outcomes. Neutrals serve other important functions. Social tension is maximized under polar opposite views, increasing the opportunity for conflict. The presence of neutrals reduces the probability of conflict [21]. Futures markets work by employing investors who are neutral to the overarching topic they are investing in but not neutral to making a profit [29]. And neutrals are where most new learning and evolution take place [30]. But again, how we plan to employ neutral agents is not solved.

Feedback. All things being equal, SDM (e.g., jury) should lead to marginally better decision-making, but no guarantee exists that it will be better. We speculate that the key ingredient is feedback. Feedback is the primary mechanism that distinguishes democracies, especially those with limited and counterbalancing centers of power [16], from those using censorship (autocracies) in exchange for stability [31]. The result with feedback in a democracy is an increase in accountability and trust [25]. For our model, individual agents will be controlled with evolutionary algorithms [47]. For virtual organizations and communities, while we have not resolved our plans with control, we plan to follow Csete and Doyle's model ([9]; see Fig. 1).

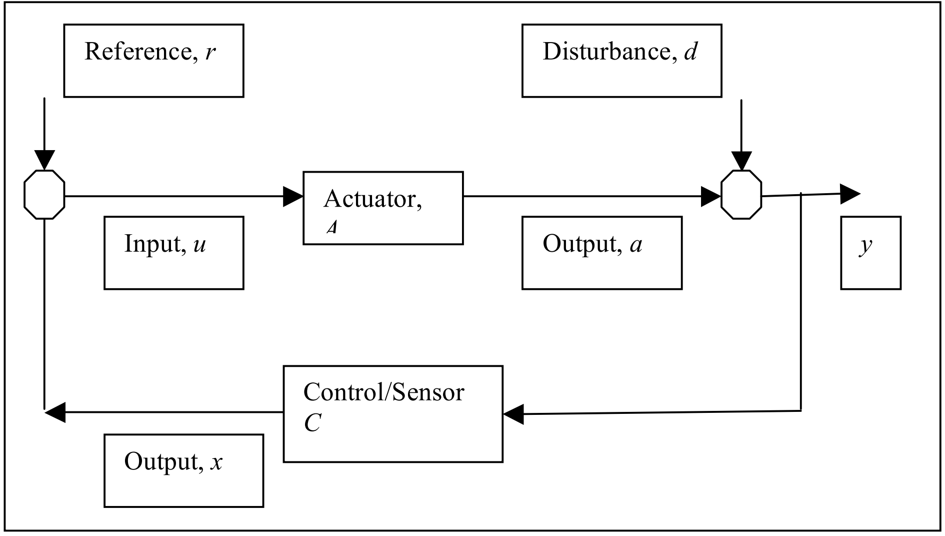


Figure 1. A notional control system is illustrated (from [9]). In general, positive and negative feedback, *F*, apply under static or steady state conditions; with *S* as the deviation from perfect control, log $|S|$ applies to the dynamic case. The goal of a control system is to amplify reference *r* independent of disturbance *d* (where *r* = mission, plan or goal). Working as a low-pass filter, controller *C* removes short-term oscillations to follow long-term average trends. That makes *A* a pure integrator with gain *g*. The goal for a system's *C* is achieved when *F* is negative and $-1/C >> 1$, in the limit making $|S| -> 0$. With Log $|S|$ as control fragility, when log $S < 0$, dynamic control is achieved; in contrast, when log $S > 0$, *d* is amplified.

3. Works in progress: Ensemble model

As of yet, we do not have a strategy for combining the results into an ensemble. Good classifier performance with training data designed to learn an underlying distribution has often not performed well with data not seen during training. In a review, Polikar [35] concluded that combining classifiers into an ensemble generalizes better by reducing the risk of selecting a poor classifier. This becomes important when decision boundaries among different data classes are complex, as we expect them to be for mobile agents. Ensembles of linear classifiers can learn complex nonlinear boundaries. Our plan is to contrast consensus and majority rules to data fusion processes. We will attempt to do this by increasing diversity of classifiers with decision boundaries different from those of others. Similar to a low pass filter, our plan is to combine a diversity of weak classifiers into a stronger classification system.

Our algorithm is incomplete at this time. But because the mean is simple to use and found to be effective [35], we plan to build from means to standard deviations and then to fitted Gaussian distributions for the data. This would permit us to introduce Fourier pairs to explore patterns of frequencies and wave numbers, Δk, where Δk equals $\Delta(1/\lambda)$. In addition, to speed up runtimes, we plan to use a control system that combines simulation and reality [46]. A similar control system was found by Bongard and his colleagues [2] to be effective at path disambiguation for robots.

4. Summary

Natural computation models will permit us to test field data and model the organizations that produce this data. We propose to test the data and organizational models with artificial agents evolved using biologically inspired natural selection [11] and social methods of decision-making (e.g. jury or "voting" ensembles). Based on our field research, we predict longer decision times and more oscillations under consensus rule (CR) than majority rule (MR). We expect CR to model serial sequential individual decision processes. Surowiecki [39] gave evidence and case studies of why human ensembles (crowds) often outperform individual experts. Earlier, Opitz and Maclin [33] empirically showed that ensembles often outperform individuals, with theoretical support provided by Brown [3] and Tang [42].

5. References

[1] Baumeister, R. F., Campbell, J.D., Krueger, J.I., & Vohs, K.D. (2005,January). "Exploding the self-esteem myth." Scientific American.

[2] Bongard, J., Zykov, V., & Lipson, H. (2006). "Resilient machines through continuous self-modeling." Science **314**: 1118-1121.

[3] Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005). “Diversity creation methods: A survey and categorization,” Journal of Information Fusion, vol. 6, pp. 5-20.

[4] Bub, J. (2006). Quantum entanglement and information, Stanford Encyclopedia of Philosophy, retrieved on 8/20/08 from plato.stanford.edu/entries/qt-entangle/.

[5] Carley, K., Director, CASOS (2008, June 25). Formal interview on D-P, at Carnegie Mellon University, Pittsburgh, PA. In attendance: Roger Hillson, AIT-NRL, Helen Purkitt, US Naval Acacemy; Cheryl Giammanco, ARL; and Bill Lawless, Paine College.

[6] Cohen, L. (1995). Time-frequency analysis: theory and applications, Prentice Hall Signal Processing Series.

[7] Coleman, J. J. (2003). The benefits of campaign financing. CATO Institute Briefing Papers, www.cato.org/pubs/briefs/bp-084es.html. Washington.

[8] Conant, R. C. (1976). "Laws of information which govern systems." IEEE Transaction on Systems, Man, and Cybernetics **6**: 240-255.

[9] Csete, M. E., & Doyle, J.C. (2002). "Reverse engineering of biological complexity." Science **295**: 1664-69.

[10] Dawes, R. M., Faust, D., & Meehl, P.E. (1989). "Clinical versus actuarial judgment." Science **243(4899)**: 1668-1674.

[11] De Jong, K. A. (2008, February). "Evolving intelligent agents: A 50 year quest." Computational Intelligence Magazine, vol. 3, number 1, IEEE, pp. 12-17.

[12] Freer, R. D., & Perdue, W.C. (1996). Civil procedure. Cincinatti, Anderson.

[13] French, A. P., & Taylor, E.F. (1979). An Introduction to Quantum Physics. Cambridge, MIT Press.

[14] Goldstein, M. (1986). "Exchangeable Belief Structures." Journal of the American Statistical Assn **81(396)**: 971-976.

[15] Goffeney, J., Schmidt, G.S., Dalton, J., D’Archangelo, J., & Willis, R. (2006, Oct 29-Nov 3), *Forecast Visualizations for Terrorist Events, Poster* IEEE Visualization Conference.

[16] Hamilton, A., Madison, James, Jay, John (1787-1788). The Federalist Papers, New York newspapers.

[17] Hagoort, P., Hald, L., Bastiaansen, M., & Petersson, K.M. (2004). "Integration of word meaning and world knowledge in

language comprehension." Science **304**: 438-441.

[18] Hartley, D. S., III (2008). DIME/PMESII modeling, DSH-08-01. Oak Ridge, TN, Hartley Consulting.

[19] Kelley, H. H. (1992). "Lewin, situations, and interdependence." Journal of Social Issues **47**: 211-233.

[20] Kenny, D. A., Kashy, D.A., & Bolger, N. (1998). Data analysis in soc psy. Handbook of Social Psychology. D. T. Gilbert, Fiske, S.T., & Lindzey, G. (Eds). McGraw. **I:** 233-268.

[21] Kirk, R. (2003). More terrible than death. Massacres, drugs, and America's war in Columbia, Public Affairs.

[22] Kohli, R., & Hoadley, E. (2006). "Towards developing a framework for measuring organizational impact of IT-enabled BPR." ACM SIGMIS Database **37(1)**: 40-58.

[23] Landers, D. M., and Pirozzolo, F.J. (1990). NAS Panel discussion: Techniques for enhancing human performance. Annual meeting of APA, Boston, MA.

[24] Lawless, W. F., Castelao, T., and Ballas, J.A. (2000a). "Virtual knowledge: Bistable reality and soln ill-defined problems." IEEE Systems Man, and Cybern. **30**(1): 119-126.

[25] Lawless, W. F., Castelao, T., & Abubucker, C.P. (2000b). Conflict as a heuristic in dev of interaction mechanics. Conflicting agents: Conflict mgt in multi-agent systems. C. Tessier, L. Chaudron, and H.J. Muller (Eds). Kluwer**:** 279-302.

[26] Lawless, W. F., Bergman, M., & Feltovich, N. (2005). "Consensus-seeking versus truth-seeking." ASCE Practice Periodical Haz, Toxic, and RadWaste Management **9(1)**: 59-70.

[27] Lawless, W. F., Bergman, M., Louçã, J., Kriegel, N.N. & Feltovich, N. (2007). "A quantum metric of organizational performance: Terrorism and counterterrorism." Computational & Mathematical Organizational Theory **13**: 241-281.

[28] Lawless, W. F., Howard, C.R., & Kriegel, N.N. (2008a). A quantum real-time metric for NVO's. In G. D. Putnik & M.M. Cunha (Eds.), Encyclopedia of Networked and Virtual Organizations. Hershey, PA: Information Science Reference, IGI Global.

[29] Lawless, W. F., Whitton, J., & Poppeliers, C. (2008b). "Case studies from the UK and US of stakeholder decision-making on radioactive waste management." ASCE Practice Periodical Haz, Toxic, and RadWaste Mgt **12(2)**: 70-78.

[30] Lawless, W. F., Poppeliers, C., Grayson, J., & Feltovich, N. (2008c). Toward a classical (quantum) uncertainty principle of organizations. QI08, In Bruza, P., Lawless, W.F., von Rijsbergen, K., Sofge, D., Coecke, B. & Clark, S. (Eds.): Oxford University, Kings College London.

[31] May, R. M. (1973/2001). Stability and complexity in model ecosystems. Princeton, NJ, Princeton University Press.

[32] Milton, J., Solodkin, A., Hluštík, P., & Small, S. L. (2007), The mind of expert motor performance is cool and focused, J Neuroimage, retrieved 8/25/08 Elsevier online.

[33] Opitz, D., & Maclin, R. (1999). "Popular ensemble methods: an empirical study," Journal of Artificial Intelligence Research, vol. 11, pp. 169-198.

[34] Pfeffer, J., & Fong, C.T. (2005). "Building Organization Theory from First Principles." Org Sci **16(4)**: 372-388.

[35] Polikar, R. (2006, Third Quarter), Ensemble based systems in decision making, IEEE Circuits and Systems Magazine, pp. 21-45.

[36] Rieffel, E. G. (2007). Certainty and uncertainty in quantum information processing. Quantum Interaction: AAAI Spring Symposium, Stanford University, AAAI Press.

[37] Sanfey, A. G. (2007). "Social decision-making: Insights from game theory and neuroscience." Science **318**: 598-602.

[38] Sood, A., & Tellis, G.J. (2008, forthcoming). "Do innovations really pay off? ." Forthcoming in Marketing Science; Retrieved 9/15/08 from ssrn.com/abstract=1121005.

[39] Surowiecki, J. (2005). The wisdom of crowds. New York, Random House.

[40] Sukthankar, G. (2008, June 10). Robust and eficient plan recognition for dynamic multi-agent teams, Presentation to the Information Technology Division, Nav Res Lab, DC.

[41] Shafir, E., & LeBoeuf, R.A. (2002). "Rationality." Annual Review of Psychology **53**: 491-517.

[42] Tang, E.K., Suganthan, P.N., & Yao, X. (2006). "An Analysis of Diversity Measures," Machine Lrn, 65, 247-271.

[43] Tetlock, P. E. (2005). Expert political judgment. Princeton, Princeton University Press.

[44] Zlot, R., Stentz, A., Dias, M.B., & Thayer, S. (2002). Market-driven multi-robot exploration (CMU-RI-TR-02-02), Carnegie Mellon University.

[45] Oppenheimer, M., O'Neill, B.C., Webster, M., & Agrawala, S. (2007). "Climate change: The limits of consensus." Science **317**: 1505-6.

[46] Sofge, D.S., Potter, M.A., & Schultz, A.C. (2003), Evolutionary robotics: From behaviorism to embodied cognition. Proceedings Int'l Conf. Computer, Comm, and Control Technologies (CCCT'03), 3: 496-502.

Knowledge Discovery in Data with selected Java Open Source Software*

Carlos Rojas Olfa Nasraoui Nurcan Durak Leyla Zuhadar
Sofiane Sellah Zhiyong Zhang Basheer Hawwash
Esin Saka Elizabeth Leon Jonatan Gomez Fabio Gonzalez Maha Soliman

Abstract

We give an overview of our experience in utilizing several open source packages and composing them into sophisticated applications to solve several challenging problems as part of some of the research projects at the Knowledge Discovery & Web Mining lab at the Universe of Louisville. The projects have a common theme of knowledge discovery, however their application domains span a variety of areas. These areas range from mining Web data streams to mining Astronomy related image data, as well as Web information retrieval in social multimedia websites and e-learning platforms. As is already known, a significant proportion of the effort in any real life project involving knowledge discovery in data (KDD) is devoted to the early and final stages of KDD, i.e., the data collection and preprocessing, and the visualization of the results. Given the nature of the data in our projects, we expose our experience in handling text data and image data as part of the KDD process. In addition to the open source packages that we used, we will briefly present some of the stand-alone software that we developed in the lab, in particular a suite of software for clustering and for stream data mining.

1 Introduction

Among the most interesting features of the Web is the ease with which an individual or group can publish documents, and make them available to everyone. In the case of software this is an astonishing feat, because it means that one can access almost immediately an immense body of working programs and applications. Moreover, as it has happened with the Open Source initiative, the Web has facilitated the cooperation of collaborators scattered around the globe.

We give an overview of our experience in utilizing several open source packages as part of some of the research projects at the Knowledge Discovery & Web Mining lab at the Universe of Louisville. We present with different degrees of detail our projects on mining solar images (Section 2), evolutionary and stream clustering techniques (Section 3), pattern discovery from transactional data streams (Section 4), an open source search engine-based recommender system (Section 5), an integrated engine for image and text search (Section 6), and an enriched search for E-learning (Section 7). In Section 8 we discuss the stages of the knowledge discovery process where we have used some of the Open Source tools. Finally, we conclude our paper in Section 9.

2 Mining Solar Images to Support Astrophysics Research

Motivation: In order to study several problems, such as the coronal heating problem, astrophysicists need many samples containing rare instances of coronal loops. Unfortunately, the identification of these images from online repositories is still done manually, which makes it very tedious and time consuming, thus slowing down the advance of science in this field.

