An Informational Energy Approach to Feature Selection

A. Cațaron* and R. Andonie**
* Transilvania University of Brasov, Romania
** Central Washington University, USA

***Abstract* – In this work, we focus on machine learning methods for handling data sets containing large amounts of irrelevant information. We address two key issues: the problem of selecting relevant features, and the problem of weighting (ranking) these features. We describe our Energy Supervised Relevance Neural Gas (ESRNG) algorithm, a kernel method which uses the maximization of Onicescu's informational energy as a criteria to compute the relevance of the input features for an LVQ classification system.**

I. INTRODUCTION

Feature selection has become the focus of much research in areas of application for which datasets with tens or hundreds of thousands of variables are available. These areas include text processing of internet documents, gene expression array analysis, and combinatorial chemistry. The objective of feature selection is three-fold: improving the performance of the predictors/classifiers, providing faster and more cost-effective predictions/classifications, and providing a better understanding of the underlying process that generated the data [9].

A common approach, especially for embedded algorithms, is to apply a weighting (ranking) function to features, in fact assigning them degrees of perceived *relevance*. Explicit feature selection is generally most natural when the result is intended to be understood by humans, or fed into another algorithm. Weighting schemes tend to be easier to implement in on-line incremental settings, and are generally more purely motivated by performance considerations. Weighting schemes can be viewed in terms of heuristic search, as we viewed explicit feature selection methods. However, because the weight space lacks the partial ordering of feature sets, most approaches to feature weighting rely on quite different forms of search. For instance, the most common is some form of gradient descent, in which training instances lead to simultaneous changes in all weights [10].

Several approaches to the feature selection problem using information theoretic criteria have been proposed (as reviewed in the March 2003 special issue of Journal of Machine Learning Research). many rely on empirical estimates of the mutual information (MI). Mutual information is a good indicator of the relevance between variables, and has been used as a measure in several feature selection algorithms. In this case, the MI evaluates the "information content" of each individual feature with regard to the output class. The feature selection method is searching for a subset of relevant features from an initial set of available features. A sensible part of this approach is the estimation of the MI, because of the requirements for the conditional density functions and the high computational complexity. Many MI-based feature selection algorithms used histogram as density estimator. In high dimensional space, histograms are neither effective nor accurate. A MI estimation technique based on Renyi's quadratic entropy and Parzen windows was proposed by Principe et al. [7] and has been used in efficient feature selection [11]. Torkkola also used this estimation method for feature extraction by MI maximization [12].

The neural-gas (NG) algorithm, introduced in [1], represents a neural model which is applied to the task of vector quantization by using a neighborhood cooperation scheme. The NG network uses a soft-max adaptation rule, similar to the Kohonen feature map. It replaces the Euclidean distance with the neighborhood ranking of the reference vectors for a given input vector. The advantage of using the NG network is avoiding the dependency on the initialization of reference vectors.

The Supervised Relevance Neural Gas (SRNG) algorithm, combines the NG and the Generalized Relevance Neural Gas (GRLVQ) algorithm [3]. The idea was to incorporate neighborhood cooperation of NG into the GRLVQ to speedup the convergence and make initialization less crucial.

In our previous work [8], we have estimated the MI using Onicescu's informational energy [4]. Our estimation was incorporated in two existent weighted LVQ type algorithms. Essentially, we have obtained incremental

learning algorithms for supervised classification and feature ranking.

In this paper we present the Energy SRNG (ESRNG) algorithm, which uses the maximization of the informational energy (IE) for computing the weights of input features. This adaptive relevance determination is used in combination with the SNG model, for feature ranking and selection.

