Feature Ranking using Supervised Neural Gas and Informational Energy

Răzvan Andonie
Computer Science Department
Central Washington University, Ellensburg, USA
Email: andonie@cwu.edu

Angel Caţaron
Department of Electronics and Computers
Transylvania University of Braşov, Romania
Email: cataron@vega.unitbv.ro

***Abstract*— In this paper we use the maximization of Onicescu's informational energy as a criteria for computing the relevances of input features. This adaptive relevance determination is used in combination with the neural gas and the generalized relevance LVQ algorithms. The idea of applying the neural gas neighborhood cooperation technique to improve the generalized relevance LVQ is due to Hammer *et al.* and is best described in [1]. Our approach gives an alternative way for determining the relevances in Hammers's algorithm, and in our experiments it shows at least the same performances. Our contribution is an incremental learning algorithm for supervised classification and feature ranking.**

I. Introduction

Mutual information (MI) 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 [2], [3]. However, in high dimensional space, histograms are neither effective nor accurate [4]. A MI estimation technique based on Renyi's quadratic entropy and Parzen windows was proposed by Principe *et al.* [5] and has been used in an efficient feature selection algorithm [4]. Torkkola also used this estimation method for feature extraction by MI maximization [6]. Other MI based feature selection methods are described in [7] and [8].

Our MI-based feature ranking approach is used in the context of Kohonen's supervised LVQ algorithms [9], [10]. Standard LVQ does not discriminate between more or less informative features: their influence on the distance function is equal. On the contrary, the Relevance LVQ (RLVQ), introduced in [11], holds a changeable relevance value for every feature and employs a weighted distance function for classification. An iterative heuristic training process is used to tune the weight values for a specific problem: the influence of features which frequently contribute to miss classifications of the system is reduced while the influence of very reliable features is increased. A modification of RLVQ has been proposed by Hammer *et al.* [12], Generalized RLVQ (GRLVQ), which obeys a stochastic gradient descent on an energy function. This method modifies the GLVQ algorithm (introduced in [13]) by using an adaptive metric, and leads to a more powerful classifier with little extra cost compared to GLVQ.

The neural-gas (NG) algorithm, introduced in [14], 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 most recent proposed model in this sequence is the Supervised Relevance Neural Gas (SRNG) algorithm, which combines the NG and the GRLVQ [1]. 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 [15], [16], rather than using Renyi's entropy, we have estimated the MI using Onicescu's informational energy [17]. Our estimation was incorporated in two existent weighted LVQ type algorithms: Relevance LVQ (RLVQ) [11] and Generalized Relevance LVQ (GRLVQ) [12]. Essentially, we have obtained incremental learning algorithms for supervised classification and feature ranking.

In this paper we introduce the Energy SRNG (ESRNG) algorithm, which uses the maximization of the informational energy (IE) as a criteria for computing the relevances of input features. This adaptive relevance determination is used in combination with the SNG model. The hierarchy of these neural models in described in Fig. 1.

In Section II we introduce the basic notations used in the NG, GLVQ, SNG, and SRNG algorithms. Section III describes our IE approximation technique and Section IV the ESRNG algorithm. In Section V we compare ESRNG to the basic SRNG model and to other algorithms of this family. Section VI, concludes with some closing remarks.

II. Supervised Relevance Neural Gas

Assume that a clustering of data into M classes, $c_1, \ldots, c_M$, is to be learned and a set of training data is given:

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

0-7803-9048-2/05/$20.00 ©2005 IEEE

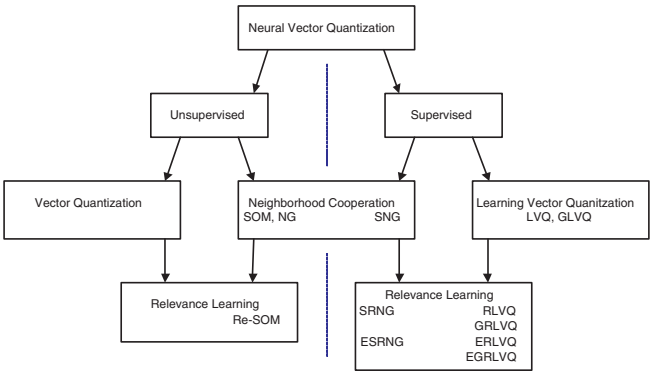

Fig. 1. Supervised Neural Gas and Relevance Learning in LVQ (adapted from [18])

The components of a vector $\mathbf{x}_i$ are $[x_{i1}, \ldots, x_{in}]$.