Data: The data for this project is captured by measuring instruments onboard several orbiting satellites that are directed at the Sun. It is publicly available on `http://umbra.nascom.nasa.gov/eit/` in the case of EIT [1], and on `http://trace.lmsal.com/` in the case of TRACE[2].

Our approach: In this project, funded by NASA, and by NSF, we work on developing an image retrieval system based on Data Mining to quickly sift through massive data sets downloaded from online NASA solar image databases and to automatically discover the rare but interesting images containing a special solar event that occurs above the surface of the sun, known as coronal loops, and essential in studies of

[*] At the time of this work, the authors were with the Knowledge Discovery & Web Mining Lab, Dept. of Computer Science and Engineering, University of Louisville, Louisville, KY 40292,USA. Elizabeth Leon, Jonatan Gomez and Fabio Gonzalez are with the Computer Systems & Industrial Engineering Dept., National University of Colombia, Bogota, Colombia.

[1] EIT: Extreme ultraviolet Imaging Telescope on board Solar and Heliospheric Observatory (SOHO): http://umbra.nascom.nasa.gov/eit/

[2] TRACE: Transition Region and Coronal Explorer is a NASA Small Explorer (SMEX) mission to image the solar corona and transition region at high angular and temporal resolution. (http://trace.lmsal.com/)

the Coronal Heating Problem. The project aims at retrieving solar images with coronal loops from online solar image databases such as EIT [3] and TRACE[4]. We rely on image processing and classification techniques to detect images having loop shapes and to locate the coronal loops. The characteristics of coronal loops are captured via several image based features to be trained by various classifiers. The model generated from the best classifier is used in the final coronal loop retrieval application to determine which images, out of a collection of input images, contain coronal loops. The input to the system is a set of images and the output is a list of images having coronal loops and the positions of the coronal loops on the retrieved images. We have presented our results in [1]. Our solar loop mining tool is called *SOLOMON* (SOlar LOop Mining for ONline collections), and uses the following Java Open Source packages:

1. ImageJ[5]: ImageJ is a public domain Java image processing program inspired by NIH Image for the Macintosh.

2. Weka[6]: Weka is a collection of machine learning algorithms for data mining tasks available through a GNU General Public License.

Training Phase of SOLOMON Training starts with an expert-marked solar image data set. For every image, the circumference of the solar disk is divided into blocks. Blocks are labeled automatically as *Loop* if they intersect with any marked loop region, and as *No-Loop* otherwise. After block labeling, specialized image features are extracted from both types of regions, i.e. with and without a loop shape. Then using these extracted features, various classifiers are trained to distinguish *Loop* blocks from *No-Loop* blocks. The flow chart of the training phase is shown in Figure 1.

Preprocessing We start by downloading FITS images in the 171 A^o wavelength, from NASA's EIT repository, then we apply preprocessing techniques such as median filtering, Sobel edge detection, global thresholding, and skeletonization methods to remove noise and enhance contours. These steps are performed using ImageJ.

Feature Extraction After preprocessing, we divide the solar circumference into regions that we call blocks. A block is defined by its position, height, and width values. These blocks are then labeled as either containing solar loops (i.e., Loop class), or not (i.e. NO-loop class). Features are extracted from both intensity level and binary level blocks. A sample of the intensity level and binary level versions of a 'Loop' block are shown in Figure 2.

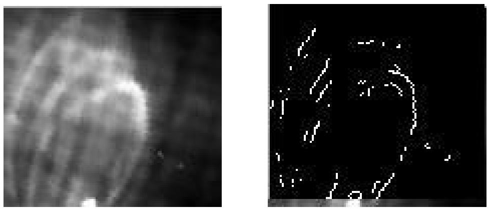

(a) Intensity Level Block (b) Binary Level Block

Figure 2: Intensity (a) and Binary (b) Level Blocks

The features are extracted from each block separately. From the intensity level versions, the following statistical features are extracted: *'Mean'*, *'Standard Deviation', 'Smoothness', 'Third Moment', 'Uniformity',* and *'Entropy'*. The remaining features are extracted from the binary level blocks. The *'Number of Edge Pixels'* is the total number of pixels located on all the edges in the binary blocks. From the Hough transform of the binary blocks, we acquire two features: *'Number of Hough Lines'* and *'Length of Longest Hough Line'* which is the number of points in the global maximum of the Hough Space. Since the directions of the lines in the blocks seemed to be discriminating features, we also computed Edge Histogram descriptors (*'Number of Vertical Edges', 'Number of Horizontal Edges', 'Number of 45^o Edges', 'Number of 135^o Edges',* and *'Number of Non-Directional Edges'*). Considering the distinct spatial edge distribution inside the Loop blocks, we further divided the block into four horizontal bands, and extracted the above edge features from each band separately. Furthermore, we applied our specially designed curve tracing algorithm on all blocks to extract potential loop curves from the midst of other kinds of undesired curves, and computed *Curvature Strength-related features.*

Training Classifiers Based on a training data set containing 403 Loop blocks and 7200 No-loop blocks, we trained several classifier models using WEKA, and obtained the results listed in Table 1, which are based on 10-fold cross-validation.

Image Based Testing Phase of SOLOMON To retrieve solar images containing loop shapes from the EIT solar image repositories, a similar process to the training part is first applied on unmarked (i.e. unlabeled) test images. After preprocessing, block generation, and feature extraction, we applied the best performing classifier model, Adaboost (using C4.5 Decision Trees as a base classifier). The final decision for an image was then made based on the predicted labels of

[3] EIT: Extreme ultraviolet Imaging Telescope on board Solar and Heliospheric Observatory (SOHO): http://umbra.nascom.nasa.gov/eit/

[4] TRACE: Transition Region and Coronal Explorer is a NASA Small Explorer (SMEX) mission to image the solar corona and transition region at high angular and temporal resolution. (http://trace.lmsal.com/)

[5] ImageJ: `http://rsbweb.nih.gov/ij/`

[6] Weka: `http://www.cs.waikato.ac.nz/ml/weka/`

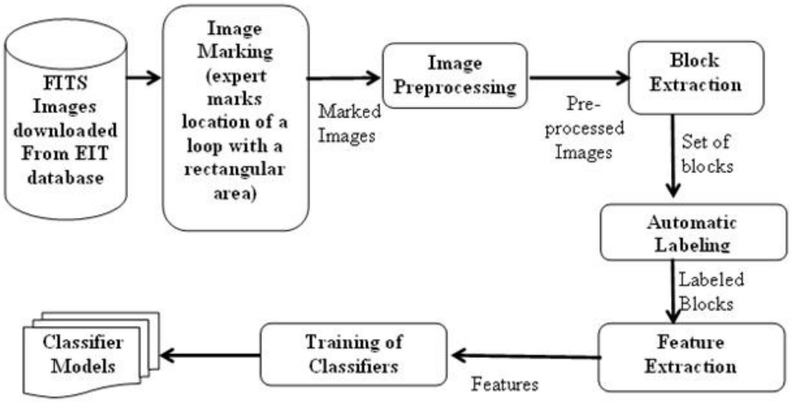


Figure 1: System Structure of SOLOMON Training Phase

Classifier	Precision	Recall
AdaBoost (C4.5)	0.63	0.662
Naive Bayes	0.363	0.768
Multi Layer Perceptron	0.621	0.694
C4.5	0.568	0.563
RIPPER	0.623	0.696
K-NN(k=5)	0.644	0.615

Table 1: Block based Cross Validation Results

Best performing image-based classifier	Precision	Recall
AdaBoost (C4.5)	0.80	0.78

Table 2: Image-based Testing Results on an independent set of online images

the blocks inside the image. If at least one block is predicted as a loop, then the image is classified into the loop class. We finally show all the predicted loop regions on the image based on the location of their Loop blocks, as shown in Figure 3. Consecutive blocks that are classified as loop blocks are merged into a bigger block that is displayed on the output image.

To evaluate the final image retrieval tool, we used images without any markings from the same years as the training data set. The testing set contained 100 images, 50 with coronal loops and 50 without any coronal loops. The final loop mining results are shown in Table 2.

System Implementation: In SOLOMON, both training and testing use the same preprocessing, block extraction, and feature extraction modules. The testing phase, however, has a different interface for the end users who cannot access the training modules. We developed an image downloading tool which connects to the EIT online database and downloads FITS images specified by users from the database, based on their wavelength and date range criteria. We have furthermore developed an additional Coronal Loop Marking tool and patched this tool into ImageJ as a plug-in, to allow our expert-markers to mark the coronal loop regions on the solar images to be used in training. The marked regions were then saved into the FITS image file, by adding their coordinates in the FITS header, which allows them to be read and used by the block labeling module. The preprocessing phase uses mostly operators that are patched into ImageJ. After preprocessing, we divided the images into blocks and labeled them using the markings saved in the FITS format. We have also developed a Block Viewing Tool to enable viewing the blocks and their labels separately.

In the Feature Extraction phase, we extracted statistical features using several function that were already embedded in ImageJ, and implemented the rest of the features in Java. We saved the extracted feature values from each block along with the label information into an arff file which will later serve as the input to WEKA's training modules. In addition, we implemented the final Image Retrieval Tool in Java with options to load different classifier models for classification. In addition to a classifier model, the inputs to this tool are a set of images, of which the tool will output only the images having coronal loops along with the loop locations on these images.

Issues in Open Source Adoption: One limitation of ImageJ, is that it does not provide users to change parameters during the coding for most functions when we call these functions from our own Java code. This includes the default edge detection method, and the size of the median filter masks. Another limitation is the lack of sample code and online discussion forums to support developers having problems. Also, some of the function documentation was not very good.

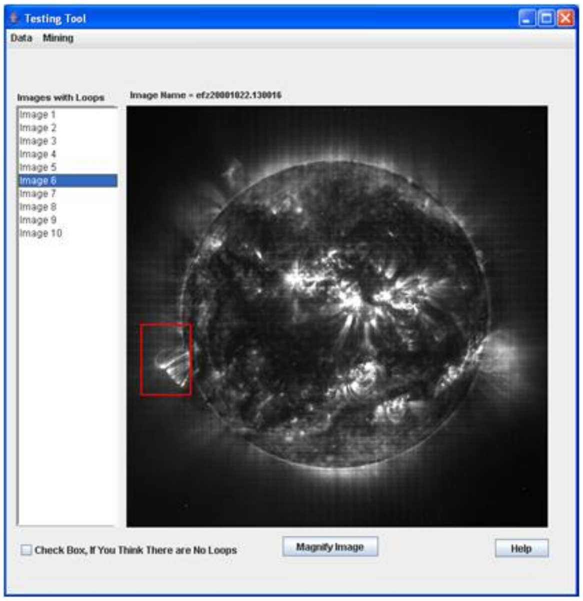


Image Name = efz20001022.130016

Figure 3: Snapshot of Solar Loop Mining Tool

One drawback of WEKA is its slow performance for certain classifiers such as MLP and Adaboost, particularly if the training data is very big. Adaboost may consume between 15 and 30 minutes on a typical Windows workstation (e.g, Pentium 4, with 3 Ghz and 2 GB of memory), and if we run WEKA on the same training file with different classifiers, the performance decreases further after the first classifier, and may run out of memory unexpectedly. Another limitation is that visualizing the ROC curves of several classifiers on the same output is possible with the Knowledge Flow tool of WEKA, but is not possible with the Explorer tool.

3 Evolutionary and Stream Clustering Techniques

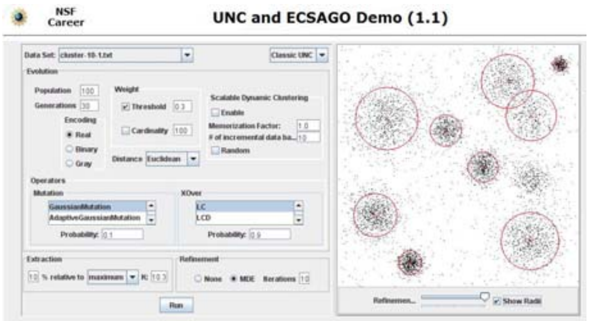
Motivation: Clustering is an important task in knowledge discovery that aims at revealing the hidden structure of an unlabeled data set, typically by partitioning a large data set into groups of similar data called clusters, and by providing descriptions of these clusters in terms of the original features of the data set. Clustering has found successful applications in many domains, ranging from automated taxonomy generation from large text collections to discovering Web user profiles from large access log data. However, many challenges remain open in clustering. These include the difficulty to handle large data sets, noise, and the difficulty to determine the number of clusters automatically.

Data: Most of the synthetic data for this project is publicly available on our website `http://webmining.spd.louisville.edu/NSF_Career/datasets.htm`. Some of the benchmark real data sets used in our experiments to demonstrate performance and success are publicly available on the UCI Machine Learning Repository `http://archive.ics.uci.edu/ml/`. However we have had to be very careful in sharing our real data sets used for Web Usage Mining experiments because of privacy concerns. In general, some of this data is definitely private, and the authority to share it rests with the website owners. However, we plan to sanitize some of the Web log data in such a way that no private information (such as IP addresses) remains. Instead private information will be indexed and obfuscated, leaving only the part of the data (anonymous Web requests) that has no privacy issues left, thus enabling experiments on Web usage mining to be conducted without any risks to the Website users. This data will be made available on `http://webmining.spd.louisville.edu/NSF_Career/datasets.htm`.

Our approach: We developed a family of techniques in Java under the umbrella of the NSF CAREER Award: *New Clustering Algorithms Based on Robust Estimation and Genetic Niches with Applications to Web Usage Mining*. The goals of this project are listed below, and a description of the techniques follows this list.

- Mining an unknown number of clusters in the presence of a significant amount of noise
- Mining evolving user profiles to represent the users' browsing activity on a website.
- Maintaining the currency and accuracy of the Web usage profiles over time.
- Enhancing the scalability of the Web usage mining in the presence of very large Web log data sets.
- Handling the evolution of the input usage data resulting from the dynamic nature of the Web.