II. Supervised Relevance Neural Gas Algorithm

Let us consider the implementation of a clustering of data into M classes, $c_1,\ldots,c_M$, by using a set of training data defined as follows:

$$X=\left\{\left(\mathbf{x}_i,c_i\right)\subset\mathbf{R}^n\times\left\{1,\ldots,M\right\}\middle|i=1,\ldots,N\right\}.$$

The n components of the training vectors are:

$$\mathbf{x}_i=\left[x_{i1},\ldots,x_{in}\right].$$

Each class will be described after the training by a subset of reference vectors from $\mathbf{R}^n$. Denote the set of all K reference vectors by:

$$W=\left\{\mathbf{w}_1,\ldots\mathbf{w}_K\right\},$$

and the components of the reference vectors by:

$$\mathbf{w}_j=\left\lfloor w_{j1},\ldots,w_{jn}\right\rfloor.$$

The neural gas algorithm uses the neighborhood ranking of the reference vectors which is determined each time a training vector is applied to the input of the neural network. For this, all Euclidean distances between the input sample $\mathbf{x}_i$ and each reference vector $\mathbf{w}_j$, $j\in\left\{1,\ldots,K\right\}$ are sorted in an increasing order. The rank of a particular reference vector $\mathbf{w}_j$ for a given input $\mathbf{x}_i$ equals to the number of reference vectors that are in the relation:

$$\left\|\mathbf{x}_i-\mathbf{w}_k\right\|\le\left\|\mathbf{x}_i-\mathbf{w}_j\right\|,\ j,k\in\left\{1,\ldots,K\right\},\ j\ne k.$$

The rank of $\mathbf{w}_j$ will be denoted by:

$$r_j\left(\mathbf{x}_i,W\right)$$

and this is a function yielding the dependence both on $\mathbf{x}_i$ and the entire set of reference vectors W.

The cost function optimized by the NG algorithm is [1], [2]:

$$C_{NG}=\frac{1}{C(\gamma)}\sum_{\mathbf{w}_j\in W}\sum_{\mathbf{x}_i\in X}h_\gamma\left(r_j\left(\mathbf{x}_i,W\right)\right)\left\|\mathbf{x}_i-\mathbf{w}_j\right\|^2$$

where

$$C(\gamma)=\sum_{r=0}^{K-1}h_\gamma(r),$$

$$h_\gamma\left(r_j\left(\mathbf{x}_i,W\right)\right)=e^{-r_j\left(\mathbf{x}_i,W\right)/\gamma},$$

and the neighborhood range is determined by γ.

The LVQ learning rule that uses the neighborhood range for updating the reference vector is [1], [2]:

$$\Delta\mathbf{w}_j=\eta h_\gamma\left(r_j\left(\mathbf{x}_i,W\right)\right)\left(\mathbf{x}_i-\mathbf{w}_j\right),$$

where η is a positive learning rate. Not only the winner is updated, but all reference vector with a degree given by h_γ.

The GLVQ algorithm [1] updates two reference vectors, $\mathbf{w}_j$ and $\mathbf{w}_k$, the closest to the input vector $\mathbf{x}_i$, the first one from the same class with $\mathbf{x}_i$ and the second one from a different class. A relative distance which ranges between -1 and 1 is defined by:

$$\mu\left(\mathbf{x}_i\right)=\frac{d_j-d_k}{d_j+d_k},$$

where $d_j=\left\|\mathbf{x}_i-\mathbf{w}_j\right\|$ and $d_k=\left\|\mathbf{x}_i-\mathbf{w}_k\right\|$. The relative distance has negative values if the input vector is classified correctly only. The GLVQ algorithm minimizes the following cost function:

$$S=\sum_{i=1}^{N}f\left(\mu\left(\mathbf{x}_i\right)\right),$$

with N the number of training input vectors and f is a monotonically increasing function. The GLVQ updating rule is:

$$\mathbf{w}_j^{(t+1)}=\mathbf{w}_j^{(t)}+\eta\frac{\partial f}{\partial\mu}\frac{d_k}{\left(d_j+d_k\right)^2}\left(\mathbf{x}_i-\mathbf{w}_j\right)$$

$$\mathbf{w}_k^{(t+1)}=\mathbf{w}_k^{(t)}-\eta\frac{\partial f}{\partial\mu}\frac{d_j}{\left(d_j+d_k\right)^2}\left(\mathbf{x}_i-\mathbf{w}_k\right).$$