A subset of reference vectors from $\mathbf{R}^n$ are assigned to each class. Denote the set of all reference vectors by $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$. The components of a vector $\mathbf{w}_j$ are $[w_{j1}, \ldots, w_{jn}]$.

The neighborhood ranking of the reference vectors is determined each time a training vector is applied to the input of the neural network in the following way. The Euclidean distances between input sample $\mathbf{x}_i$ and all reference vectors $\mathbf{w}_j$, $j \in \{1, \ldots, K\}$ are sorted in increasing order. Therefore, the rank of a particular vector $\mathbf{w}_j$ for a given input vector $\mathbf{x}_i$ equals to the number of reference vectors that are in the relation $\|\mathbf{x}_i - \mathbf{w}_k\| \leq \|\mathbf{x}_i - \mathbf{w}_j\|$, where $j, k \in \{1, \ldots, K\}$ and $j \neq k$. We will denote the rank of $\mathbf{w}_j$ by $r_j(\mathbf{x}_i, W)$ because it depends on both $\mathbf{x}_i$ and the whole set W of reference vectors.

The NG algorithm optimizes the following cost function [14], [1]:

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

where $h_\gamma(r_j(\mathbf{x}_i, W)) = e^{-r_j(\mathbf{x}_i, W)/\gamma}$, γ determines the neighborhood range and $C(\gamma) = \sum_{r=0}^{K-1} h_\gamma(r)$. The learning rule is [14], [1]:

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

where η is a positive learning rate. By incorporating the neighborhood range, this learning rule not only adapts the winner, but all reference vectors, with a degree given by h_γ.

Two reference vectors, $\mathbf{w}_j$ and $\mathbf{w}_k$ are considered. They are 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 another class. A relative distance difference is defined by $\mu(\mathbf{x}) = \frac{d_j - d_k}{d_j + d_k}$, where $d_j = \|\mathbf{x}_i - \mathbf{w}_j\|$, $d_k = \|\mathbf{x}_i - \mathbf{w}_k\|$. This function, which ranges between -1 and 1, has negative values if the input vector is classified correctly and has positive values if the input vector is classified incorrectly. The following cost function is minimized: $S = \sum_{i=1}^{N} f(\mu(\mathbf{x}_i))$, where N is the number of input vectors used in the training process and f is a monotonically increasing function. The codebook vectors are modified as follows:

$$\mathbf{w}_j^{(t+1)} = \mathbf{w}_j^{(t)} - \eta \frac{\partial S}{\partial \mathbf{w}_j}$$
$$\mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)} - \eta \frac{\partial S}{\partial \mathbf{w}_k},$$

where η is the learning rate. Hence, the update rule of the GLVQ algorithm is:

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

By incorporating the NG rule into the GLVQ algorithm, the objective function of NG can be reformulated as follows [1]:

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

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

The Generalized Relevance LVQ (GRLVQ) algorithm [12] associates a relevance factor to each input component. We define the relevance vector by $\lambda = [\lambda_1, \ldots, \lambda_n]$, $\sum_{k=1}^{n} \lambda_k = 1$, where n is the dimension of the input vectors $\mathbf{x}_i$, $i = 1, \ldots, N$. The GRLVQ algorithm uses a weighted distance between an input vector $\mathbf{x}_i$ and a reference vector $\mathbf{w}_j$:

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

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

The GLVQ algorithm can be reformulated to minimize an objective function based on this modified distance, yielding the GRLVQ. In order to update the reference vectors, the following criteria is maximized: $S = \sum_{i=1}^{N} f(\mu_\lambda(\mathbf{x}_i))$. In this formula, the relative distance $\mu_\lambda(\mathbf{x}_i) = \frac{D_{ij} - D_{ik}}{D_{ij} + D_{ik}}$ has values between -1 and 1, negative for correct classification and positive if the classification is not correct, according to the weighted distance.