The Unsupervised Niche Clustering Algorithm (UNC) The Unsupervised Niche Clustering (UNC) is an algorithm for unsupervised robust clustering based on genetic niching as an optimization strategy. It uses a Genetic Algorithm (GA) to evolve a population of candidate solutions through generations of competition and reproduction. Unlike most other clustering algorithms, UNC can handle noise in the data and can automatically determine the number of clusters. In addition, evolutionary optimization allows the use of any domain-specific optimization criterion and any similarity measure. In particular a subjective measure that exploits domain knowledge or ontologies (as was used for example for web usage mining). However, unlike purely evolutionary search-based algorithms, UNC combines evolution with local Piccard updates to estimate the scale



NSF Career

UNC and ECSAGO Demo (1.1)

Data Set: cluster-10-1.txt   Classic UNC

Evolution

Population 100

Generations 30

Encoding: ● Real ○ Binary ○ Gray

Weight: ☑ Threshold 0.3 □ Cardinality 100

Distance: Euclidean

Scalable Dynamic Clustering: □ Enable

Memorization Factor: 1.0

# of incremental data ba... 10

□ Random

Operators

Mutation: GaussianMutation, AdaptiveGaussianMutation Probability: 0.1

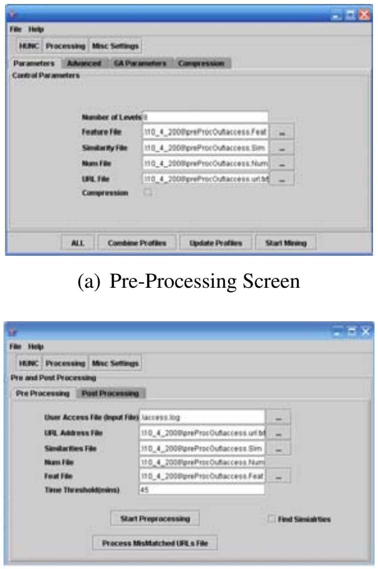
XOver: LC, LCD Probability: 0.9

Extraction: 10 % relative to maximum K: 10.3

Refinement: ○ None ● MDE Iterations 10

Refinement... ☑ Show Radii

Figure 4: UNC Snapshot



Number of Levels: 8

Feature File: \10_4_2008\preProcOut\access.Feat

Similarity File: \10_4_2008\preProcOut\access.Sim

Num File: \10_4_2008\preProcOut\access.Num

URL File: \10_4_2008\preProcOut\access.url.txt

Compression □

(a) Pre-Processing Screen

User Access File (Input File): \access.log

URL Address File: \10_4_2008\preProcOut\access.url.txt

Similarities File: \10_4_2008\preProcOut\access.Sim

Num File: \10_4_2008\preProcOut\access.Num

Feat File: \10_4_2008\preProcOut\access.Feat

Time Threshold(mins): 45

□ Find Similarities

(b) Mining Screen

Figure 5: HUNC Snapshot

of each profile, thus converging fast (about 20 generations). UNC has been successfully used, for instance, in anomaly detection [2], and in clustering spatial data sets [3]. It has been implemented from scratch first in C, then in Java. A snapshot of an online applet implementing UNC is shown in Figure 4. This applet is available on our demo website `http://webmining.spd.louisville.edu/NSF_Career/software/clustering/ECSAGO/demo/`.

The Hierarchical Unsupervised Niche Clustering Algorithm (HUNC) HUNC is a divisive hierarchical version of **UNC**. The implementation of the HUNC algorithm exists in C (older version) and in Java (recent version). All HUNC modules are developed in-house. HUNC is described in [4], and a complete framework and a real case study is presented in [5]. HUNC has proved its effectiveness when compared to other clustering methodologies. In a recent experiment, HUNC profiles were compared to the profiles resulting from traditional pattern discovery, where the entire usage data from all time periods is used to discover usage patterns in one shot. The latter can be considered as the best output possible since all usage data is mined at once. However, HUNC has proved that it too can discover profiles that are as good (or better) than using the traditional one-shot method. Most importantly, HUNC has the critical advantage of enabling scalability in handling very large usage data that makes it impossible to mine all patterns in one shot. A snapshot of the HUNC interface is shown in Figure 5. HUNC can be described as follows

Input: Web Logs (ex: 122.33.124.128 - - [22/Jan/1998:14:19:35 -0600] "GET /faculty.html HTTP/1.0" 304 -)

Output: A set of profiles where each profile consist of a set of URLs with their weight. A sample profile is shown in Figure 7.

Preprocessing Module: the web logs are cleaned by removing all irrelevant requests such as image requests, requests from search agents, and unsuccessful requests. Then the page requests are grouped into units called sessions, where each session represents all the pages visited by a particular user within a predefined period of time. Moreover, a URL index (URL Map) is created that includes all the URLs accessed in the web logs. This map is kept through future pattern discovery phases, and is always updated to reflect new URLs in the web site.

Updating Profiles: The new sessions at time t_i are compared against profiles generated at time t_{i-1}. All unique sessions (Distinct Sessions) at t_i will go through the HUNC mining module. The matching sessions at t_i are used to update the profiles from t_{i-1} which result in the Updated Profiles at t_i.

HUNC Mining: the HUNC Algorithm is used to discover new clusters at t_i from the distinct sessions at t_i .

Post Processing: The sessions in the input data set are matched to the closest cluster, and their URL frequencies are summarized by averaging over all sessions in the cluster. The set of all these URLs and their frequency of access in the same cluster constitute the cluster Profile. This generates a set of New Distinct Profiles at t_i.

Combine Profiles: Combine the updated profiles at t_i and the newly discovered profiles at t_i into one set of profiles which will serve as the Seed Profiles for the next mining cycle at t_{i+1}.

Other clustering algorithms In addition to HUNC, we have developed the following in-house clustering algorithms.

1. ECSAGO (Evolutionary Clustering with Self Adaptive Genetic Operators) and Scalable ECSAGO (in Java): These algorithms are an extension of **UNC** to use self

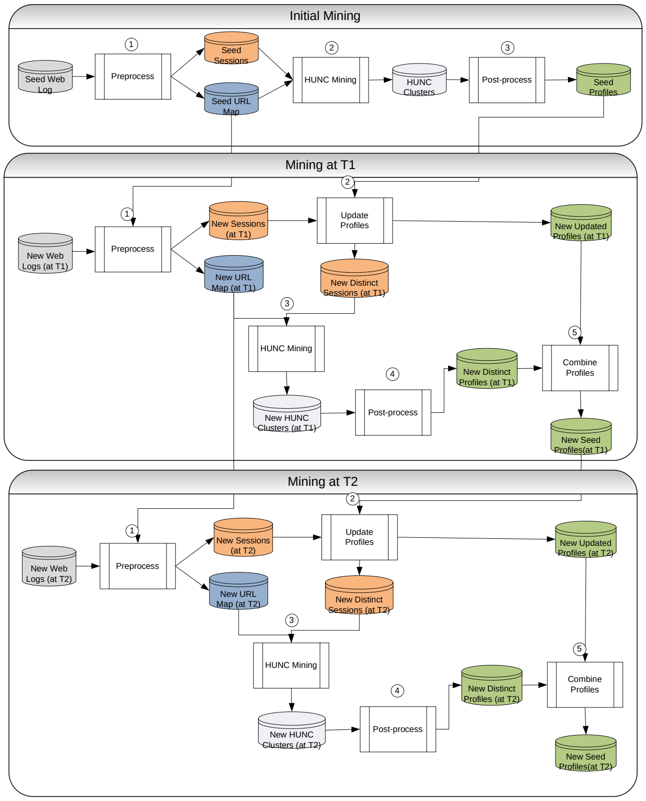


Figure 6: Evolutionary Clustering Methodology

```
Profile:  9, Num.URLS: 11, Cardinality:  58
StartDate:  30/Jan/1998:17:30:33 , End Date:
04/Feb/1998:13:37:34 , Variance:  0.0857
{0.98 - /courses.html}
{0.98 - /courses100.html}
{0.96 - /courses_index.html}
{0.82 - /}
{0.74 - /cecs_computer.class}
{0.34 - /courses300.html}
{0.20 - /courses200.html}
{0.17 - /courses_webpg.html}
{0.12 - /~joshi/courses/cecs352}
{0.10 - /courses400.html}
{0.10 - /people.html}
```

Figure 7: Sample Profile

adaptive genetic operators that dynamically determine the optimal crossover and mutation rates throughout the evolution process. ECSAGO is a single-batch version, while Scalable ECSAGO is a scalable multi-batch data streaming version. A Java applet impementing both algorithms is available on our demo website `http://webmining.spd.louisville.edu/NSF_Career/software/clustering/ECSAGO/demo/`. More details can be found in [6] with successful application to anomaly detection, in particular, network intrusion detection and success on several benchmark machine learning data sets.

2. TECNO-STREAMS (Tracking Evolving Clusters in NOisy data STREAMS): a scalable clustering algorithm that can discover clusters in a single pass over a dynamic data stream. This algorithm uses an optimization method that is inspired by the natural Immune System. It was implemented in Java, and published in [7], and is implemented entirely in Java without any open source adoptions.

3. TRAC-STREAMS (TRacking Adaptive Clusters in data STREAMS): a scalable clustering algorithm that can discover clusters in a single pass over a dynamic data stream. This algorithm uses an optimization method based on alternating gradient-based hill climbing, combined with Chebyshev hypothesis testing for outlier detection, and for merging compatible clusters. It was implemented in C and Matlab, and published in [8].

Issues in Open Source Adoption: Our clustering algorithms are implemented in-house, so we had no issues with open source adoption. On the other hand, we plan to make our source code available through our website via a GNU open source license.

4 Pattern Discovery from Transactional Data Streams

In this project, we focus on analyzing evolving streams of transactional data to discover patterns. In data streams, very large amounts of data are generated during a short time. Data streams can be observed in, for example, the activity of mobile phone networks; the interaction between users, advertisement servers and web publishers; Internet traffic; dynamic text collections, such as e-mail, chats, blogs or news. A particular type of data streams are *evolving* data streams, that reflect ever-changing environments, like e-mail spamming, web usage patterns, or news reports.

Motivation: The nature of data streams adds many difficulties to pattern discovery tasks. In a stream context, space and computational time become critical, nonnegotiable, and scarce resources, thus typically requiring online (incremental) processing, and very small memory (compared to the whole stream). In many cases, the underlying patterns change frequently (e.g., new web pages are added and deleted constantly, and the users themselves learn to interact better with the website) making obsolete what has been discovered from previously collected data. Thus, knowledge discovery techniques that are well suited for these challenges should be able to quickly evaluate current data against the recent past, recognize patterns already seen as well as novelties, and swiftly forget the patterns that the new data no longer supports.

Our approach: Our general approach is to maintain a dynamic memory of the stream, which is constantly updated based on the arriving records, and to maintain a model that reflects the relevant interactions between attributes. Relevant interactions are determined based on statistical tests, such as χ^2 and correlation tests, or on information measures. We applied our approach to the analysis of RSS news feeds and the analysis of newsgroup messages to automatically discover *topics*, defined as sets of keywords that consistently co-occur. We have also applied them to web logs and system audit logs.

Data: We have used mostly text collections, which have the advantages of easy acquisition, including custom collection of the data, and the interpretability of the results. We have used the 20 Newsgroups[7], taking advantage of the rarely used timestamps. Because this data is well known and acceptably accurate, it is possible to build specific scenarios (e.g. mild or strong dynamics) to evaluate our techniques and parameters. We have also collected and used New York Times RSS feeds, and system audit and web logs.

Preprocessing: Because there is no option for repeated passes over the data, preprocessing is limited to quick and simple operations. For example, in our application to text documents, pruning based on frequency is always referred to the current memory, not to global counts. Also, stemming and stop word removal are the vocabulary reduction techniques we use that incorporate some domain knowledge.

Pattern Discovery: Our published work ([9]) includes the use of a dynamic prefix tree as memory, and a graph of relevant attribute co-occurrences as model, extracted periodically using an information-based criterion. We have also used sliding windows as a memory mechanism (based on time and number of records), and an approximation of the distribution of the attribute interactions as a model that is updated permanently.

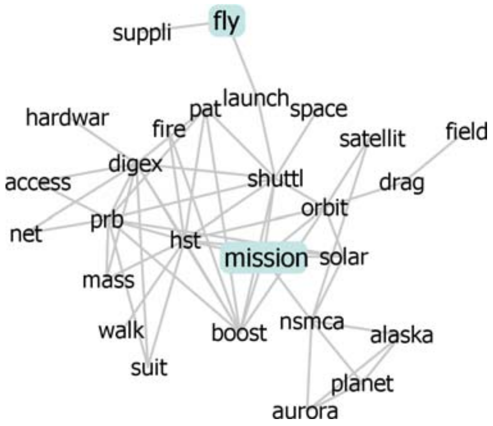


Figure 8: Example of topic visualization.

Visualization: Because our models are based on interactions between attributes, they can be visualized using graphs, which helps to interpret the results. We use the Prefuse visualization toolkit (described below) for this purpose. An example of this visualization is shown in Figure 8, for a topic related to space.

Software: We have implemented our pattern discovery algorithms in Java. Besides, we use the following Java Open Source packages:

- JDOM[8]: "...robust, light-weight means of reading and writing XML data...". Apache-style open source license.
- ROME (<u>R</u>SS and At<u>om</u> utilities)[9]: "...set of open source Java tools for parsing, generating and publishing RSS and Atom feeds...". Built on top of **JDOM**. It has an Apache 2.0 Licensing. We use it for retrieving and extracting text from the RSS feeds.
- Snowball[10]: Open Source (Java, C, and others) implementations of the Porter Stemmer for various languages, with a BSD License. We use it to perform stemming and stop word removal.
- Prefuse Visualization Toolkit[11]: "a set of software tools for creating rich interactive data visualizations. The Prefuse toolkit provides a visualization framework for the Java programming language". Also with a BSD License. Prefuse uses **Lucene**, explained in Section 6.

[7] 20 Newsgroups dataset `http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html`

[8] JDOM: `www.jdom.org/`

[9] ROME: `https://rome.dev.java.net/`

[10] Snowball: `http://snowball.tartarus.org/`

[11] Prefuse `http://prefuse.org/`

Issues in Open Source Adoption: In general, the major difficulty was to find the best way to use the packages, because documentation is sketchy in many cases. Another problem is that, occasionally, the projects are not completely stable, and newer versions are not necessarily backward compatible. However, once an Open Source package was incorporated in the process, the benefit was clear; the applications are usually well written and, thanks to the community effort, better than what could be expected from a single person trying to solve a complex problem completely on their own.

5 Open Source Search Engine-based Recommender System for Multiple Websites

Motivation: Our motivation was the difficulty in implementing Web recommender systems from scratch, particularly when this had to be done very fast for experimentation purposes. Thus our goal is to easily "implement" (existing) recommendation strategies by using a search engine software when it is available, and thus to benefit research and real life applications by taking advantage of search engines' scalable and built-in indexing and query matching features, instead of implementing a strategy from scratch. Thus we developed an application that has the following benefits:

- Multi-Website Integration by Dynamic Linking enabling: (i) dynamic, personalized, and automated linking of partnering or affiliated websites, (ii) Crawling several websites and connecting them through a common proxy
- Giving Control Back to the User or Community (who can set up their own proxy) instead of the website/business
- Taking advantage of the Open Source edge
- Tapping into the established Information Retrieval / Web search legacy.

Our approach: We developed a systematic framework for a fast and easy implementation and deployment of a recommendation system that works on one or several affiliated or subject-specific websites, and based on any available combination of open source tools that includes: **(i)** crawling, **(ii)** indexing, and **(iii)** searching capabilities. A detailed description with some experiments showing success can be found in [10, 11]. The system can provide on-the-fly recommendation for web surfers based on their clickstream data which are transformed into a dynamic user session profile. The recommendations consist of links to pages contained within a given collection of websites that have been previously indexed. The system uses a search engine behind the scene to search for pages that are similar to the user's profile by formulating an implicit query automatically from the user's profile. An inverted index must have been previously formed by crawling, parsing and indexing several participating websites, thus accelerating the recommendation process.

Software: Our implementation is based mostly on open source modules, and is explained in detail on `http://webmining.spd.louisville.edu/open-source-recommender/index.html`. The architecture is shown in Figure 9. We use the following open source components:

- Squid Proxy Cache: We used Version Squid 2.5.STABLE12.
- Nutch Search Engine: Currently we are using version nutch-0.6.

In addition, we implemented our own Recommender System Module in C, which can be downloaded from `http://webmining.spd.louisville.edu/open-source-recommender/codes/recommender.tar.gz`.