By incorporating the neural gas rule into the GLVQ algorithm, the objective function of neural gas can be reformulated as [2]:

$$C_{SNG} = \sum_{\mathbf{x}_i \in X} \sum_{\mathbf{w}_j \in W^{\mathbf{x}_i}} \frac{h_\gamma\left(r_j\left(\mathbf{x}_i, W^{\mathbf{x}_i}\right)\right) f\left(\mu\left(\mathbf{x}_i, \mathbf{w}_j\right)\right)}{C\left(\gamma, K^{\mathbf{x}_i}\right)},$$

Where $W^{\mathbf{x}_i}$ is a subset of W which contains the reference vectors from the same class with $\mathbf{x}_i$, $\mu\left(\mathbf{x}_i, \mathbf{w}_j\right) = \frac{\left\|\mathbf{x}_i - \mathbf{w}_j\right\| - d_k}{\left\|\mathbf{x}_i - \mathbf{w}_j\right\| + d_k}$, $C\left(\gamma, K^{\mathbf{x}_i}\right) = \sum_{r=0}^{K^{\mathbf{x}_i}-1} h_\gamma(r)$ and $K^{\mathbf{x}_i}$ is the cardinality of $W^{\mathbf{x}_i}$.

The GRLVQ algorithm [3] is an extension of GLVQ and associates a relevance factor to each input component of the classification system. We denote the relevance vectors by:

$$\lambda = \left[\lambda_1, \ldots \lambda_n\right]$$

where n is the dimension of the input vectors $\mathbf{x}_i$, i=1,…,N. The relevance factors have the following property:

$$\sum_{k=1}^{n} \lambda_k = 1.$$

Instead of the Euclidean distance, the GRLVQ algorithm uses a weighted distance between an input vector $\mathbf{x}_i$ and a reference vector $\mathbf{w}_j$:

$$D_{ij} = \sqrt{\sum_{k=1}^{n} \lambda_k \left(x_{ik} - w_{jk}\right)^2}.$$

By using this distance, one can reformulate the relative distance defined by the GLVQ algorithm as follows:

$$\mu_\lambda\left(\mathbf{x}_i\right) = \frac{D_{ij} - D_{ik}}{D_{ij} + D_{ik}}.$$

The SRNG algorithm was obtained by including the NG idea in the GRLVQ algorithm [2] and the cost function optimized by this algorithm is:

$$C_{SRNG} = \sum_{\mathbf{x}_i \in X} \sum_{\mathbf{w}_j \in W^{\mathbf{x}_i}} \frac{h_\gamma\left(r_j\left(\mathbf{x}_i, W^{\mathbf{x}_i}\right)\right) f\left(\mu_\lambda\left(\mathbf{x}_i, \mathbf{w}_j\right)\right)}{C\left(\gamma, K^{\mathbf{x}_i}\right)},$$

with $\mu_\lambda\left(\mathbf{x}_i, \mathbf{w}_j\right) = \frac{\left|\mathbf{x}_i - \mathbf{w}_j\right|_\lambda^2 - D_{ik}}{\left|\mathbf{x}_i - \mathbf{w}_j\right|_\lambda^2 + D_{ik}}$ and D_{ik} is the weighted distance between $\mathbf{x}_i$ and the closest reference vector that does not belong to $W^{\mathbf{x}_i}$. The SRNG update rule [2] applies to all reference vectors from $W^{\mathbf{x}_i}$:

$$\begin{aligned} \Delta\mathbf{w}_j = & -\sum_{\mathbf{w}_j \in W^{\mathbf{x}_i}} \eta_1 \lambda \mathbf{I} \frac{\partial f}{\partial \mu} \frac{\left|\mathbf{x}_i - \mathbf{w}_j\right|_\lambda^2}{\left(\left|\mathbf{x}_i - \mathbf{w}_j\right|_\lambda^2 + D_{ik}\right)^2} \cdot \\ & \cdot \left(\mathbf{x}_i - \mathbf{w}_k\right) \frac{r_j\left(\mathbf{x}_i, W^{\mathbf{x}_i}\right)}{C\left(\gamma, K^{\mathbf{x}_i}\right)} \end{aligned} \quad (1)$$

and to the closest reference vector that does not belong to this set:

$$\begin{aligned} \Delta\mathbf{w}_j = & \eta \lambda \mathbf{I} \frac{\partial f}{\partial \mu} \frac{D_{ik}}{\left(\left|\mathbf{x}_i - \mathbf{w}_j\right|_\lambda^2 + D_{ik}\right)^2} \cdot \\ & \cdot \left(\mathbf{x}_i - \mathbf{w}_j\right) \frac{r_j\left(\mathbf{x}_i, W^{\mathbf{x}_i}\right)}{C\left(\gamma, K^{\mathbf{x}_i}\right)} \end{aligned} \quad (2)$$

with η and η_1 two positive constants. In our tests we used the sigmoid function:

$$f(\mu) = \frac{1}{1 + e^{-\mu\varepsilon}},$$

where ε is a positive constant, for which:

$$\frac{\partial f}{\partial \mu} = f(\mu)(1 - f(\mu)).$$

III. Computation Of The Relevance Factors With Informational Energy

The relevance factors are a set of coefficients associated to the input features. We will describe a method to obtain them by using an informational energy approach.

The discrete informational energy of a random variable X with probabilities p_k, k=1,…,n was defined by Onicescu in [4]:

$$E(X) = \sum_{k=1}^{n} p_k^2$$

and the continuous informational energy of the continuous random variable Y was defined by [5]:

$$E(Y)=\int_{-\infty}^{\infty} p^2(\mathbf{y})d\mathbf{y}$$

where $p(\mathbf{y})$ is the probability density of the random variable.

We will determine the relevance factors by maximizing the following measure of unilateral dependency between two random variables X and Y [6]:

$$o(Y,X)=E(Y\mid X)-E(Y)$$

where $E(Y \mid X)$ is the conditional informational energy. For two random variables Y continuous and C discrete, we can write:

$$E(Y\mid C)=\int_{\mathbf{y}}\sum_{p=1}^{M} p(c_p)p^2(\mathbf{y}\mid c_p)d\mathbf{y}\ .$$

The dependence measure $o(Y,X)$ is not symmetrical with respect to its arguments, is positive if and only if Y and X are independent and is upper limited by 1-$E(Y)$ if and only if Y is completely dependent on X.

We consider M classes labels $c_1,\ldots,c_M$ as samples of a discrete random variable denoted by C. The reference vectors $\mathbf{w}_j$, j=1,…,P are the prototypes of the classes and are determined by training with an algorithm from the LVQ category. The training vectors $\mathbf{x}_i$, i=1,…,N belong to one of the M classes. Then, we can introduce a continuous variable Y with its samples $\mathbf{y}_i$, i=1,…,N defined by the following transform that relates an input vector $\mathbf{x}_i$ and their corresponding class j by the means of the reference vector $\mathbf{w}_j$ and the relevance vector $\boldsymbol{\lambda}$:

$$\mathbf{y}_i=\lambda\mathbf{I}(\mathbf{x}_i-\mathbf{w}_j).$$

Therefore, the relevance factors can be updated using an ascending gradient procedure that maximizes the dependence measure $o(Y,C)$**:**

$$\lambda^{(t+1)}=\lambda^{(t)}+\alpha\sum_{i=1}^{N}\frac{\partial o(Y,C)}{\partial\mathbf{y}_i}\mathbf{I}(\mathbf{x}_i-\mathbf{w}_j).$$

