Energy Generalized LVQ with Relevance Factors

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

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

***Abstract*— Input feature ranking and selection represent a necessary preprocessing stage in classification, especially when one is required to manage large quantities of data. We introduce a weighted generalized LVQ algorithm, called Energy Generalized Relevance LVQ (EGRLVQ), based on the Onicescu's informational energy [1]. EGRLVQ is an incremental learning algorithm for supervised classification and feature ranking.**

I. Introduction

Kohonen introduced Learning Vector Quantization (LVQ) [2] as a prototype based supervised clustering algorithm. This algorithm can be used as a simple, universal and efficient adaptive classifier. The idea of LVQ is to approximate optimal Bayesian decision borders between different classes in an n-dimensional feature space with a number of labeled prototypes named *codebook vectors*. An example vector $\mathbf{x}$ is classified to the label of its closest codebook vector according to a distance function as, for instance, the Euclidean distance. The codebook vectors are trained iteratively, using a set of training data. Various modifications to the basic algorithm were proposed to ensure a faster convergence (OLVQ), a better adaptation of the borders (LVQ2.1, LVQ3), an adequate initialization of the codebook vectors according to the data distribution (LVQ+SOM) [2], [3], or an adaptation for complex data structures [4], to name just a few.

Standard LVQ does not discriminate between more or less informative features: their influence on the distance function is equal. On the contrary, the Distinction Sensitive Learning Vector Quantizer (DSLVQ), introduced by Pregenzer *et al.* [5], [6], holds a changeable weight 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. The weighting process can be seen as a scaling transformation from the original Euclidean distance based feature space into a weighted distance based feature space: it increases distances between feature values of class discriminating features and decreases distances at little informative dimensions. This facilitates class discrimination and makes the system less sensitive to noise [6].

From Pregenzer's *et al.* approach we can extract the following ideas:

- The weighted distance function used by Pregenzer *et al.* is based on the weighted Euclidean approach. However, it is stated that other weighted distances could be applied as well.
- Weights and codebook vectors are updated in parallel during the learning phase. The updating can be performed on-line, after processing each learning sample. The modification of the LVQ algorithm is minimal: only the distance function has to be replaced.
- The resulted weights may be regarded as feature ranks and used for dimensional reduction of the input space (i.e., feature selection).

Another LVQ variation similar to DSLVQ is Relevance LVQ (RLVQ), introduced in [7], the weights being called *relevances*.

A different heuristic updating scheme of the feature weights, used for computing a modified distance in LVQ, is described in [8]. The weights (relevances) are computed as Ordered Weighted Aggregation (OWA) weights.

The above approaches share the following characteristic: they are only heuristically motivated and they do not obey a gradient dynamic. In DSLVQ and RLVQ, the updating of the weights is related to simple perceptron learning and difficulties arise if provided with non separable data [7].

For these reasons, a modification of RLVQ has been proposed by Hammer *et al.* [9], [10], Generalized RLVQ (GRLVQ), which obeys a stochastic gradient descent on an energy function. This method modifies the GLVQ algorithm (introduced in [11]) by using an adaptive metric, and leads to a more powerful classifier with little extra cost compared to GLVQ. GRLVQ is a large margin method with dimension independent mathematical generalization bounds [12] and it can be used together with any differentiable similarity measure [13].

Input feature selection is a necessary preprocessing stage in classification especially when one is required to manage large quantities of data. Recently, the mutual information (MI) was used by feature selection methods [14], [15]. 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. The subset should maximize MI. A sensible part of this approach is the estimation of MI. Estimating MI is hard because of the requirements for the conditional density functions and the high computational complexity. Many MI-based feature selection algorithms [14], [15] used histogram as density estimator. However, in high

dimensional space, histogram is neither effective nor accurate [16]. A MI estimation based on Renyi's quadratic entropy and Parzen windows estimation method was proposed by Principe *et al.* [17] and has been recently used in an efficient feature selection algorithm [16]. Torkkola also used this estimation method for feature extraction by mutual information maximization [18]. He computed the MI between transformed features and the class labels and used it as a criterion for learning the discriminative features transformations.

In this paper we introduce the Energy Generalized Relevance LVQ (EGRLVQ) algorithm which uses the estimation of the informational energy (IE) as a maximization criteria for the computing of the feature relevance. We then use a weighted distance based on the relevance factors instead of the Euclidian distance to redefine the GLVQ algorithm.