The Supervised Relevance NG (SRNG) was obtained by including the NG idea in the GRLVQ algorithm [1]. The cost function optimized by this algorithm is:

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

with $\mu_\lambda(\mathbf{x}_i, \mathbf{w}_j) = \frac{|\mathbf{x}_i - \mathbf{w}_j|_\lambda^2 - D_{ik}}{|\mathbf{x}_i - \mathbf{w}_j|_\lambda^2 + D_{ik}}$, where D_{ik} is the weighted distance between $\mathbf{x}_i$ and the closest reference vector that does not belong to $W^{\mathbf{x}_i}$. According to this cost function, all reference vectors from $W^{\mathbf{x}_i}$ and the closest reference vector that does not belong to this set are updated by [1]:

$$\Delta \mathbf{w}_j = \eta \lambda \mathbf{I} \frac{\partial f}{\partial \mu} \frac{D_{ik}}{(|\mathbf{x}_i - \mathbf{w}_j|_\lambda^2 + D_{ik})^2} \cdot \cdot (\mathbf{x}_i - \mathbf{w}_j) \frac{r_j(\mathbf{x}_i, W^{\mathbf{x}_i})}{C(\gamma, K^{\mathbf{x}_i})} \quad (1)$$

where $\mathbf{w}_j$ is the closest reference vector from $\mathbf{x}_i$ that does not belong to $W^{\mathbf{x}_i}$, and

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

for all reference vectors from $W^{\mathbf{x}_i}$. η and η_1 are two positive constants.

In our experiments, we have used the sigmoid function $f(\mu) = \frac{1}{1+e^{-\mu\epsilon}}$, for which $\frac{\partial f}{\partial \mu} = f(\mu)\,(1 - f(\mu))$, with ϵ a positive constant.

III. Informational energy for feature ranking

For a discrete random variable X with probabilities p_k, $k = 1, \ldots, n$, the discrete informational energy was introduced by Onicescu [17] as $E(X) = \sum_{k=1}^{n} p_k^2$. For a continuous random variable Y, the informational energy is defined by [19]:

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

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

The conditional informational energy for a continuous random variable Y and a discrete random variable C is defined as

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

The following measure of unilateral dependency between two random variables X and Y was defined in [20]:

$$o(Y, X) = E(Y|X) - E(Y)$$

with the following properties:

- o is not symmetrical with respect to its arguments;
- $o(Y, X) \geq 0$ and the equality holds iff Y and X are independent;
- $o(Y, X) \leq 1 - E(Y)$ and the equality holds iff Y is completely dependent on X.

This measure can be regarded as an indicator of the unilateral dependence characterizing Y with respect to X and corresponds to the amount of information that X has about Y.

The mutual information $I(Y, X) = H(Y) - H(Y|X)$ and the value $o(Y, X) = E(Y|X) - E(Y)$ actually measure the same phenomenon and there is an obvious similarity between them. The measure o is unilateral and not mutual, but this does not influence our approximation procedure.

We will briefly describe a method to obtain the relevance values by maximizing $o(Y, C)$. Details can be found in [16]. Let us consider a continuous random variable Y with its samples $\mathbf{y}_i, i = 1, \ldots, N$:

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

where λ is the relevance vector, $\mathbf{x}_i, i = 1, \ldots, N$, is the set of training vectors each of them belonging to one of the $c_1, c_2, \ldots, c_M$ classes and $\mathbf{w}_j, j = 1, \ldots, P$, are the prototypes of the classes and are determined with a LVQ-type algorithm. The reason behind the choice of this transform is the connection it makes between the input vector and the class, represented by prototype $\mathbf{w}_j$, that it is assigned to. We consider the M classes labels are samples of a discrete random variable denoted by C.