Issues in Open Source Adoption: We have had to make several changes to the open source components to make them useful for our purpose. These changes include a modified "client_side.c" code for Squid, and the changed code can be downloaded from our wesbite at `http://webmining.spd.louisville.edu/open-source-recommender/codes/client_side.c`. In order for nutch to support similar page query and termvector query, we modified nutch's source code. Our new source code can be downloaded from our website at `http://webmining.spd.louisville.edu/open-source-recommender/nutch/nutch-0.6_rec.tar.gz`. Also, a list of the changes that we made is available on `http://webmining.spd.louisville.edu/open-source-recommender/nutch/changes.html`.

6 Show And Tell: A Seamlessly Integrated Image and Text Search Engine.

Show And Tell, published in [12], is a Web based image search tool that combines keyword and image content feature querying and search. We used the following Java Open Source packages:

1. Lucene[12]: "Apache Lucene is a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform".

[12] Lucene: `http://lucene.apache.org/`

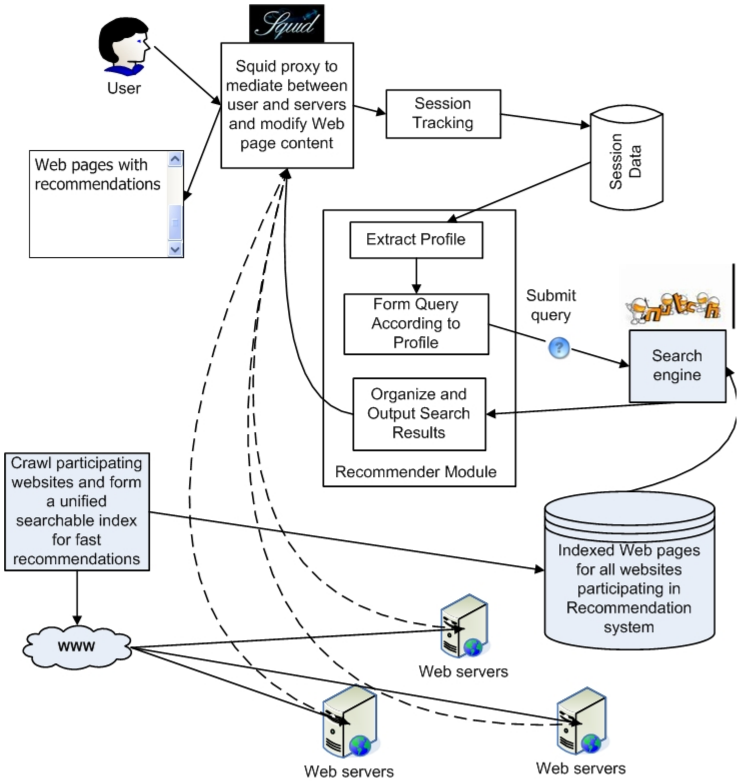

Squid
User
Squid proxy to mediate between user and servers and modify Web page content
Session Tracking
Session Data
Web pages with recommendations
Extract Profile
Form Query According to Profile
Submit query
Search engine
Organize and Output Search Results
Recommender Module
Crawl participating websites and form a unified searchable index for fast recommendations
Indexed Web pages for all websites participating in Recommendation system
www
Web servers
Web servers
Web servers

Figure 9: System Architecture of the open source search-engine-based recommender system.

2. Nutch[13]: Apache "Nutch is an open source web-search software. It builds on **Lucene** Java, adding web-specifics, such as a crawler, a link-graph database, parsers for HTML, and other document formats, etc."

Data: Most of the data for this project is publicly available by crawling various social multimedia websites such as Flickr `http://www.flickr.com/`.

7 Personalized Cluster-based Semantically Enriched Web Search for E-learning:

In this project, we developed an approach for personalized search in an e-learning platform, that takes advantage of semantic Web standards (RDF[14] and OWL[15]) to represent the content and the user profiles, and then using the learner's context to improve the precision and recall in e-learning search, particularly by re-ranking the search results based on the learner's past activities (profile). Our model consists of the following algorithms: (1) bottom-up pruning algorithm to building the learner's semantic profile, (2) learner-to-best cluster mapping algorithm, and (3) re-ranking a learner's search results. **Nutch** is embedded in our "HyperManyMedia" [16] platform so that online students could fetch many different media format resources: text, MS PowerPoint, audio, video, podcast, and vodcast of online resources (lectures). Additionally, we use:

1. Protégé[17]: a Java-based ontology editor and knowledge-base framework. "The Protégé platform supports two main ways of modeling ontologies, via the Protégé-Frames and Protégé-OWL editors. Protégé ontologies can be exported into a variety of formats including RDF, OWL, and XML Schema". Mozilla Public License (MPL).

2. Cluto[18]: a clustering package. We use it to cluster the complete e-Learning domain textual contents. The resulting clusters can later be used to determine each learner's semantic profile. The cluster centroids (keywords) are used both to provide recommendation terms for a specific learner during search, and to add the key-

[13] Nutch: `http://lucene.apache.org/nutch/`

[14] Resource Description Framework (RDF) `http://www.w3.org/RDF/`

[15] OWL Web Ontology Language `http://www.w3.org/TR/owl-features/`

[16] WKU HyperManyMedia Distance Learning Platform: `http://blog.wku.edu/podcasts`

[17] Protégé: `http://protege.stanford.edu/`

[18] Cluto: `http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview`

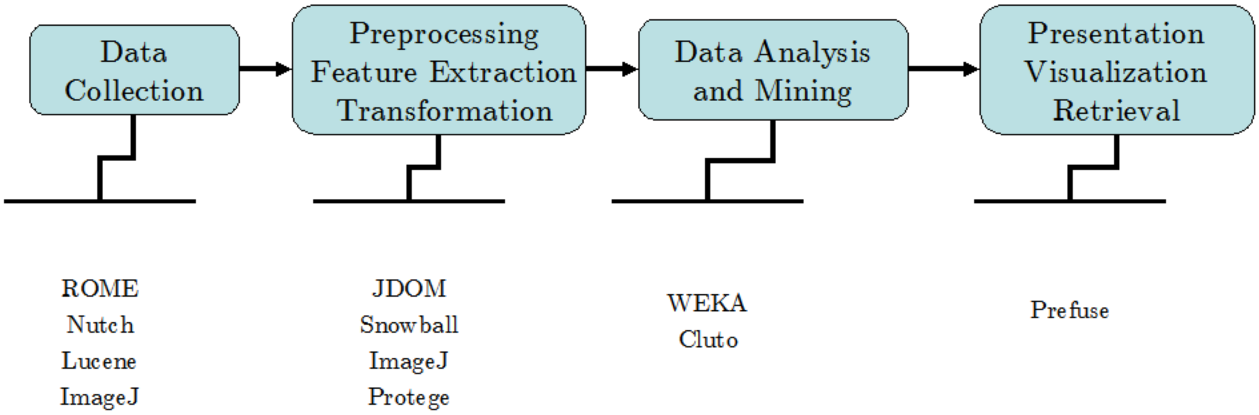


Figure 10: Open Source packages in the KDD process in our projects.

words to the domain ontology as subclass relations. This is publicly available software, but not open source, and free to use in a research environment.

Data: Most of the data for this project is publicly available on WKU's hyper-many media e-learning platform `http://161.6.105.103:8084/nutch-0.9/manymediaplatform.html`.

8 Open Source packages in the KDD process.

As is already known, a significant proportion of the effort in any real life project involving knowledge discovery in data (KDD) is devoted to the early and final stages of KDD, i.e., the data collection and preprocessing, and the visualization of the results. We use them mostly in tasks where well known methods exist, and consequently, there are good Open Source tools available. The following diagram summarizes the packages that we have used, and where they fit in the KDD process. This diagram does not include our own software that was developed in-house, such as the evolutionary clustering and stream clustering algorithms. However this software will be made available through a suitable open source license in the near future, on our wesbite `http://webmining.spd.louisville.edu/NSF_Career/software/software_clustering.htm`. In general, our in-house software touches all the phases of the KDD process.

9 Summary and conclusions.

We have described a variety of projects in which the Knowledge Discovery & Web Mining lab at the University of Louisville has been involved, and we have described the Open Source tools that we have adopted to our purposes. Below, we list certain issues and lessons that we have gathered from these endeavors, besides what was mentioned in the individual projects above.

- At times several Open Source projects attempt to solve the same problem. It is important to be careful in picking the right tool not only in the sense of solving the problem at hand, but also the one that is probably going to be healthy for the duration of the project.

- Open Source tools are completely the opposite of black boxes. They invite the developer to explore them, to change them, and to be an active user (or contributor), rather than just a consumer.

- They are Open Source tools, and in many cases free, but they are *not work-free*, in the sense that one should expect a certain degree of time investment in understanding the tool; and if more complex tasks are required, one must be capable of digging into the code and willing to do it.

- There are a variety of licensing schemes. For research purposes, there are generally no restrictions. However, for commercial purposes, one should read the details, but there are plenty of opportunities if one so wishes. For example, Apache 2.0 licensing allows commercial use, and does not require the additions to be Open Source.

- It is important to try to contribute. Open Source tools are possible because of the good will and skill of competent people who are willing to give their work virtually for free. If one is using those tools, one is probably skilled enough to give some feedback, and thus be a useful contributor to the Open Source community.

Acknowledgment

This work was partially supported by the National Aeronautics and Space Administration under Grant No. AISR-03-0077-0139 issued through the Office of Space Sciences, by the National Science Foundation CAREER Award IIS-0133948 to O. Nasraoui, by NSF grant IIS-0431128, and by a Fellowship from the Logistics and Distribution Institute, University of Louisville, to C. Rojas.

References

[1] N. Durak and O. Nasraoui, "Feature Exploration for Mining Coronal Loops from Solar Images," in *IEEE International Conf. on Tools with Artificial Intelligence (ICTAI)'08*, 2008, to appear.

[2] E. Leon, O. Nasraoui, and J. Gómez, "Network intrusion detection using genetic clustering," in *Genetic and Evolutionary Computation Conference (GECCO 2004)*.

[3] O. Nasraoui and R. Krishnapuram, "A novel approach to unsupervised robust clustering using genetic niching," *Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on*, vol. 1, pp. 170–175 vol.1, May 2000.

[4] O. Nasraoui and R. Krishnapuram, "One step evolutionary mining of context sensitive associations and web navigation patterns," in *SIAM Conf. on Data Mining (SDM 2002)*.

[5] O. Nasraoui, M. Soliman, E. Saka, A. Badia, and R. Germain, "A web usage mining framework for mining evolving user profiles in dynamic web sites," *IEEE Trans. Knowledge Data Engineering (TKDE)*, vol. 20, no. 2, pp. 202–215, 2008.

[6] E. Leon, O. Nasraoui, and J. Gomez, "ECSAGO: Evolutionary Clustering with Self Adaptive Genetic Operators," in *IEEE Conf. Evolutionary Computation (CEC)*, 2006, pp. 1768–1775.

[7] O. Nasraoui, C. Cardona, C. Rojas, and F. A. González, "TECNO-STREAMS: Tracking Evolving Clusters in Noisy Data Streams with a Scalable Immune System Learning Model," in *IEEE International Conf. on Data Mining (ICDM'03)*.

[8] O. Nasraoui and C. Rojas, "Robust clustering for tracking noisy evolving data streams." in *SIAM Conf. on Data Mining (SDM 2006)*.

[9] C. Rojas and O. Nasraoui, "Summarizing Evolving Data Streams using Dynamic Prefix Trees," in *WI' 07*, 2007.

[10] O. Nasraoui, Z. Zhang, and E. Saka, "Web Recommender System Implementations in Multiple Flavors: Fast and (Care) Free for All," in *SIGIR Open Source Information Retrieval workshop*, 2006, pp. 46–53.

[11] Z. Zhang and O. Nasraoui, "Efficient Web Recommendations Based on Markov Clickstream Models and Implicit Search," in *IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, 2007.

[12] Z. Zhang, C. Rojas, O. Nasraoui, and H. Frigui, "SHOW AND TELL: A Seamlessly Integrated Tool For Searching with Image Content And Text," in *ACM-SIGIR Open Source Information Retrieval workshop*, 2006.

Toward Multidisciplinary Collaboration in the CIML Virtual Community

Jacek M. Zurada, Janusz Wojtusiak, Maciej A. Mazurowski,
Devendra Mehta, Khalid Moidu, Steve Margolis

***Abstract*—The importance of virtual scientific communities is constantly growing, as they provide opportunity for collaboration between members located around the world. The Computational Intelligence and Machine Learning (CIML) Virtual Community aims at providing resources to researchers, students, and general public interested in the area. This paper describes how the CIML Virtual Community can support collaboration between CIML and researchers in the healthcare profession. It also describes how it can be generalized to support other disciplines interested in applying CIML methods.**

***Index Terms*—Collaboration, Computational Intelligence, Machine Learning, Virtual Community**

I. Introduction

VIRTUAL communities play an important role in modern science. They allow community members to quickly exchange information across the globe. A *virtual scientific community* is a group of people, often researchers and students, who share multiple resources related to the scientific field, and whose main medium of communication is the Internet. Communication between members often requires creating web portals used to host the resources. While sharing resources is crucial, it is not sufficient for the existence of a virtual community. For example, many web portals that provide community resources, e.g. lists of publications, are not virtual communities. One example of such a portal is DBLP [1], which is one of the most popular lists of publications in Computer Science. Because there is no actual group of members that collaborate through DBLP, it cannot be considered a virtual community. Probably the best organized contemporary virtual communities are those oriented towards specific topics in medicine, bioinformatics, and related areas. This is due to the requirement (in those fields) that all material must be submitted to a well established repository in order to be considered for publication. An example of such a community is the Biomedical Informatics Research Network (BIRN), which provides access to data, tools, and collaborative infrastructure [3].

Wikipedia [2] is also a great source of information in many domains. It follows the ideas of Web 2.0, in which users create content for the web. This content is, however, sometimes not well organized and incomplete. Although most articles are intended for the general public, some of them are very technical and can be understood only by experts. The advantage of Wikipedia is that each article can be modified by multiple authors and therefore reflects their diverse and expansive body of knowledge. The downside of using Wikipedia is that the information is often unverified and expresses personal opinions of the authors which may not necessarily reflect those of the scientific community. In contrast, papers published in scientific journals most often require strict review process which significantly increases the reliability of the published material.

Computational Intelligence and Machine Learning (CIML) is a rapidly growing discipline. Despite the relatively mature state of virtual communities, the idea of building a global CIML community is unfortunately very new. As a result, the current status of virtual cooperation within CIML is worse than in many other domains. Although there have been several attempts to create collaboration websites, data and software repositories, and actual virtual communities, these efforts haven't been coordinated, and respond only to a few specific aspects of CIML community needs as a whole. Among the most noticeable efforts in building CIML virtual communities are the PASCAL and PASCAL2 networks [2] which are European initiatives that support collaboration and research in cognitive systems. Particular areas of interest of the network within CIML include machine learning, pattern analysis, machine vision, and natural language processing. Despite the fact that the networks' website is rich in content (e.g. publications, video lectures, competitions) many items are available only to its members.

Most CIML resources are distributed over countless websites maintained by single researchers, groups, laboratories, and departments. These websites are usually focused on specific topics of interest and leave little room for

This work was supported in part by the National Science Foundation Grant CBET 0742487. The findings and opinions expressed here are those of the authors, and do not necessarily reflect those of the sponsoring organization.

J.M. Zurada is with the University of Louisville, Louisville, KY 40292 USA (phone: 1 502-852-6314; 1 502-852-3940;
e-mail: jacek.zurada@louisville.edu).