In Section II we introduce the basic notations used in LVQ and GLVQ, and define the class of "weighted GLVQ algorithms". A brief description of the informational energy is given in Section III. The relevance factors computation is developped in Section IV and the EGRLVQ algorithm is defined in Section V. We present in Section VI the experimental results and the conclusions are formulated in Section VII.

II. The Weighted GLVQ Algorithm

Assume that a clustering of data into M classes is to be learned and a set of training data is given:

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

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

LVQ chooses codebook vectors in $\mathbf{R}^n$ for each class. Denote the set of all codebook vectors by $\{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$. The components of a vector $\mathbf{w}_j$ are $[w_{j1}, \ldots, w_{jn}]$.

Let us consider we have two codebook vectors $\mathbf{w}_j$ and $\mathbf{w}_k$, the first one in the same class with the vector $\mathbf{x}_i$ and the second one from another class. We focus on LVQ2.1 [2] which is a clustering algorithm that adapts better the prototypes along the Bayes decision boundary. When the training process starts with properly defined initial values, the two codebook vectors can be updated in order to converge to the solution as follows:

$$\mathbf{w}_j^{(t+1)} = \mathbf{w}_j^{(t)} + \eta^{(t)}(\mathbf{x}_i^{(t)} - \mathbf{w}_j^{(t)})$$
$$\mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)} - \eta^{(t)}(\mathbf{x}_i^{(t)} - \mathbf{w}_k^{(t)})$$

where $\eta^{(t)}$ ranges between 0 and 1 and may decrease monotonically with time. The input vector $\mathbf{x}_i$ must fall into a window defined around the midplane of the two codebook vectors and the following condition must be sastisfied:

$$\min\left(\frac{d_j}{d_k}, \frac{d_k}{d_j}\right) > s$$

where $d_j = \|\mathbf{x}_i - \mathbf{w}_j\|$, $d_k = \|\mathbf{x}_i - \mathbf{w}_k\|$ and s is a constant determined by the window width l:

$$s = \frac{1-l}{1+l}.$$

A generalization of this algorithm, named Generalized LVQ (GLVQ) was proposed in [11]. A steepest descent method which minimizes a cost function was used to define the codebook vectors update.

A relative distance difference is defined by:

$$\mu(\mathbf{x}) = \frac{d_j - d_k}{d_j + d_k}.$$

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 we denoted the learning rate with η. Therefore, the update rule of the GLVQ algorithm is:

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

The weighted GLVQ algorithm adapts the codebooks for as least as possible quantization error on all feature vectors, as follows:

1) Find the two codebook vectors $\mathbf{w}_j$ and $\mathbf{w}_k$, the first one in the same class with the vector $\mathbf{x}_i$ and the second one from another class.
2) Update $\mathbf{w}_j$ and $\mathbf{w}_k$ according to the GLVQ algorithm.
3) Update the relevance factors.
4) Update the rank of each feature as an average of the ranks resulted in the previous steps.

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

III. Informational Energy

We will give in this section a brief description of the concept of informational energy as it was presented in [1].

The discrete informational energy was defined by Onicescu [1], and for a continuous random variable Y it 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.

When we have a continuous random variable Y and a discrete random variable C, the conditional informational energy is defined as following:

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

In order to study the interaction between two random variables X and Y, the following measure of unilateral dependency 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 detained by X about Y.

IV. Informational energy for feature ranking

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.

Let us consider $\mathbf{y}_i, i = 1, \ldots, N$ the samples of Y and

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

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 that 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.

We intend to obtain the relevance values by maximizing $o(Y, C)$:

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

We will denote by M_p the number of training samples from class c_p.

From the definition, we can write:

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

where

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

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

We obtain

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

This expression involves a considerable computational effort and we will aproximate the probability densities from the integrals using the Parzen windows estimation method. The multidimensional Gaussian kernel has the following form:

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

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 [17]:

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

where I is the unity matrix.

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, and $\mathbf{y}_k$, $\mathbf{y}_l$ represent two training samples from any class. With these relations, equation (2) can be rewritten and we finally obtain the following update formula of the relevance factors, as we explain in the Appendix:

$$\lambda^{(t+1)} = \lambda^{(t)} - \alpha \frac{1}{4\sigma^2} G(\mathbf{y}_1 - \mathbf{y}_2, 2\sigma^2 \mathbf{I}) \cdot (\mathbf{y}_2 - \mathbf{y}_1) \mathbf{I} \cdot$$
$$\cdot (\mathbf{x}_1 - \mathbf{w}_{j(1)} - \mathbf{x}_2 + \mathbf{w}_{j(2)}), \qquad (3)$$

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

We have obtained this formula using a gradient ascent method similar to the technique described by Torkkola in [18], but using informational energy instead of the mutual information.