To compute this expression, we need to evaluate the partial derivative. By rewriting the definition of the dependence measure, we obtain:

$$o(Y,C)=\sum_{p=1}^{M}\frac{1}{p(c_p)}\int_{\mathbf{y}}p^2(\mathbf{y},c_p)d\mathbf{y}-\int_{\mathbf{y}}p^2(\mathbf{y})d\mathbf{y}\ .$$

The two integrals involve a considerable computational effort and we will choose to approximate them with the Parzen windows estimation method with the Gaussian kernel $G(\mathbf{y}-\mathbf{y}_i,\sigma)=\frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{\|\mathbf{y}-\mathbf{y}_i\|^2}{2\sigma}}$. The probability density $p(\mathbf{y})$ can be expressed as [7]:

$$p(\mathbf{y})=\frac{1}{N}\sum_{i=1}^{N}G(\mathbf{y}-\mathbf{y}_i,\sigma^2).$$

We can then write [8]:

$$\int_{\mathbf{y}}p^2(\mathbf{y},c_p)d\mathbf{y}=\frac{1}{N^2}\sum_{k=1}^{N_p}\sum_{l=1}^{N_p}G(\mathbf{y}_{pk}-\mathbf{y}_{pl},2\sigma^2)$$

and

$$\int_{\mathbf{y}}p^2(\mathbf{y})d\mathbf{y}=\frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N}G(\mathbf{y}_k-\mathbf{y}_l,2\sigma^2),$$

where $\mathbf{y}_{pk}$, $\mathbf{y}_{pl}$ are two training samples from class p, $\mathbf{y}_k$, $\mathbf{y}_l$ are two training samples from any class and N_p is the number of the training samples from the class p.

Using these two expressions, we obtain [8]:

$$\begin{aligned}o(Y,C) &= \frac{1}{N}\left(\sum_{p=1}^{M}\frac{1}{N_p}\right)\sum_{k=1}^{N_p}\sum_{l=1}^{N_p}G(\mathbf{y}_{pk}-\mathbf{y}_{pl},2\sigma^2)-\\ &-\frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N}G(\mathbf{y}_k-\mathbf{y}_l,2\sigma^2).\end{aligned}$$

To evaluate this expression, we use two consecutive samples $\mathbf{y}_1$ and $\mathbf{y}_2$ as classes representatives. In the case when the two training vectors are from different classes, we obtain:

$$o(Y,C)=G(0,2\sigma^2)-\frac{1}{2}G(\mathbf{y}_1-\mathbf{y}_2,2\sigma^2).$$

If the samples belong to the same class, $o(Y,C)$ cannot be evaluated.

The rule to update the relevance factors will finally become:

$$\lambda^{(t+1)} = \lambda^{(t)} - \alpha \frac{1}{4\sigma^2} G\left(\mathbf{y}_1 - \mathbf{y}_2, 2\sigma^2\right)\left(\mathbf{y}_2 - \mathbf{y}_1\right)\mathbf{I} \cdot \cdot \left(\mathbf{x}_1 - \mathbf{w}_{j(1)} - \mathbf{x}_2 + \mathbf{w}_{j(2)}\right).$$

It is straightforward to proof that $o(Y,C)$ is a positive defined kernel. When the two samples are different, we have $\|\mathbf{y}_1 - \mathbf{y}_2\|^2 > 0$ and $G(0,2\sigma^2) > G(\mathbf{y}_1 - \mathbf{y}_2, 2\sigma^2)$, meaning that $o(Y,C) > 0$. The weighted Euclidean metric we use allows for a direct interpretation as kernelized NG if the relevances are fixed [2]. Therefore, the relevances should remain unchanged after processing each input pattern. This may be achieved if we allow a preprocessing of the patterns with the relevances computed first.