J. Wojtusiak is with the George Mason University, Fairfax, VA 22030, USA (phone: 1 703-993-4148; e-mail: jwojtusi@gmu.edu)

M.A. Mazurowski is with the University of Louisville, Louisville, KY 40292 USA (phone: 1 502-852-3165; fax: 1 502-852-3140;
e-mail: maciej.mazurowski@louisville.edu).

K. Moidu is with the George Mason University, Fairfax, VA 22030, USA and Orlando Health, Orlando, FL 32806 (phone: 1 703-993-9734;
e-mail: kmoidu@gmu.edu)

D. Mehta is with Orlando Health, Orlando, FL 32806
(e-mail: Devendra.Mehta@OrlandoHealth.com)

S.S. Margolis is with Orlando Health, Orlando, FL 32806
(e-mail: smargoli@OrlandoHealth.com)

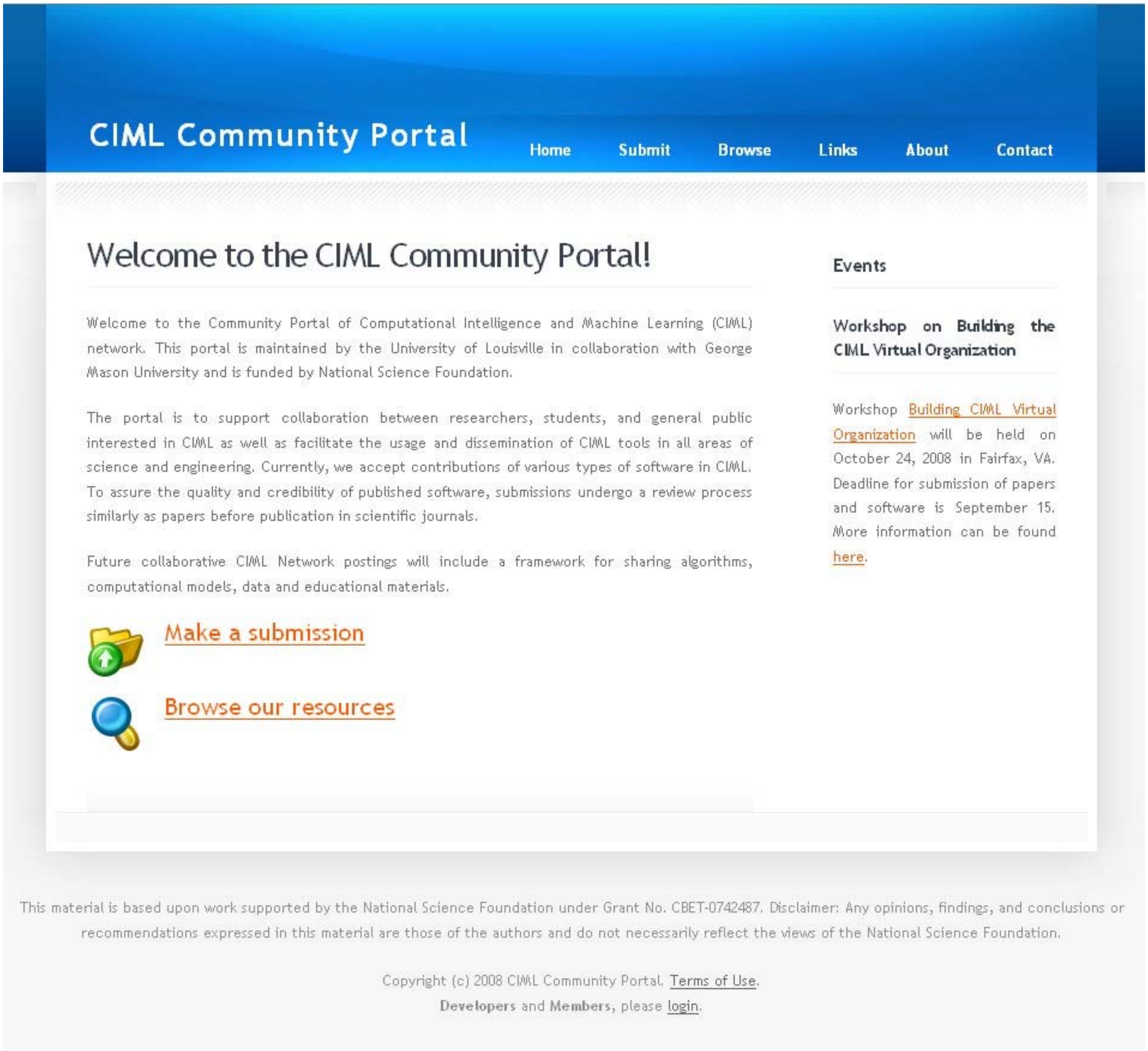
any comprehensive or broad view of the field. In addition, most of these sites also lack any objective content evaluation. Probably the most well known websites in CIML is one which hosts implementation of some standard machine learning algorithms in Java™ within the Weka system [5]. The software is available from the University of Waikato website [6]. Another example of a very popular site is UCI machine learning repository with collection of benchmark data [7].

The above typically concentrate on a single aspect of collaboration, while a more global look at a CIML virtual community would provide not only access to its particular components or functions (such as data sharing or networking). It would also create an interconnection between the components making such a community more integrated and better informed. In this paper, we present our initial efforts towards implementing CIML virtual community, and present how it may lead to multidisciplinary collaboration across disciplines [8].

II. Computational Intelligence and Machine Learning Virtual Community

A. Role of the CIML Virtual Community

As the fields of computational intelligence and machine learning mature, there is a growing need to provide researchers with the ability to exchange information, share resources, discuss problems and new directions, and learn about others' work. In the past, scientific journals were the most important medium of communication between researchers. In the rapidly changing and very dynamic field of CIML, this form of communication is simply too slow for everyday exchange of information. Very quick review processes still take months. In many cases, particularly for high quality journals, it may take two or three years between the original submission and the actual publication. Professional conferences provide the opportunity to meet other researchers as well as present and discuss results. With shorter



Welcome to the CIML Community Portal!

Welcome to the Community Portal of Computational Intelligence and Machine Learning (CIML) network. This portal is maintained by the University of Louisville in collaboration with George Mason University and is funded by National Science Foundation.

The portal is to support collaboration between researchers, students, and general public interested in CIML as well as facilitate the usage and dissemination of CIML tools in all areas of science and engineering. Currently, we accept contributions of various types of software in CIML. To assure the quality and credibility of published software, submissions undergo a review process similarly as papers before publication in scientific journals.

Future collaborative CIML Network postings will include a framework for sharing algorithms, computational models, data and educational materials.

Make a submission

Browse our resources

Events

Workshop on Building the CIML Virtual Organization

Workshop Building CIML Virtual Organization will be held on October 24, 2008 in Fairfax, VA. Deadline for submission of papers and software is September 15. More information can be found here.

This material is based upon work supported by the National Science Foundation under Grant No. CBET-0742487. Disclaimer: Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Copyright (c) 2008 CIML Community Portal. Terms of Use.
Developers and Members, please login.

Figure 1: The main page of CIML virtual community portal.

review processes, often in the order of a few months, these conferences allow more rapid communication and discussion of research results. Despite these benefits however, high travel costs often prevent potential attendees, in many cases students and distant researchers, from attending.

The aforementioned limitations, along with others, of traditional scientific communication inspired us to create a CIML virtual community. The goal of the community is to create a place where scientists, students, and the general public can work together despite any of their geographic limitations. The next section briefly describes our initial efforts to create such a community and presents its current status.

B. Current Status

The CIML virtual community board membership currently consists of 25 people, which are well established researchers in the area. Thirteen members are form the United States and twelve are from other countries. These members help in building the community and its various components.

Currently the main initiative is building The Computational Intelligence and Machine Learning community portal. The portal will serve as a medium to exchange data and software, as a professional networking platform, and as a source for help in obtaining educational materials. A screenshot of the main page of the portal is presented in Figure 1. The portal's core development team consists of two professors from the University of Louisville, one professor from George Mason University, and three students from the University of Louisville. Despite its youth, the community is already accepting submissions of CIML software. The software goes through a review process similar to that used by scientific journals, and upon its acceptance is published on the portal.

III. Collaboration with The Medical Community

A. Goals

Healthcare is an area with diverse problems, types of datasets, and study objectives. Researchers in the medical and general healthcare domains frequently use popular statistical methods, but are not familiar with the wide range of methods and tools available in CIML. Moreover, current physicians in training are asked to do research. Also the challenge of search for solutions drives the senior physicians to conduct research. The power of a collaborative would empower them to compare the data sets and results they have with others researching similar issues around the core problem.

National collaborations in Oncology, for example, have shown the importance of networks. These collaborations are, however, only possible in a few heavily funded fields. Partly owing to the success of such collaborations, areas like clinical research increasingly depend on multicentered studies. Networks that allow collaborations in similar or overlapping areas would be particularly advantageous and allow large collaborations to develop, especially in areas not traditionally well funded. Access to the CIML virtual community would facilitate this process tremendously by connecting individual physicians, assigned representatives, and geographically distributed experts. Even down to individual practices, an environment of ongoing case review and outcome analysis leads to an environment of data driven changes, which is advocated by multiple organizations including the Institute of Medicine. These measures are often taken on a small scale, yet improvements have lead to their having substantial impacts. Application of advanced analytical tool may still not necessarily occur, however, as resources or research staff may be lacking knowledge or resources. Useful presentations and publications of this type may not be in the mainstream journals of the discipline. Collaboration within the CIML virtual community could lead to larger scale review of similar cases or scenarios, a more robust meta-analysis of data, which in turn could lead to more defined strategies to improve outcomes. Consequences of such a large, open, and more diverse community would include availability of immediate feedback from peers, as well as immediate dissemination of successful strategies. Therefore, research results obtained by using CIML in healthcare have the potential for a very high impact.

Concluding from the discussion above, healthcare is a domain in which CIML tools and virtual collaboration could yield significant results.

B. Examples of Possible Applications

The first example concerns managing nosocomial infections, an important step in reducing overall morbidity and mortality rates. By the careful logging of critical variables, measures undertaken, and incidence of such infections, comparisons in specific patient populations, institutions, or approaches would be possible on an ongoing manner and in real time. Moreover, measures associated with lower rates can be readily identified. Sharing of such data would allow powerful analysis on large sample sizes, and identify potential risk factors that would otherwise not be recognized. While identification of outstanding institutions and approaches would occur readily, subtle gains as well as major breakthroughs from a myriad of approaches could also be more quickly identified and adopted by others. Therefore, application of CIML tools available within the virtual community would allow previously impossible analysis of nosocomial infections data.

Another example concerns reflux disease and asthma. These two diseases commonly coexist. Recently, the ability to identify acidic ($pH<4$), weakly acidic ($pH<4\leq7$) or alkali ($pH>7$) reflux and level up to the upper esophagus has been possible using combined impedance and pH probe studies. Furthermore, the ability to detect microaspiration has improved with the recent advent of airway pepsin as a biomarker of gastric aspiration. The potential role of pepsin as a cause of irritation, inflammation needs to be evaluated along with other gastric fluid constituents or properties including acidity, hypotonicity, and even bile or microbes. Using cough as a defined event in asthma, we studied 117 children with asthma to assess pH of refluxate immediately prior to a cough.

In a subset who had bronchoscopy, we assessed the prevalence of pepsin positivity in the airway. There were 27 attributes in the core study. Some attributes related to severity of asthma, based on need of medications to control asthma, and measures such as spirometry. This is a composite score because with young children, spirometry is not available, and their classification may be less robust. Additional attributes emanate from impedance pH probe studies. These are studies carried out over 18 to 24 hours, with or without acid suppression. We look at characteristics of reflux, as well as correlation with any symptoms. These results are collected by the physician after manual review or electronically from the software. Specific elements are entered into a database. Several attributes help relate characteristics of reflux and subsequent symptoms. Normative scales are used to assess categorical interpretation of the studies as normal or abnormal by the physician. Other attributes collected include symptoms that suggest gastrointestinal reflux, evaluation results of any tests for reflux, and pulmonary symptoms over time. These are collected at bedside and entered in an electronic medical record. The relevant information is then parsed and transferred to a database. In a subset who had undergone bronchoscopy, description of the airway in terms of inflammation, narrowing or other lesions, as well as data from airway washings for pepsin activity were collected. This data comes from studies performed by pulmonologists at a separate time point to impedance study, and are collected via chart review (these are performed at separate institutions with varied electronic record systems, hence the need for a manual review and entry). For part of the data collection, entry into paper forms was performed, which can later be scanned into an electronic format.

By using a CIML approach, predictors and determinants can be derived from additional historical information. In prospective phases where interventions are planned, the multiple factors affecting asthma would be possible. These include psychosocial, seasonality, allergy, and a clinical course over a lengthy timeline that would allow small changes to be detected. Such an approach could also elucidate the potential role of reflux and microaspiration on wheezing infants and toddlers, interstitial lung disease, and other chronic lung diseases. Likewise, potentially significant results can be obtained in patients with poorly protected airways, including in national and pediatric intensive care unit populations. Finally, the role of pepsin in other etraesophageal manifestations of reflux, such as laryngitis, otitis media, and perhaps sinuses could be elucidated. As a result, predictive models could be created to offer point of care decision support.

C. Methodology

Healthcare researchers interested in applying CIML tools to solve problems in their field of study can utilize tools, methods, and expertise available at the community's portal. In general, we can assume that they have some data to be analyzed. Such data can come in many different forms: structured (from existing databases), text, images, time series (e.g. EEG), and others. There is often a background knowledge associated with the particular domain in the form of prexisting models, rules, ontologies, dictionaries, hierarchies, etc.

Once the data and possible background knowledge is available, the user needs to define the problem of interest. This is one of the most difficult parts of the process as it requires mapping of sometimes very vague problem description into the CIML methodology. For example, the statement "I want to find out what are possible reasons for chest pain among my patients" is too imprecise to be directly addressed by CIML tools. It needs to be translated into the right problem such as classification, clustering, or optimization. This process should be supported by a series of questions that correspond to entries illustrated in the Figure 2**.**

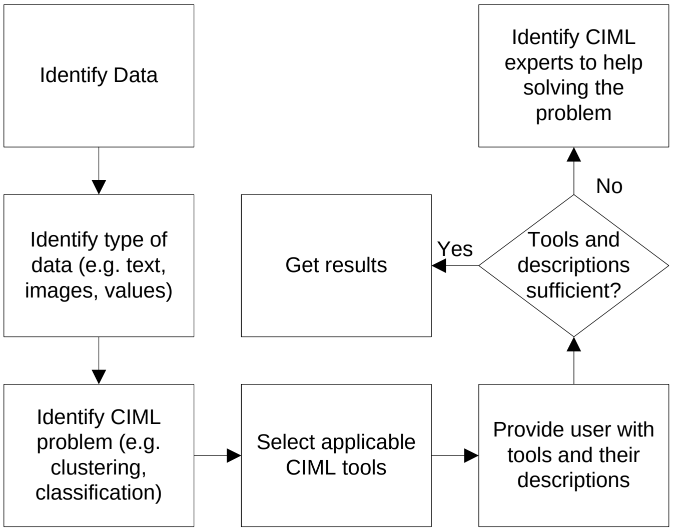

Identify Data
Identify type of data (e.g. text, images, values)
Identify CIML problem (e.g. clustering, classification)
Select applicable CIML tools
Provide user with tools and their descriptions
Tools and descriptions sufficient?
Yes
Get results
No
Identify CIML experts to help solving the problem

Figure 2: Steps in selecting CIML tools.