V. EGRLVQ - Energy GRLVQ

We use the weighted distance between an input vector $\mathbf{x}_i$ and a codebook 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 must be reformulated to minimize an objective function based on this modified distance.

Let us consider the feature vector $\mathbf{x}_i$ and $\mathbf{w}_j$ to be its nearest codebook vector that belongs to the same class. We also consider $\mathbf{w}_k$ which is its nearest codebook vector that

belongs to a different class. We define the following relative distance:

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

which has values betwen -1 and 1, negative for correct classification and positive if the classification is not correct, according to weighted distance. Considering f as a non-linear, monotonically increasing function, the codebook vectors could be updated by a relation that minimizes the following criteria:

$$S = \sum_{i=1}^{N} f(\mu_\lambda(\mathbf{x}_i)).$$

In this paper we used the sigmoid function

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

for which

$$\frac{\partial f}{\partial \mu} = f(\mu, \gamma)\left(1 - f(\mu, \gamma)\right).$$

We obtain a modified GLVQ rule, which is just the GRLVQ rule [10]:

$$\Delta \mathbf{w}_j = \eta\lambda\mathbf{I}\frac{\partial f}{\partial \mu}\frac{D_{ik}}{(D_{ij} + D_{ikj})^2}(\mathbf{x}_i - \mathbf{w}_j) \tag{4}$$

if $\mathbf{x}_i$ and $\mathbf{w}_j$ are in the same class, and

$$\Delta \mathbf{w}_k = -\eta\lambda\mathbf{I}\frac{\partial f}{\partial \mu}\frac{D_{ij}}{(D_{ij} + D_{ik})^2}(\mathbf{x}_i - \mathbf{w}_k) \tag{5}$$

if $\mathbf{x}_i$ and $\mathbf{w}_k$ are in different classes.

The following procedure updates on-line both the relevances and the feature ranks.

1) Initialize η and α. Initialize the relevance vector: $\lambda_k = \frac{1}{n}$, $k = 1, \ldots, n$.
2) Initialize the codebook vectors.
3) Update the codebook vectors using the relations (4) and (5).
4) Update the relevances using formula (3).
5) Normalize the relevances.
6) Compute the weight of each feature as an average of its before ordering position index in the input vector, for all previous steps.
7) Repeat steps 3-6 for each training pattern.

LVQ2.1 is an enhanced version of the LVQ1 algorithm. In a previous paper [21], we have introduced a LVQ1 version of the above algorithm, called ERLVQ.

VI. Experiments and Results

We have used the following datasets on tests of EGRLVQ algorithm: Iris, Vowel Recognition, and Ionosphere.

The Iris database [22] consists of 3 classes, 50 vectors each. Two of them are not linearly separable. The problem is to detect the classes based on 4 features. While training 6 codebooks with EGRLVQ, we obtained a recognition rate of 97.33%. The relevance vector that resulted after the experiment presented here was [0.81 0.18 0.47 0.28]. We used $\eta = 0.3$ and $\alpha = 20$. In Table I we present the ranking resulted after the RLVQ, GRLVQ, OWA-RLVQ, ERLVQ, and EGRLVQ training, obtained for the test data, using the same initial set of codebook vectors. The results obtained with ERLVQ and EGRLVQ are different from RLVQ and OWA-RLVQ due to the different approach involved in the developing of our new algorithm. When we use Iris database, the last feature is the most important for algorithms which find well defined classes' centroids, such as RLVQ and OWA-RLVQ. When our goal is to define as much as possible class discrimination to the neighbourhood of the borders with an algorithm such as ERLVQ and EGRLVQ, the first feature is the most important. Our ranking is similar to the results reported in [23] where the bidimensional projection of patterns from classes "1" and "2" including the first feature leads us to two well delimited clusters, but the centroids are relatively close one to the other. On the other hand, the bidimensional projection of the same patterns based on the last two features creates two clusters with quite distanced centroids, but the classes delimitation is not very clear.

TABLE I

Feature ranking for the Iris database.