IV. The ESRNG Algorithm And Experiments

This algorithm adapts the reference vectors for as least as possible quantization error on all feature vectors. After initializing the relevance vector with the values $\lambda_k = 1/n$, k=1,…,n, the reference vectors and the parameters η, α and σ, we apply the following steps to incrementally update the relevances, the reference vectors and the feature ranks for a given input $\mathbf{x}_i$:

1. Update the codebook vectors using the SRNG relations.
2. Update the relevances according to our formula and apply a transform on the new values.
3. Update the overall rank of each feature as an average over all previous steps.

The transform applied to relevances in the step 2 is an operation that keeps the relevance values in a reasonable domain. The squared weighted distance between an input vector $\mathbf{x}_i$ and a reference vector $\mathbf{w}_j$, $D_{ij}^2 = \sum_{k=1}^{n} \lambda_k \left(x_{ik} - w_{jk}\right)^2$, requires that all relevances to be positive. If at least one relevance is negative, this condition can be realized by transforming all relevances with:

$$\lambda_k = \frac{e^{\lambda_k}}{\sum_{i=1}^{n} e^{\lambda_i}} + \varepsilon$$

or by scaling:

$$\lambda_k = \lambda_k + \min_{i=1,\ldots,n} \lambda_i + \varepsilon$$

where ε is a positive constant.

We tested ESRNG on three well known databases [13]: Iris, Ionosphere and Vowel recognition. The experimental results describe the behavior of different systems in similar conditions. In Table I we present the comparative recognition rates obtained with ESRNG versus RLVQ, GRLVQ, SRNG, ERLVQ and EGRLVQ on the three datasets. Tables II, III and IV show the ranking of the input features generated by the same algorithms, while the Figure 1 offers a visual image of the average values of the feature relevances obtained with ESRNG.

TABLE I.
The Comparative Recognition Rates

	Iris	Vowel	Ionosphere
RLVQ	95.33%	46.32%	92.71%
GRLVQ	96.66%	46.96%	93.37%
SRNG	96.66%	47.61%	94.03%
ERLVQ	97.33%	47.18%	94.03%
EGRLVQ	97.33%	47.18%	94.40%
ESRNG	97.33%	47.61%	94.03%

TABLE II.
The Ranking Of The Features From The Iris Database

Rank	1	2	3	4
RLVQ	4	2	3	1
GRLVQ	4	3	2	1
SRNG	3	4	2	1
ERLVQ	1	2	3	4
EGRLVQ	1	3	4	2
ESRNG	3	1	4	2

TABLE III.
The Ranking Of The Features From The Vowel Recognition Database

Rank	1	2	3	4	5	6	7	8	9	10
RLVQ	2	5	1	9	6	3	4	8	7	10
GRLVQ	2	5	4	6	3	1	9	7	8	10
SRNG	1	4	6	2	3	9	8	5	7	10
ERLVQ	2	1	3	4	6	8	9	5	10	7
EGRLVQ	3	1	2	6	5	4	9	8	7	10
ESRNG	2	1	3	8	9	4	5	10	8	7

TABLE IV.
The Ranking Of The Features From The Ionosphere Dataset

Rank	1	2	3	4	5
RLVQ	20	28	26	12	6
GRLVQ	12	4	22	8	6
SRNG	24	15	12	10	21
ERLVQ	8	24	16	12	6
EGRLVQ	4	5	12	8	27
ESRNG	14	8	5	16	3

The Iris database consists of 150 vectors from 3 classes. We used 6 reference vectors and the values of the training parameters were η = 1 and η_1 = 0.5. The third component was ranked as most important, while the least important was the second component.

The Vowel recognition tests were performed by using 59 reference vectors with the training parameters having

the values $\eta = 0.7$ and $\eta_1 = 0.5$. The second feature was considered as most important and the features from positions 10 and 7 were ranked between the least important.

Finally, for the Ionosphere dataset tests we used 8 reference vectors trained with 200 instances and tested with the remaining 151 instances, as specified in [13]. The training parameters were $\eta = 0.04$ and $\eta_1 = 0.03$.