Interaction with the CIML community portal should lead users to:

- Selection of *relevant* CIML tools
- Access to *relevant* tutorials and articles
- Contact information for CIML community members whose area of expertise is *relevant* to the considered problem.

The key word in the above is "relevant" because the portal should guide users through its resources. The healthcare researchers may then access and apply the selected tools to obtain actual results. The complete process is illustrated in Figure 3. Because a significant part of the data in medical records is stored in the form of text (e.g. diagnoses information), we emphasize it in the below diagram. Using text recognition tools it can be transformed into structured data that's viable for further analysis.

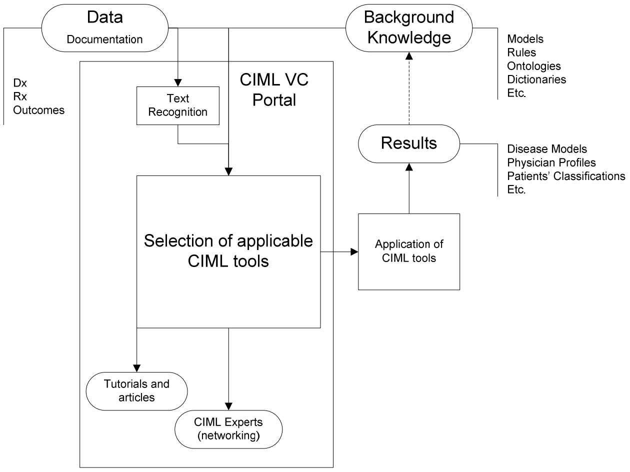


Figure 3: Diagram of a general methodology for choosing applicable CIML tools in medical domain.

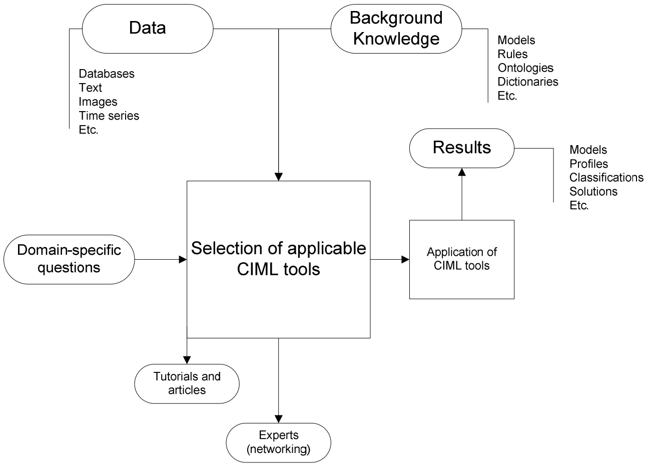


Figure 4: Diagram of a general methodology for choosing applicable CIML tools across disciplines.

IV. Collaboration with Other Disciplines

While the medical domain provides an important area of application of CIML tools, there is also great potential for the CIML community to impact other disciplines. The fact that scientists from various disciplines may be interested in applying methods developed within CIML does not require much justification. It is sufficient to look at the content of conferences and journals in both CIML and other disciplines to discover a wide range of current and potential applications.

The methodology for collaboration between the CIML community and health care researchers presented in the previous section can be generalized to other disciplines as depicted in Figure 4. The key part is generalization of the most important modules, namely selection of applicable CIML tools and experts. Creation of such a module to work across disciplines will require creating an adaptive methodology that will be able to automatically incorporate new community members and software tools submitted to the community.

The methodology for enabling collaboration between CIML and any other discipline is the same. The challenge is to adapt terminology used by the portal to what is easily understood by users at different levels of CIML knowledge. Researchers in different fields, or even in the same field, tend to use different terms when talking about the same things.

A potential solution to the above problem is to create ontology of terms used in computational intelligence and machine learning, and then map it onto terms used in different disciplines. Thus, experts from different disciplines would be asked questions using their own domain-specific language.

V. Conclusion

Benefits from multidisciplinary collaboration within computational intelligence and machine learning are numerous. Researchers working with multiple disciplines clearly benefit from the access to state-of-the-art CIML tools, their descriptions, articles, and researchers. On the other hand CIML researchers benefit by having the possibility to drive research by real world problems. It is our intention to initiate and support multidisciplinary collaboration whose central part is development and use of CIML tools and methodologies.

Acknowledgement

The authors thank Jordan Malof for his valuable comments that helped to improve this paper.

References

[1] M. Ley, DBLP website, http://www.informatik.uni-trier.de/~ley/db/
[2] Wikipedia website: www.wikipedia.org
[3] The Biomedical Informatics Research Network (BIRN) http://www.nbirn.net
[4] The PASCAL network website: www.pascal-network.org
[5] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Series, 2005.
[6] www.cs.waikato.ac.nz/~ml/
[7] The Machine Learning Repository, University of California, Irvine website: http://archive.ics.uci.edu/ml/
[8] J. M. Zurada, J. Wojtusiak, F. Chowdhury, J. E. Gentle, C. J. Jeannot, and M. A. Mazurowski, *Computational intelligence virtual community: framework and implementation Issues*, International Joint Conference on Neural Networks (IJCNN 2008), June 1-6, 2008, Hong Kong, pp. 3152-3156.

Workflow considerations in the emerging CI-ML virtual organization

Chris Boyle[1], Artur Abdullin[1], Rammohan Ragade[1], Maciej A. Mazurowski[1], Janusz Wojtusiak[2], Jacek M. Zurada[1]

Abstract—**In a virtual organization, the interaction of its members for any purpose generates a sequence of activities referred to as a workflow. This paper seeks to identify the workflows needed for the Computational Intelligence and Machine Learning Virtual Organization. The underlying architecture of the repository should support these workflows in a smooth and efficient manner.**

Index Terms—**virtual organization, computational intelligence, machine learnining**

I. Introduction

In a companion paper [1], a vision for a collaborative virtual organization (VO) for the Computational Intelligence and Machine Learning (CI-ML) community was introduced. The purpose of this VO is to provide a portal for peer reviewed software, algorithms, tools, data, and models. Members of the VO are allowed to review submitted content from outside developers and determine if it is acceptable for redistribution in the organization's repositories. Outside learners are allowed to access this content for educational purposes. In a VO, the interactions of the different types of users that utilize computer resources generate a sequence of pre-defined activities known as workflows.

Workflows have been widely discussed in the development of virtual organizations. The work of Khoshafian and Buckiewicz [2], describes their consideration of workflows in collaborative computing. In the area of machine learning and distributed AI, Singh and Huhns [3] laid out challenges for cooperative computing. With agents they handled issues of workflows in the computing task. More recently, we see examples of workflows in cooperative, collaborative computing in bioinformatics, computer security applications, text mining and other large scale applications. Java based agent technologies have enabled smoother software engineering tasks for the infrastructure of virtual organizations [4]. Even assessment processes, which has many factors in common with reviewing, has seen the need for workflows in computer supported collaborative tasks [5]. Dennis Gannon and his associates identify workflows for e-science [6] which is clearly an aspect in CI-ML. At the January 2008 workshop titled "Building Effective Virtual Organizations" hosted by the National Science Foundation, Marru, Gannon and Plale [7] gave a presentation to identify the role of workflows in e-science.

The workflows for the CI-ML VO are categorized as: sharing data, sharing software resources, sharing computational resources, education and networking. These workflow categories, while common for many other VO's, gain some unique qualities when applied to CI-ML. These qualities were considered at the NSF workshop to identify many established and emerging VOs and their characteristics. A common characteristic is that each VO has a domain of application, in which most of the members of the VO are familiar. The infrastructure being established by the VO should nurture and sustain the regular and evolving activities of the VO. Therefore, the lifecycle of the many workflows in a VO should be considered in detail.

The life cycle of a workflow identifies the resources and interactions between a sequence of logical activities. The computing resources set up for each interaction within the VO is dependent on how these interactions are identified. These interactions are easily defined as a use case to describe how a type of system user is able to interface with the rest of the system. A use case is usually defined with an actor, which can be considered as a user or subsystem, and a definition of acceptable behaviors across system boundaries. The formal definition of all possible use cases for a particular actor is important due to implications on system security and information assurance. The collection of use cases define the functional requirements for the VO and define the services it provides. Given the functional requirements and its accompanied workflows a software architecture can be created that is modular and scalable.

The importance of creating workflows has major implications for the future growth of a VO. Considerations for the CI-ML VO were made to allow its expansion given the addition of new functional requirements and use cases. This forced the software achitecture to be well defined to accomodate these changes while having minimal impact on the system as a whole. This paper will discuss these workflows, their impact on the software architecture, and security issues to accomodate the use case requirements.

II. Workflows envisaged

The goal of the CI-ML VO is to create a community that will eventually become a place where researchers, students, and the general public come to seek information about computational intelligence, machine learning, and related topics. The section below discusses the scenarios in which workflows are needed. These scenarios are then considered for the workflow

[1]University of Louisville, Louisville, KY
[2]George Mason University, Fairfax, VA

categories: sharing data, sharing software resources, sharing computational resources, education and networking.

A. Scenarios

There are a wide variety of evolving scenarios for the VO. The main focus will be on the sharing and collaborative efforts among users of the system.

Most researchers in CI-ML, as in any other research community will review scholarly publications in monographs, e-journals, journals and conferences. If the content for review is not familiar, researchers seek other work that is closely related. This activity results in a variety of tasks associated with reviewing and referring. If it is a new method or algorithm, the researcher may even try out the method or tool on a familiar problem, before taking further action. If there is an associated software package, depending on the complexities of the package further workflows would ensue.

A researcher may seek to collaborate with other reviewers to assess a contribution. In this case, the researcher must establish the compatibility of the datasets with the proposed tools and domains. If needed, datasets will be transformed to enable appropriate tools to be used. Subsequently, comparative results require normalization for fairness in evaluation. Benchmarks often provide frameworks for this task.

When developing a new method, use of known software toolsets in the composition of proposed methods brings in additional considerations of software versions, licences, permissions, languages used and operating system environments. Through dialog in the CI-ML community, the researcher can obtain appropriate components for the proposed new method. This dialog process generates yet another appropriate workflow that should be facilitated in the emerging VO for the CI-ML community.

Established VO's such as those in genomics have already standardized processes for access to shared community resources. However, due to serious considerations of intellectual property (IP) and concern for prior disclosure, even queries have to be conducted in a secure non-disclosure proof environment. These concerns will also carry over to the emerging CI-ML VO, atleast where software tools are concerned.

Based on these scenarios, we have identified basic top level categorization of broad use cases for a review process. These are from a member submitter's perspective and from an administrative perspective.

B. Categories

The five workflow categories of the VO are: sharing data, sharing software resources, sharing computational resources, education, and networking. Listed below are the various activities and processes for each category. Each element in a category defines a workflow along with the interactions between elements. The permutations and combinations of the interactions are too numerous to be listed. This highlights the importance of defining an element as independently as possible. For example, proposing a new algorithm may require the use of many other processes but these do not need to be considered.

1) Sharing Data: The sharing data category encompasses the requirement that the VO handle algorithms, data, and models [1]. An outside developer has the ability to make a proposal for one of these types for review with a VO member. For example, when a new algorithm is proposed, a set of testing procedures must be defined for verification.

These are the activities and processes for the category:

- Proposing a new algorithm, data, or model. The proposal of a new elements initiates the review and testing for the user's submission.
- Reviewing/Testing
  - Benchmarking
  - Comparison with other methods/data
  - Scaling for real problems
  - Storing associated data
  - Accessing relevant data
  - Provide collaborative review of results
  - Data format transfer issues

2) Sharing Software: The sharing software category encompasses the requirement that the VO handle software and tools [1]. An outside developer can propose software in either its source code or executable form for review.

These are the activities and processes for the category:

- Finding quality software
- Making modules work together
- Check for language syntax/formatting/documentation issues
- Evaluate appropriateness for the CI-ML domain

3) Sharing Computational Resources: Computational resources for the system must be protected and managed correctly. Given that computational resources can be released, workflows must be defined for control.

- Working towards an open Framework development for large scale application
- Allowing diverse distributed resources/grid approach, etc.

4) Education: A future goal of the VO is to allow the collaborative sharing of educational content.

- Activities for generating, posting, updating Tutorials
- Creation of Frequently Asked Question (FAQ) sections
- Utilizing newer modes, such as access of multimedia content

5) Networking: Another future goal is to allow members, developers, and learners to collaborate and network together.

- Identifying focused CIML interests among members
- Providing channels of communication among members

The above workflows are substantially complex and the architecture should accommodate the workflows productively. As we move towards identifying appropriate architecture, issues of security and priorities for work flows take center stage. The next two sections briefly address our approaches towards these tasks.

III. Current VO Architecture

The basis for the current architecture for the CIMLVO is from Martin Folwer's book for Patterns of Enterprise Architectures [8]. Three principal layers are identified for presentation, services, and data access. These are created in order to divide responsibilities into smaller more manageable entities. For example, the required workflows for the presentation layer have a differing focus than the data access layer. This separation of concerns promotes modularity throughout the system.

The presentation layer has the responsibility for providing a user interface for the application. Given that this is a virtual organization, the presentation layer will be accessible through the World Wide Web and the services that the VO provides are defined through this gateway. The responsibilities for its workflows are focused with interacting directly with the user. For example, when the user types in an invalid input, a process is defined to display error messages. The presentation layer has a direct dependency to the service layer.

The service layer provides all of the application logic or business logic. The previous section has defined the major workflows that are implemented in this layer. If the presentation layer needs to access a table in a database, the service layer will provide the functionality through its dependency with the data access layer. Most workflows originate for this layer in the architecture because it provides all of the foundational logic for the entire system. This layer is divided into smaller managers that further modularize functionalities. For example, the submission manager allows submission of new content.

The data access layer gives access to any persistent storage that is needed. For example, if direct access is needed for a database. The VO uses a database for persistent storage and has an object-relational mapping (ORM) for interacting with the database. This provides a nice abstraction layer so that specific knowledge of the storage method is not needed.

The VO is coded in Java® and hosted inside a Java Application Server. The application server is responsible for the handling of requests and instantiation of workflows. The server generates a finite number of threads for new requests and responds accordingly. The number of concurrent threads is variable and can be changed to fit the needs of the VO. It is important to maintain this property by monitoring the needs of the system and providing the necessary system improvements.

IV. SECURITY ISSUES

The workflow architecture for the VO should contain explicit requirements for the management and enforcement of security and privacy of sensitive information. The workflows must be implemented in a way to guarantee that the functionality of the network does not override any security concerns. More specifically, workflows should enforce three principals for the VO: integrity, authorization, and availability [9]. These properties become exceedingly more important because the VO typically handles sensitive data and protected functionalities.

A. Integrity

The VO secures its workflow tasks by assigning each user a set of roles that classify the user's responsibilities. The layered structure of the software architecture helps by dividing the Service/Workflow layer from the Data Access layer. The Service/Workflow layer maintains data integrity by correctly defining workflows and maintaining user responsibilities.

A role based security model provides many benefits because it reduces complexity and can easily be incorporated with existing technologies. Most importantly it allows the system to run by the principal of least privilege, where the minimal set of privileges is allowed to execute [10].

B. Authorization

Each role within the VO has permission to perform a set of defined responsibilities on the data. The originating paper describing the VO [1], discusses a layered model for access of developer's submission. The figure below shows the layered nature of the submission network.