Rank	1	2	3	4
RLVQ	4	2	3	1
GRLVQ	4	3	2	1
OWA-RLVQ	4	3	2	1
ERLVQ	1	2	3	4
EGRLVQ	1	3	4	2

The second dataset that we used in our tests was the Vowel Recognition database (Deterding data) [22]. It contains vectors extracted from 15 individual speakers pronouncing vowels in 11 contexts, 6 times each. The problem is to use the pronunciations of the first 8 speakers for training and the pronunciations of the remaining 7 speakers for tests. We trained 59 codebooks and the recognition score was 47.18%. In the EGRLVQ experiments we set $\eta = 0.7$ and $\alpha = 200$. The ranking that we obtained for the 10 features of the test vectors from this dataset is presented in Table II. The most important resulted to be the feature number 3. The second feature was selected as one of the most important, as resulted when we used other algorithms. The features from the positions 7 and 10 are the least important for EGRLVQ, also confirming the previous results. The relevance vector that we obtained in the experiment presented here is [0.354 0.349 0.439 0.287 0.299 0.310 0.268 0.271 0.284 0.253].

The last set of tests was performed in the Ionosphere dataset [22]. It consists of 351 instances of radar collected data, with 34 continuous attributes each. The vectors are labelled with "bad" or "good", being a binary classification task. The first 200 instances that are balanced between positive and negative examples were used for the training of 8 codebook vectors. The remaining 151 patterns were used for tests. The values of the EGRLVQ training parameters were $\eta = 0.9$ and $\alpha = 0.9 \cdot 10^{13}$. We obtained a recognition rate of 94.40% that was better than the rates obtained with GRLVQ and ERLVQ. Table

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
OWA-RLVQ	8	2	4	5	6
ERLVQ	2	1	3	4	6
EGRLVQ	3	1	2	6	5
Rank	6	7	8	9	10
RLVQ	3	4	8	7	10
GRLVQ	1	9	7	8	10
OWA-RLVQ	9	3	7	1	10
ERLVQ	8	9	5	10	7
EGRLVQ	4	9	8	7	10

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
OWA-RLVQ	14	12	1	3	28
ERLVQ	8	24	16	12	6
EGRLVQ	4	5	12	8	27

III contains the ranking of the most important 5 features of the test vectors.

We finally listed in Table IV a comparative set of recognition rates that we obtained with LVQ, RLVQ, GRLVQ, OWA-RLVQ, ERLVQ and EGRLVQ in similar conditions, for the test data.

VII. CONCLUSIONS

EGRLVQ is an incremental learning algorithm for feature ranking and supervised classification. It is computational attractive for large datasets, where dimensionality reduction is required. The EGRLVQ algorithm was sucessfully tested on different standard datasets and, compared to RLVQ, GRLVQ, OWA-RLVQ, and ERLVQ, provided better recognition rates.

The behaviour of EGRLVQ around the boundaries of the receptive fields is an interesting future research direction.

APPENDIX

The relevance updating formula. The equation (2) can be rewritten as following:

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

that is

$$o(Y,C)=\frac{1}{N}\left(\sum_{p=1}^{M}\frac{1}{M_p}\right)\sum_{k=1}^{M_p}\sum_{l=1}^{M_p}G(\mathbf{y}_{pk}-\mathbf{y}_{pl},2\sigma^2\mathbf{I})-$$
$$-\frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N}G(\mathbf{y}_k-\mathbf{y}_l,2\sigma^2\mathbf{I}). \quad (6)$$

To find the relevance update formula, we will use two consecutive samples as classes representatives $\mathbf{y}_1$ and $\mathbf{y}_2$, as was suggested in [24]:

1) When the two samples belong to the same class, $N=2$, $M=2$, $M_1=2$ and $M_2=0$, and the first term of relation (6) cannot be calculated due to the non-determination introduced by the value of M_2 from the denominator. Therefore, this case will be ignored.
2) When the two samples belong to different classes, $N=2$, $M=2$, $M_1=1$ and $M_2=1$. Equation (6) becomes:

$$o(Y,C)=$$
$$=\frac{1}{2}\left(\sum_{p=1}^{2}\frac{1}{M_p}\right)\sum_{k=1}^{M_p}\sum_{l=1}^{M_p}G(\mathbf{y}_{pk}-\mathbf{y}_{pl},2\sigma^2\mathbf{I})-$$
$$-\frac{1}{4}\sum_{k=1}^{2}\sum_{l=1}^{2}G(\mathbf{y}_k-\mathbf{y}_l,2\sigma^2\mathbf{I})=$$
$$=\frac{1}{2}\left[G(\mathbf{y}_{11}-\mathbf{y}_{11},2\sigma^2\mathbf{I})+G(\mathbf{y}_{21}-\mathbf{y}_{21},2\sigma^2\mathbf{I})\right]$$
$$-\frac{1}{4}\left[G(\mathbf{y}_1-\mathbf{y}_1,2\sigma^2\mathbf{I})+G(\mathbf{y}_1-\mathbf{y}_2,2\sigma^2\mathbf{I})+\right.$$
$$\left.+G(\mathbf{y}_2-\mathbf{y}_1,2\sigma^2\mathbf{I})+G(\mathbf{y}_2-\mathbf{y}_2,2\sigma^2\mathbf{I})\right]=$$
$$=G(0,2\sigma^2\mathbf{I})-\frac{1}{2}\left[G(0,2\sigma^2\mathbf{I})+G(\mathbf{y}_1-\mathbf{y}_2,2\sigma^2\mathbf{I})\right].$$

Finally, the relation will be:

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

According to (2), we must calculate the partial derivatives of $o(Y,C)$ with respect to $\mathbf{y}_1$ and then to $\mathbf{y}_2$. For this purpose, we use the following relation:

$$\frac{\partial}{\partial\mathbf{y}_i}G(\mathbf{y}_i-\mathbf{y}_j,2\sigma^2\mathbf{I})=$$
$$=G(\mathbf{y}_i-\mathbf{y}_j,2\sigma^2\mathbf{I})\frac{\mathbf{y}_j-\mathbf{y}_i}{2\sigma^2}=$$
$$=-\frac{\partial}{\partial\mathbf{y}_j}G(\mathbf{y}-i-\mathbf{y}_j,2\sigma^2\mathbf{I}).$$

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%
OWA-RLVQ	96.66%	46.75%	93.37%
ERLVQ	97.33%	47.18%	94.03%
EGRLVQ	97.33%	47.18%	94.40%

Then:

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

Because

$$G(0,2\sigma^2\mathbf{I}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\sigma|^{\frac{1}{2}}}$$

is a constant value, we have:

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

Using a similar method, we can write:

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

Replacing these results in (1), we obtain:

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

Hence,

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

or

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

REFERENCES

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

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

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

[4] P. Somervuo and T. Kohonen, "Self-organizing maps and learning vector quantization for feature sequences," *Neural Processing Letters*, vol. 10, pp. 151–159, 1999.

[5] M. Pregenzer, D. Flotzinger, and G. Pfurtscheller, "Distinction sensitive learning vector quantization - a noise-insensitive classification method," in *Proc. IEEE Int. Joint Conf. on Neural Networks, ICNN-94*, Orlando, Florida, 1994, pp. 2890–2894.

[6] M. Pregenzer, G. Pfurtscheller, and D. Flotzinger, "Automated feature selection with a distinction sensitive learning vector quantizer," *Neurocomputing*, vol. 11, pp. 19–29, 1996.

[7] 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.

[8] A. Cataron and R. Andonie, "RLVQ determination using OWA operators," in *Proceedings of the Third IASTED International Conference on Artificial Intelligence and Applications (AIA 2003), Benalmadena, Spain, September 8-10, 2003*, M. Hamza, Ed., ACTA Press, 2003, pp. 434–438.

[9] B. Hammer and T. Villmann, "Estimating relevant input dimensions for self-organizing algorithms," in *Advances in Self-Organizing Maps*, N. Allinson, H. Yin, L. Allinson, and J. Slack, Eds., Springer Verlag, 2001, pp. 173–180.

[10] ——, "Generalized relevance learning vector quantization," *Neural Networks*, vol. 15, pp. 1059–1068, 2002.

[11] 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.

[12] B. Hammer, M. Strickert, and T. Villmann, "On the generalization ability of GRLVQ networks," *Osnabrcker Schriften zur Mathematik, Preprint, no. 249, 10/2003*, 2003.

[13] ——, "Supervised neural gas with general similarity measure," *to appear in Neural Processing Letters*, 2004.

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

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

[16] 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.

[17] 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.

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

[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] 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.

[22] 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

[23] R. De, N. Pal, and S. Pal, "Feature analysis: Neural network and fuzzy set theoretic approaches," *Patterns Recognition*, vol. 30, pp. 1579–1590, 1997.

[24] K. Torkkola and P. Daly, "On feature extraction by mutual information maximization," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Orlando, Florida*, 2002.