The update of the relevance values can be written as:

$$\lambda^{(t+1)} = \lambda^{(t)} + \alpha \sum_{i=1}^{N} \frac{\partial o(Y, C)}{\partial \mathbf{y}_i} \mathbf{I}\,(\mathbf{x}_i - \mathbf{w}_j)\,. \quad (3)$$

From the definition, we have:

$$o(Y, C) = E(Y|C) - E(Y)$$

and we obtain:

$$o(Y, C) = \sum_{p=1}^{M} \frac{1}{p(c_p)} \int_{\mathbf{y}} p^2(\mathbf{y}, c_p) d\mathbf{y} - \int_{\mathbf{y}} p^2(\mathbf{y}) d\mathbf{y}. \quad (4)$$

This expression involves a considerable computational effort and the probability densities from the integrals can be approximated by the Parzen windows estimation method. The multidimensional Gaussian kernel is [5]:

$$G(\mathbf{y}, \sigma^2 \mathbf{I}) = \frac{1}{(2\pi)^{\frac{d}{2}} \sigma^d} \cdot e^{-\frac{\mathbf{y}^t \mathbf{y}}{2\sigma^2}},$$

where d is the dimension of the definition space of the kernel and $\sigma^2 \mathbf{I}$ is the covariance matrix. The probability density $p(\mathbf{y})$ is:

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

where I is the identity matrix.

We will denote by M_p the number of training samples from class c_p. Using the Parzen windows approximations, we have:

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

and

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

where $\mathbf{y}_{pk}$, $\mathbf{y}_{pl}$ are two training samples from class p. $\mathbf{y}_k$, $\mathbf{y}_l$ represent two training samples from any class.

With these relations, equation (4) can be rewritten and we finally obtain the following update formula of the relevance factors:

$$\lambda^{(t+1)} = \lambda^{(t)} - \alpha \frac{1}{4\sigma^2} G(\mathbf{y}_1 - \mathbf{y}_2, 2\sigma^2 \mathbf{I}) \cdot (\mathbf{y}_2 - \mathbf{y}_1) \mathbf{I} \cdot \cdot(\mathbf{x}_1 - \mathbf{w}_{j(1)} - \mathbf{x}_2 + \mathbf{w}_{j(2)}), \quad (5)$$

where $\mathbf{w}_{j(1)}$ and $\mathbf{w}_{j(2)}$ are the closest prototypes to $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively.

It is perhaps of interest to mention why we became interested in using a relatively old and simple information measure like the informational energy at all. As always with L^2 type methods, quadratic optimization functions lead to linear gradients and thus simpler computations. This was the first reason why we choosed the informational energy and not another type of informational measure in our approach. The second reason was the fact that we have previously introduced and studied the measure o [20].

Compared to the Renyi's quadratic entropy based estimation procedure [5], our method leads to a similar MI approximation formula, but in a more simple way. It is also interesting to note that only $o(Y, C)$ can be maximized like this. The similar procedure for maximizing $o(C, Y)$ does not work. This is because the measure o is not symmetrical.

IV. The ESRNG Algorithm

The algorithm adapts the reference vectors for as least as possible quantization error on all feature vectors. After initializing the relevance vector $\lambda_k = 1/n$, $k = 1, \ldots, n$, the codebook vectors, η, α, and σ, the following procedure updates incrementally the codebook vectors, the relevances and the feature ranks, for a given input $\mathbf{x}_i$:

1) Update the codebook vectors using the SRNG relations (1) and (2).
2) Update the relevances according to our formula (5) and normalize them.
3) Normalize the relevances.
4) Update the overall rank of each feature as an average over all previous steps.

Relevance determination can be used after LVQ learning, or simultaneously, this second version yielding an on-line algorithm.

V. Experiments and Results

We tested the ESRNG algorithm on three standard databases selected from [21]: Iris, Vowel Recognition, and Ionosphere. We compared our experimental results with experiments performed under similar conditions.

Tables I, II and III present the ranking of the input features obtained by our algorithm for Iris, Vowel and Ionosphere databases. In Table IV we list the recognition rates obtained by ESRNG compared to RLVQ, GRLVQ, SRNG, ERLVQ and EGRLVQ.

The problem of detecting the classes of the 150 vectors from the Iris database was tested on 6 reference vectors. The third component was ranked as most important, while the least important was the second component. We used $\eta = 1$, $\eta_1 = 0.5$, and the recognition rate was 97.33%.