The most basic VO roles are defined as: Learner/User, Developer, and Member. The responsibility of each type of basic VO user is a subset of the responsibilities for a higher layer. The basic roles can be defined as:

$$U \subseteq D \subseteq M$$

where

$$\begin{aligned} U &= Responsibilities\, of\, Users/Learners \\ D &= Responsibilities\, of\, Developers \\ M &= Responsibilities\, of\, Members \end{aligned}$$

A more complex set of roles can be assigned to a VO user to handle administration duties. For example, an editor role exists that has the ability to assign a review to an appropriate member. There are many disjoint administration roles than contain smaller responsibilities. This is done to provide flexible management of the VO, which has the ability to grow to a very large size. It is vital to plan for future growth by allowing micromanagement of the VO. All administrative users have the same responsibilities as developers but are not required to be members.

Using this methodology, integrity is maintained by allowing data manipulation to occur by responsible VO users. Each role within the VO interacts with the Service/Workflow layer of the VO software architecture. The Service layer is responsible for maintaining the authorization and data integrity. It then interacts with the Data Access layer for data access and modification.

C. Availability

An important part in the development of a workflow is that required resources are available upon request. The workflow must be able to execute within a reasonable amount of time so that the VO can function reliably. A secure workflow that provides the availability property is defined as: For every task there must be at least one agent who is able to execute the task [9].

The availability property is also maintained by proper testing of workflow logic. The distributed and concurrent nature of the VO requires analysis of the workflow to avoid negative symptoms like dead-lock. Thorough unit, integration, and system testing is required for verification that all three principals are maintained.

V. Conclusions

It is well known that in any networking activity, complexities of workflows and tasks increase as the number of interacting entities increase. For an emerging VO, it is essential to maintain coherence of the activities to its stated goals and purpose. The community must evolve a mechanism to make adjustments and adaptations. Portals are established for this purpose, which allow filtering of activities. The modular and layered approach will allow new workflows to be integrated into the architecture, with careful consideration. As long as the main focus is to nurture the VO community, these additions and adaptations can be accommodated.

Acknowledgements

This work was supported in part by the National Science Foundation Grant CBET 0742487. The findings and opinions expressed here are those of the authors, and do not necessarily reflect those of the sponsoring organization.

Chris Boyle's contribution supported by the University of Louisville Intramural Research Incentive Grant. Also, we would like to thank Jordan Malof on his editing suggestions for this paper.

References

[1] J. M. Zurada, M. A. Mazurowski, J. Wojtusiak, R. Ragade, and J. Gentle, "Building virtual community in computational intelligence and machine learning," *Computational Intelligence Magazine*, 2008.

[2] S. Khoshafian and M. Buckiewicz, *Introduction to Groupware, Workflow, and Workgroup Computing*. John Wiley and Sons, 1995.

[3] M. P. Singh and M. N. Huhns, "Challenges for machine learning in cooperative information systems," in *Distributed artificial intelligence meets machine learning, Lecture Notes in Artificial in Artificial Intelligence*, pp. 11–24, Springer-Verlag, 1997.

[4] M. L. Griss, "Software engineering with java agent components," in *Borcon 2003*.

[5] K. R. Lai and C. H. Lan, "Modeling peer assessment as agent negotiation in coputer supported collaborative learning," ICMAS'96 Workshop LIOME, 2006.

[6] I. Taylor, E. Deelman, D. Gannon, and M. Shields, *Workflows for e-Science: Scientific Workflows for Grids*. Secaucus, NJ, USA: Springer-Verlag, 2006.

[7] S. Marru, D. Gannon, and B. Plale, "Lead workflow use cases," [powerpoint presentation at the NSF workshop on Emerging Virtual Organizations, Washington DC, Jan 15 -17, 2008.].

[8] M. Fowler, *Patterns of Enterprise Application Architecture*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.

[9] P. C. K. Hung and K. Karlapalem, "A secure workflow model," in *ACSW Frontiers '03: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003*, (Darlinghurst, Australia, Australia), pp. 33–41, Australian Computer Society, Inc., 2003.

[10] J. Whittaker, "Why secure applications are difficult to write," *IEEE Security and Privacy*, vol. 1, no. 2, pp. 81–83, 2003.

Virtual Communities – Large-Scale Human-Computer Networks

Robert Kozma and Marko Puljic

Abstract – **Virtual communities are viewed as large-scale complex systems operating between populations of humans and computers. We analyze these systems using neuropercolation theory, thus extending previous results based on studying spatio-temporal neurodynamics in brains. Phase transitions in spatially extended networks play critical role in robust functioning. We argue that optimally designed human-computer networks must operate near criticality, thus generating the desired fast and reliable operation at large-scales.**

SUMMARY

TODAY we can witness a paradigm shift in science and technology due to the enormous complexity of the problems researchers attempt to rigorously analyze using powerful digital computers. Virtual communities were nonexistent just a few years ago, but they have explosively expanded in recent years. Virtual communities are examples of large-scale complex systems with emergent behaviors between humans and computers. The forefront of research in this field explores human cognitive functions, both in individuals and in populations of individuals [8, 11, 14].

These are highly nonlinear and nonstationary systems and traditional mathematical tools have limited success in analyzing them. Methods of discrete mathematics, combinatorics, statistics, and the theory of random graphs and networks are especially useful in describing such complex phenomena. Scale-free random graphs and networks pose very difficult mathematical questions. There has been rather little rigorous mathematical work in this area. Progress has been made with scale-free random graph models of large-scale real-world networks. Significant novel results concern inhomogeneous random graphs and their general scaling properties [3]. There is a yet inadequate modeling of dynamic behavior of large-scale graphs influenced by complex topology. Novel mathematical tools are required to rigorously describe these phenomena. Initial steps to this direction are indicated in [4].

This work is based on the studies of brains as large networks in the framework of neuropercolation theory, and extends these studies to networks formed by brains and computers [9,10,12]. There is a critical link between mesoscopic brain activities manifested in the form of wave packets, and macroscopic activities involving the entire hemisphere, measured by brain imaging using fMRI, PET, SPECT, EEG and EMG. The waves often have large-scale, highly textured spatial patterns of cortical activity. Synchronization manifests continuous distributions of activity [8, 11, 13] in cortical neuropil that modulate firings of selected neural networks [5-7].

Neuropercolation theory offers a fresh beginning, in which the discreteness of network connections can be approximated with numerical representations in percolation theory [9, 13]. It is readily adapted to describing microscopic, mesoscopic, and macroscopic levels, and the relations among spatial and temporal variables between levels in phase transitions. Modeling of structural and functional connectivity by neuropercolation theory is well advanced, particularly in modeling the interplay of long connections, inhibitory feedback, and additive noise in the genesis of self-regulated spontaneous activity of large nets of nodes at the mesoscopic level.

The results obtained based on brain studies can be generalized to the description of various networks, including collaborations, school friendships, mobile phone calls, word associations, protein interactions. The explored graphs demonstrate that the web of modules has highly non-trivial correlations and specific scaling properties [1, 15]. Statistical features of populations have been analyzed and clustering properties described [2, 12] leading towards a much needed step to uncover the modular structure of complex systems. Efficient techniques are introduced to explore overlapping communities on a large scale.

Due to the highly interdisciplinary nature of virtual communities and large-scale human-computer networks, major advances must be made in identifying the common language across disciplines, which include physical, biological, social, and behavioral networks. This workshop can play an important role in that direction.

REFERENCES

[1] Albert R, Barabasi A-L [2002] Statistical mechanics of complex networks. *Rev Mod Physics* 74: 47-97.

[2] Balister P., Bollobas B., Kozma R. [2006] Large deviations for mean field models of probabilistic cellular automata. *Random Structures and Algorithms* 29: 399-415.

[3] Bollobas B, Riordan O [2006] *Percolation.* Cambridge University Press, 2006.

[4] Bollobas B., Kozma, R., Miklos, D. [2008] *Handbook of Large-Scale Complex Networks,* Bolyai Studies in Mathematics, Springer Verlag, Heidelberg.

[5] Freeman, W.J., Burke, B.C., Holmes, M.D. [2003] Aperiodic phase re-setting in scalp EEG of beta-gamma oscillations by state transitions at alpha-theta rates. *Human Brain Mapping*, 19(4):248-272.

This work was supported in part by the U.S. Air Force Office of Scientific Research (AFOSR) and National Research Council (NRC).

The authors are with the Computational Neurodynamics Laboratory, The University of Memphis, 373 Dunn Hall, Memphis, TN 38152, USA (email: rkozma@memphis.edu, neuropercolation@yahoo.com). Robert Kozma is also affiliated with U.S. Air Force Research Laboratory, Sensors Directorate, Hanscom Air Force Base, MA 01731, USA.

[6] Freeman WJ [2006] *Neurodynamics. An Exploration of Mesoscopic Brain Dynamics.* London: Springer. Electronic version: http://sulcus.berkeley.edu/.

[7] Freeman WJ [2007] Proposed cortical "shutter" mechanism in cinematographic perception. In: *Neurodynamics of Cognition and Consciousness,* Perlovsky L, Kozma R (eds.) Heidelberg: Springer Verlag, pp. 11-38.

[8] Kelso JAS [1995] *Dynamic Patterns: The Self Organization of Brain and Behavior.* Cambridge MA: MIT Press.

[9] Kozma R, Puljic M, Balister P, Bollobas B, Freeman WJ. [2005] Phase transitions in the neuropercolation model of neural populations with mixed local and non-local interactions. *Biol. Cybern* 92: 367-379.

[10] Kozma R [2007] Intentional systems: Review of neurodynamics, modeling, and robotics implementations. *Phys of Life Rev* 5(1): 1-21.

[11] Pikovsky A, Rosenblum M, Kurths J [2001] *Synchronization. A Universal Concept in Non-linear Sciences.* Cambridge UK Cambridge UP.

[12] Puljic M, Kozma R [2005] Activation clustering in neural and social networks. *Complexity,* 10(4): 42-50.

[13] Puljic M, Kozma R [2008] Narrow-band Oscillations in Probabilistic Cellular Automata. *Phys. Rev. E.* 78, 026214.

[14] Tognoli, E., Lagarde, J, DeGuzman, G.C., Kelso, J.A.S. [2007] The Phi complex as a neuromarker of human social coordination, *Proc. Nat. Acad. Sci. PNAS,* 104(19), 8190-8195.

[15] Wang XF, Chen G [2002] Synchronization in scale-free dynamical networks: Robustness and fragility. *IEEE Trans Circuits Syst 1*. Fund Theory and Appl. 49: 54-62.

Comparison of Learning Methods: Pitfalls and Challenges

Vladimir Cherkassky and Wuyang Dai

EXTENDED ABSTRACT

The growing importance of discovering regularities in observed data in many applications has led to a number of diverse data-analytic methodologies, such as Pattern Recognition, Machine Learning, Data Mining, Artificial Neural Networks etc. Most algorithms developed in these fields pursue the goal of estimating (learning) predictive models from available data. Such a predictive model is then used for prediction with new (test) inputs. The prediction (or generalization) performance of a model can be objectively evaluated using an independent test set. Over the past 10 years, hundreds or maybe thousands of new learning algorithms have been proposed. Typically, introduction of a new algorithm includes empirical comparisons suggesting that the proposed method is 'better' (in terms of generalization performance) than already existing methods. Remarkably, the process of inventing new algorithms continues at a rapid pace, even though it is logically inconsistent to have hundreds of 'best' algorithms. A closer inspection of empirical comparisons used to justify new learning methods suggests that:

(a) Often the experimental procedure (used for comparisons) is poorly designed (or not described at all).
(b) The authors of a proposed learning method put special effort in tuning its parameters.
(c) Sometimes comparisons fail to differentiate between resampling error (used for model complexity control) and true prediction error.
(d) Comparisons fail to account for the fact that generalization performance always depends on statistical characteristics of the data (such as sample size, amount of noise etc.).

This paper describes issues arising in empirical comparison of learning methods, and illustrates potential pitfalls via simple examples.

Inductive Learning Setting:

Our discussion starts with a brief review of inductive learning setting underlying most learning algorithms. Standard inductive learning [3,4,5,6] attempts to estimate a model or function f which maps an input vector $\mathbf{x} \in \mathbf{X}$ to an output $y \in \mathbf{Y}$. This model is selected from a set of possible models $f(\mathbf{x}, w)$ parameterized by a general set of parameters w. Estimation (or learning) is performed using finite training samples that are identically and independently generated from an unknown probability distribution $P(\mathbf{x}, y)$. The goal is to find the best function *f* such that the expected loss

$$R(w) = \int L(f(\mathbf{x}, w), y) P(x, y) d\mathbf{x} dy$$

is minimized. Here $L(f(\mathbf{x}, w), y)$ denotes a loss function appropriate for a given application (i.e., classification error, squared loss etc.). This standard inductive learning setting implies that:

(a) the model is estimated using *only* finite training set.
(b) Prediction accuracy is estimated using *large* test set.

In practice, test set is finite (at best) or not available (at worst). In the latter case, prediction accuracy is usually estimated using resampling of available training data. As a result, estimated prediction error always depends on selected resampling procedure and pure luck. Remarkably, many empirical comparisons do not give any details of the resampling procedure.

Further, prediction performance of a method is strongly affected by implementation of model selection, i.e. tuning of method's parameters, such as the value of k in k-nearest neighbors, the number of hidden units in MLP networks, regularization parameter C in SVM etc. These parameters are typically selected via resampling. However, different implementations of resampling, i.e. leave-one-out vs 5-fold cross-validation may yield different model complexity. Unfortunately, detailed description of the experimental procedure for model selection is (almost) never reported.

High-Dimensional Data:

Particular issues and challenges arise with application of learning methods to sparse high-dimensional data. Such a data is common in biomedical applications, imaging, text categorization etc. These applications usually favor simple and robust methods such as k-nearest neighbors, linear SVM and linear discriminant analysis. Due to sparseness of such data, clear and detailed description of resampling procedure used for comparisons becomes especially important.

Non-Standard Learning Settings:

Many recent powerful algorithms do not follow standard inductive setting. For example, transduction and semi-supervised learning incorporate the knowledge of x-values of test data into learning. Comparisons involving such non-

Vladimir Cherkassky and Wuyang Dai are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA. (e-mail: cherk001@umn.edu) .

standard learning formulations are especially challenging because:

(a) one has to clearly understand underlying assumptions of these non-standard formulations vs assumptions used in inductive learning.

(b) Such new formulations have more tuning parameters, which makes model selection more difficult. Hence, performance estimates become especially sensitive to resampling procedure used in comparisons.

We discuss in detail several examples of non-standard learning setting where the training data includes additional (group) information. This leads to new learning settings known as Multi-Task Learning [1,2,8] and Learning with Structured Data (aka SVM+) [7], as discussed next.

Suppose that training data can be represented as a union of *t* related groups, i.e. each group $r \in [1,2,..,t]$ contains n_r samples independently and identically generated from a distribution P_r on $\mathbf{X} \times \mathbf{Y}$. Therefore, available data is a union of *t*>1 groups:

$$\{\{X_r, Y_r\}, r = 1,...,t\}, \{X_r, Y_r\} = \{\{\mathbf{x}_{r_1}, y_{r_1}\},...,\{\mathbf{x}_{r_{nr}}, y_{r_{nr}}\}\}$$

and can be though as samples identically and independently generated from the distribution $P = \cup_{r=1,..t} P_r$.

If the group labels of future test samples are not given, the problem is "**L**earning **W**ith **S**tructured **D**ata (LWSD)" formulation [7]. In this formulation, the goal is to find one best mapping function *f* such that the expected loss

$$R(w) = \int L(f(\mathbf{x}, w), y) P(\mathbf{x}, y) d\mathbf{x}dy$$

is minimized. Note that even though the expected loss is in the same form as in the supervised learning setting, the difference is that in supervised learning setting *P* is unknown, while in LWSD, *P* is a union of *t* sub-distributions.