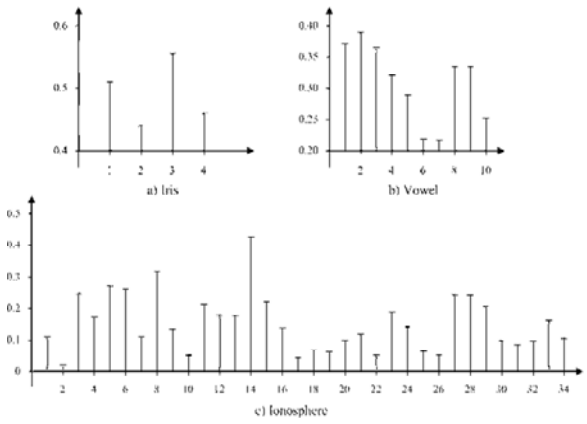

Figure 1. The average values of the feature relevances

V. CONCLUSIONS

Our ESRNG algorithm is an incremental learning algorithm for feature ranking and supervised classification. It was successfully tested on different standard datasets. In our future work, we plan to use this method for backward feature selection. In backward feature selection [9,10], we start with the maximum number of input features and decrement at each iteration, if necessary, the number of features. This approach modifies on-line the structure of the LVQ network and this may create a problem.

REFERENCES

[1] T.M. Martinetz, S.G. Berkovich and K.J. Schulten, *Neural-gas network for vector quantization and its application in time-series prediction*, IEEE Trans. on Neural Networks, 4, 558-569, 1993.

[2] B. Hammer, M. Strickert and T. Villmann, *Supervised neural gas with general similarity measure.* Neural Processing Letters, vol. 21, no. 1, 21-44, 2005.

[3] B. Hammer and T. Villmann, *Generalized relevance learning vector quantization*, Neural Networks, vol. 15, 1059-1068, 2002.

[4] O. Onicescu, *Theorie de l'information. Energie informationelle*. C. R. Acad. Sci., Ser. A-B, vol. 263, 841-842, 1966.

[5] S. Guiasu, *Information theory with applications*, McGraw Hill, New York, 1977.

[6] R. Andonie and F. Petrescu, *Interacting systems and informational energy*, Foundation of Control Engineering, vol. 11, 53-59, 1986.

[7] J.C. Principe, D. Xu and J.W. Fisher III, *Information-theoretic learning*, In: Unsupervised Adaptive Filtering, S. Haykin, Wiley, New York, 2000.

[8] A. Cataron and R. Andonie, *Energy generalized LVQ with relevance factors*. Proceedings of the IEEE International Joint Conference on Neural Networks IJCNN2004, Budapest, Hungary, July 26-29, 1421-1426, 2004.

[9] I. Guyon and A. Elisseeff, *An introduction to variable and feature selection,* Journal of Machine Learning Research, 3, 2003, 1157-1182.

[10] A. Blum and P. Langley, *Selection of relevant features and examples in machine learning*, Artificial Intelligence, 97, 1997, 245-271.

[11] D. Huang and T. Chow, *Searching optimal feature subset using mutual information*, in Proceedings of the European Symposium on Artificial Neural Networks (ESANN 2003), M. Verleysen, Ed., D-side publications, 2003, pp. 161–166.

[12] K. Torkkola, *Feature extraction by non-parametric mutual information maximization,* Journal of Machine Learning Research, vol. 3, pp. 1415–1438, 2003.

[13] K. Blacke, E. Keogh and C.J. Merz, *UCI Repository of Machine Learning Databases.* Available: http://www.ics.uci.edu/~mlearn/MLSummary.html, 1998.

[14] M. Tesmer and P.A. Estevez, *AMIFS: Adaptive Feature Selection by Using Mutual Information,* Proceedings of the IEEE International Joint Conference on Neural Networks IJCNN2004, Budapest, Hungary, July 26-29, 303-308, 2004.