For the Vowel recognition database (Deterding data) we trained 59 reference vectors and we obtained a recognition accuracy of 47.61%, with $\eta = 0.7$ 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, as we reported in [16]. We have obtained the relevance vector [0.369 0.390 0.364 0.322 0.290 0.220 0.218 0.338 0.337 0.254].

In the Ionosphere dataset test we used 8 reference vectors trained with 200 instances out of 351. The remaining 151 instances where used for testing phase, as specified in [21]. The recognition rate was 94.03%, with $\eta = 0.04$ and $\eta_1 = 0.03$.

TABLE I

Feature ranking for 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 II

Feature ranking for the Vowel Recognition dataset.

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

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

VI. Conclusions

ESRNG is an incremental learning algorithm for feature ranking and supervised classification. It is computationally attractive for large datasets, where dimensionality reduction is required. The ESRNG algorithm was tested on different standard datasets. Further experiments are necessary especially for comparing ESRNG and SRNG. Our present assumption is that using an informational theory approach for approximating the input feature relevances is especially profitable for feature ranking and selection.

TABLE III

Feature ranking for the Ionosphere database. Only the five most important features are represented.

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

TABLE IV

COMPARATIVE RECOGNITION RATES FOR THE TEST DATA.

	Iris	Vowel	Ionosphere
LVQ	91.33%	44.80%	90.06%
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%

REFERENCES

[1] B. Hammer, M. Strickert, and T. Villmann, "Supervised neural gas with general similarity measure," *Neural Process. Lett.*, vol. 21, no. 1, pp. 21–44, 2005.

[2] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Networks*, vol. 5, pp. 537–550, 1994.

[3] N. Kwak and C.-H. Choi, "Input feature selection for classification problems," *IEEE Trans. Neural Networks*, vol. 13, pp. 143–159, 2002.

[4] 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.

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

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

[7] N. Kwak and C.-H. Choi, "Input feature selection by mutual information based on parzen window," *IEEE Trans. PAMI*, vol. 24, no. 12, pp. 1667–1671, 2002.

[8] M. Tesmer and P. Estévez, "AMIFS: adaptive feature selection by using mutual information," in *IEEE International Joint Conference on Neural Networks IJCNN 2004, Budapest, Hungary, July 26-29*, 2004, pp. 303–308.

[9] T. Kohonen, *Self-Organizing Maps*. Springer Verlag, 1997.

[10] ——, "Improved versions of learning vector quantization," in *Proc. Int. Joint Conf. on Neural Networks*, San Diego, 1990, pp. 545–550.

[11] T. Bojer, B. Hammer, D. Schunk, and K. von Toschanowitz, "Relevance determination in learning vector quantization," in *Proceedings of the European Symposium on Artificial Neural Networks (ESANN 2001)*, M. Verleysen, Ed., D-side publications, 2001, pp. 271–276.

[12] B. Hammer and T. Villmann, "Generalized relevance learning vector quantization," *Neural Networks*, vol. 15, pp. 1059–1068, 2002.

[13] A. Sato and K. Yamada, "Generalized learning vector quantization," in *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, Eds., vol. 7, MIT Press, 1995, pp. 423–429.

[14] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten, "Neural-gas network for vector quantization and its application to time-series prediction," *IEEE Trans. Neural Networks*, vol. 4, pp. 558–569, 1993.

[15] R. Andonie and A. Cataron, "An informational energy LVQ approach for feature ranking," in *The European Symposium on Artificial Neural Networks (ESANN 2004), Bruges, Belgium, April 28-30*, M. Verleysen, Ed., D-side publications, 2004, pp. 471–476.

[16] A. Cataron and R. Andonie, "Energy generalized lvq with relevance factors," in *IEEE International Joint Conference on Neural Networks IJCNN 2004, Budapest, Hungary, July 26-29*, 2004, pp. 1421–1426.

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

[18] T. Villmann, F. Schleif, and B. Hammer, "Supervised neural gas and relevance learning in learning vector quantization," in *SelfOrganization of AdaptiVE behavior (SOAVE)*, Workshop presentation, Ilmenau, September 28–30, 2004, Germany, 2004.

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

[20] R. Andonie and F. Petrescu, "Interacting systems and informational energy," *Foundation of Control Engineering*, vol. 11, pp. 53–59, 1986.

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