On the other hand, if the group labels of future test samples are given, the problem is **M**ulti-**T**ask **L**earning (MTL) problem [1,2,8]. The goal in multi-task learning is to find *t* mapping functions $\{f_1, f_2,..., f_t\}$ such that the sum of expected losses for each task

$$R(w) = \sum_{r=1}^{t} (\int L(f_r(\mathbf{x}, w), y) P_r(\mathbf{x}, y) d\mathbf{x}dy)$$

is minimized. Figure 1 illustrates that standard supervised learning, multi-task learning and learning with structured data handle training and test data in different ways.

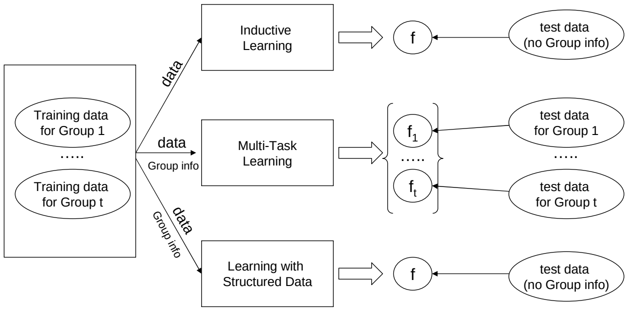


Figure 1: Inductive leaning, Multi-task learning, and Learning with structured data use different ways to handle training and test data.

"Learning with structured data" formulation and multi-task learning formulation are similar in the sense that they all try to exploit the group information. However, there are several important differences: (1) LWSD comes out one model, while MTL comes out t models; (2) LWSD does not require group membership of new testing data, but MTL does require that. Let's consider two realistic application problems that distinguish the two formulations. One example is handwritten digit recognition, where the training data originates from t persons (each person provides labeled examples of all 10 digits). Then goal 1 (LWSD) is to find a classifier that can generalize well for other (previously unseen) samples written by these people (we don't know who writes this test samples). In contrast, goal 2 (MTL) is improved generalization for each person who contributed to training data (ie. group membership for future samples is known). Another application example is fMRI data analysis or more generally, medical diagnosis. Here you try to estimate a predictive model (predict/diagnose a disease) from the training samples from t patients. Then goal 1 (LWSD) is to find a predictive model that has good generalization for other (new) samples from these patients, whereas the goal 2 (MTL).

We show empirical comparisons between different learning approaches for utilizing group information in the data. In particular, we compare:

- multiple SVM approach where a separate SVM classifier is estimated for each group
- SVM+ approach implementing LWSD setting
- SVM+MTL implementing multi-task learning using SVM+ methodology [9].

Comparisons are performed using synthetic data. Comparison results indicate that there is no single winner, and that relative performance strongly depends on the size of training data set.

Acknowledgement: This work was supported, in part, by NSF grant ECCS 0802056, and by the A. Richard Newton Breakthrough Research Award from Microsoft Research

REFERENCES

[1] Ando, R. and Zhang, T. A Framework for Learning predictive structures from multiple tasks and unlabeled data, Journal of Machine Learning Research, 2005.

[2] Ben-David, S., Gehrke, J. and Schuller, R. A theoretical framework for learning form a pool of disparate data sources. ACM KDD, 2002.

[3] Cherkassky, V. and Mulier, F. Learning from Data, John Wiley & Sons, New York, second edition, 2007.

[4] Hastie, T. , Tibshirani, R. and Friedman, J. The Elements of Statistical Learning. Data Mining, Inference and Prediction, Springer, 2001.

[5] Vapnik, V. Estimation of Dependences Based on Empirical Data, Springer Verlag, New York, 1982.

[6] Vapnik, V. Statistical Learning Theory, Wiley, New York, 1998.

[7] Vapnik, V. Empirical Inference Science Afterword of 2006, Springer, 2006.

[8] Evgeniou, T. and Pontil, M.. Regularized multi--task learning. In Proc. 17th SIGKDD Conf. on Knowledge Discovery and Data Mining, 2004.

[9] Liang, L. and V. Cherkassky, Connection between SVM+ and multi-task learning, Proc. IJCNN-2008, Hong Kong, China.

Computational Challenges in Modeling, Control and Optimization in Electric Power and Energy Systems

Ganesh Kumar Venayagamoorthy, *Senior Member*, IEEE
Real-Time Power and Intelligent Systems Laboratory
Department of Electrical and Computer Engineering
Missouri University of Science and Technology, Rolla, MO 65409, USA
gkumar@ieee.org

Extended Abstract

The electric power grid is faced with deregulation and an increased demand for high-quality and reliable electricity for our digital economy, and coupled with interdependencies with other critical infrastructures, it is becoming more stressed. The power grid control essentially requires a continuous balance between electrical power generation and a varying load demand, while maintaining system frequency, voltage levels and the power grid security. However, generator and grid disturbances can vary between minor and large imbalances in mechanical and electrical generated power, while the characteristics of a power system change significantly between heavy and light loading conditions, with varying numbers of generator units and transmission lines in operation at different times. The result is a highly complex and non-linear dynamic electric power grid with many operational levels made up of a wide range of energy sources with many interaction points. Thus, calls for fast and advanced modeling, control and optimization techniques. The dynamic stochastic optimization (DSO) of the electric power and energy systems and its parts can be formulated as minimization and/or maximization of certain quantities. Intelligent systems technology have an important role to play in carrying out DSO to improve the network efficiency and eliminate congestion problems without seriously diminishing reliability and security especially as the wind, solar and other forms of energy sources, which are intermittent, are integrated to the electric grid.

Computational intelligence is the study of adaptive mechanisms to enable or facilitate intelligent behavior in complex, uncertain and changing environments. These adaptive mechanisms include those artificial intelligence paradigms that exhibit an ability to learn or adapt to new situations, to generalize, abstract, discover and associate [1]. The typical paradigms of CI are neural networks, fuzzy systems, swarm intelligence, evolutionary computation and artificial immune systems. These paradigms can be combined to form hybrids algorithms [2].

Recently, computational intelligence techniques have received increasing attention in the area of modeling, control and optimization of power and energy systems. Power System Stabilizers (PSSs) are used as supplementary control devices to provide extra damping and improve the dynamic performance of the power system. Two bio-inspired algorithms, a Small Population based Particle Swarm Optimization (SPPSO) and Bacterial Foraging Algorithm (BFA), for the simultaneous design of multiple optimal PSSs has been investigated by the author in [3]. SPPSO is capable of exploration and exploitation like particle swarm optimization. The involvement of a number of stages in BFA greatly reduces the possibility of getting trapped in the local minima during the search process. This approach is an effort towards determining efficacies of small population based algorithms as a first step towards online optimization. These algorithms are selected in an effort to reduce the computational burden.

Adaptive critic designs (ACDs) are neural network designs capable of carrying out dynamic optimization, under conditions of noise and uncertainty. This family of ACDs brings new optimization techniques which combine concepts of reinforcement learning and approximate dynamic programming, thus making them powerful tools [4, 5]. The adaptive critic method provides a methodology for designing optimal nonlinear controllers using neural networks for complex systems such as the power system where accurate models are difficult to derive. The author has demonstrated that such techniques have the potential for the design of optimal controllers for generators and compensation devices such as FACTS (Flexible AC Transmission Systems) [6, 7].

In order to truly identify the changing dynamics of the power grid at all times and provide the appropriate control actions, the high computational power for fast dynamic modeling capability is needed. Traditional neural networks require more time to learn the system dynamics compared to echo state networks [8]. Wide area control based on wide area monitoring is critical to ensure the stability and security of the power grid [9]. This even makes the more demand on the computational speed and capabilities. The possible ways to overcome the computational challenges for real-time online modeling and optimization of power and energy systems will be discussed at the workshop.

The author's laboratory at the Missouri University of Science and Technology (Missouri S&T) – the Real-Time Power and Intelligent Systems (RTPIS) Laboratory - is developing computational intelligence techniques for the modeling, control and optimization of power and energy systems and numerous projects undertaken and currently in progress are depicted in Fig. 1.

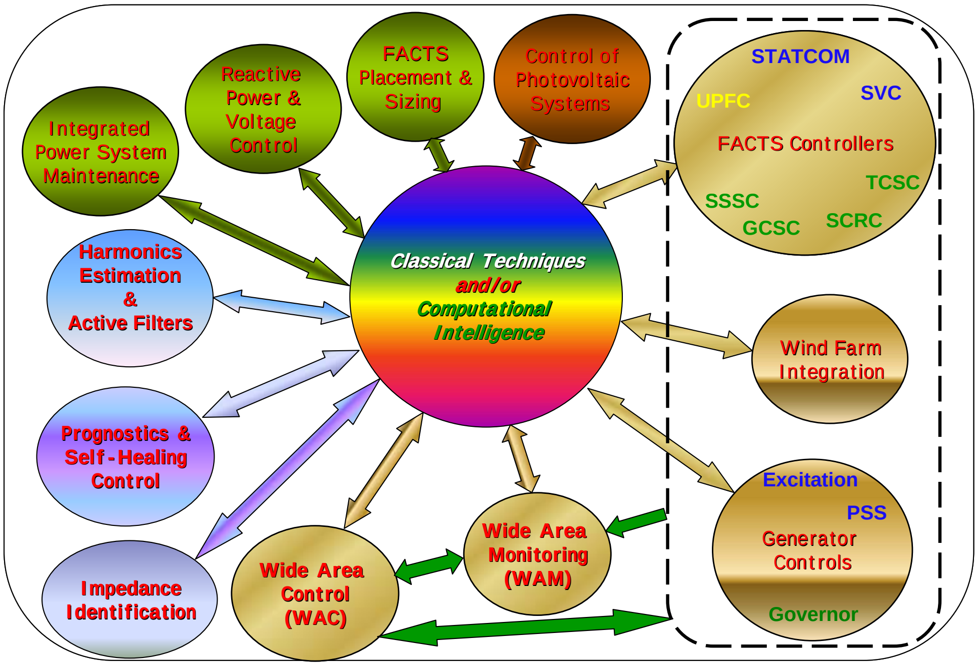

Integrated Power System Maintenance
Reactive Power & Voltage Control
FACTS Placement & Sizing
Control of Photovoltaic Systems
STATCOM
UPFC
SVC
FACTS Controllers
SSSC
TCSC
GCSC
SCRC
Harmonics Estimation & Active Filters
Classical Techniques and/or Computational Intelligence
Wind Farm Integration
Prognostics & Self-Healing Control
Excitation
PSS
Generator Controls
Governor
Wide Area Monitoring (WAM)
Wide Area Control (WAC)
Impedance Identification

Fig. 1 Projects at the RTPIS laboratory at Missouri S & T (http://rtpis.mst.edu).

References

[1] A. Engelbrecht, *Computational Intelligence: An Introduction*, John Wiley & Sons, Ltd, England, 2007, ISBN 978-0-470-03561-0.

[2] G. K. Venayagamoorthy, "A Successful Interdisciplinary Course on Computational Intelligence",*IEEE Computational Intelligence Magazine*, Vol. 4, No. 1, Feb. 2009.

[3] T. K. Das, G. K. Venayagamoorthy, U. O. Aliyu, "Bio-inspired Algorithms for the Design of Multiple Optimal Power System Stabilizers: SPPSO and BFA", *IEEE Transactions on Industry Applications*, Vol. 44, Issue 5, September/October 2008, pp. 1445-1457.

[4] P. J. Werbos, "Approximate Dynamic Programming for Real Time Control and Neural Modelling", in White DA and Sofge DA (Eds.), *Handbook of Intelligent Control*, Van Nostrand Reinhold, New York, 1992, ISBN 0-442-30857-4, pp. 493 – 525.

[5] D. V. Prokhorov, D. C. Wunsch, "Adaptive Critic Designs" *IEEE Trans. on Neural Networks*, Vol. 8, No. 5, September 1997, pp. 997 – 1007.

[6] G. K. Venayagamoorthy, "Dynamic Optimization of a Multimachine Power System with a FACTS device Using Identification and Control ObjectNets", *39th IEEE IAS Annual Meeting on Industry Applications*, Seattle, WA, USA, October 2004, pp. 2643-2650.

[7] G. K. Venayagamoorthy, R. G. Harley, D. C. Wunsch, "Comparison of Heuristic Dynamic Programming and Dual Heuristic Programming Adaptive Critics for Neurocontrol of a Turbogenerator", *IEEE Transactions on Neural Networks*, Volume: 13, Issue: 3, May 2002, Page(s): 764 -773.

[8] S. Ray, G. K. Venayagamoorthy, "Real-Time Implementation of a Measurement based Adaptive Wide Area Control System Considering Communication Delays", *IET Proceedings on Generation, Transmission and Distribution*, Vol. 2, 1, Jan. 2008, pp. 62-70.

[9] G. K. Venayagamoorthy, "Online Design of an Echo State Network Based Wide Area Monitor for a Multi-machine Power System", *Neural Networks*, vol. 20, issue 3, April 2007, pp. 404-413.

List of participants

Name	Affiliation	E-mail
Artur Abdullin	University of Louisville	ar.abdullin@louisville.edu
Chris Boyle	University of Louisville	chris.boyle@louisville.edu
Oscar Castillo	Tijuana Institute of Technology	ocastillo@tectijuana.mx
Vladimir Cherkasski	University of Minnesota	cherk001@umn.edu
Fahmida N. Chowdhury	NSF	fchowdhu@nsf.gov
Jan Gehrke	Universität Bremen, Bremen, Germany	jgehrke@tzi.de
James Gentle	George Mason University	jgentle@gmu.edu
Otthein Herzog	University of Bremen	herzog@tzi.de
Keith A. Howell	George Mason University	khowell@gmu.edu
Mo Jamshidi	University of New Mexico	moj@wacongo.org
Robert Kozma	University of Memphis	rkozma@memphis.edu
William F. Lawless	Paine College	lawlessw@mail.paine.edu
Jordan Malof	University of Louisville	jmmalo03@gmail.com
Maciej A. Mazurowski	University of Louisville	maciej.mazurowski@louisville.edu
Patricia Melin	Tijuana Institute of Technology	pmelin@tectijuana.mx
Scott F. Midkiff	NSF	smidkiff@nsf.gov
Khalid Moidu	George Mason University	kmoidu@gmail.com
Jaroslaw Pietrzykowski	George Mason University	jarek@mli.gmu.edu
Danil Prokhorov	Toyota Technical Center	danil.prokhorov@tema.toyota.com
Rammohan Ragade	University of Louisville	ragade@louisville.edu
Leon Reznik	Rochester Institute of Technology	lr@cs.rit.edu
Carlos Rojas	University of Louisville	ccroja01@louisville.edu
Yung C. Shin	School of Mechanical Engineering Purdue University	shin@purdue.edu
Usha Varshney	NSF	uvarshne@nsf.gov
Ganesh K. Venayagamoorthy	Missouri University of Science and Technology	ganeshv@mst.edu
Thomas Wagner	University of Bremen	twagner@tzi.de

Duminda Wijesekera	George Mason University	dwijesek@gmu.edu
Janusz Wojtusiak	George Mason University	jwojt@mli.gmu.edu
Jacek M. Zurada	University of Louisville	jmzura02@louisville.edu
Mark R. Zurada	Redon Group LLC, Louisville	mark@zurada.com