Informational Energy Kernel for LVQ

Angel Caţaron[1] and Răzvan Andonie[2]

[1] Transylvania University of Brasov, Romania
[2] Central Washington University, Ellensburg, USA

Abstract. We describe a kernel method which uses 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. Our quadratic optimization function, as an L^2 type method, leads to linear gradient and thus easier computation. We obtain an approximation formula similar to the mutual information based method, but in a more simple way.

1 Introduction

Relevance LVQ (RLVQ) [2] uses a weighted distance function for the LVQ classification. A modification of RLVQ has been proposed by Hammer *et al.* [3], Generalized RLVQ (GRLVQ), which obeys a stochastic gradient descent on an energy function.

The neural-gas (NG) algorithm [4] represents a neural model which is applied to the task of vector quantization by using a neighborhood cooperation scheme. The NG network uses an adaptation rule similar to the Kohonen feature map. It replaces the Euclidian distance with the neighborhood ranking of the reference vectors for a given input vector. The Supervised Relevance Neural Gas (SRNG) algorithm [1] combines the NG and the GRLVQ. 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 we have introduced two LVQ classificators based on Onicescu's informational energy (IE): the Energy RLVQ (ERLVQ) [5] and the Energy GRLVQ (EGRLVQ) [6]. We have obtained incremental learning algorithms for feature ranking and supervised classification. The sensible part of such an approach is the mutual information estimation, which poses great difficulties as it requires the knowledge on the underlying probability density functions of the data space and the integration on these functions [13]. Our technique proved to be an efficient solution to this problem.

In this paper, we describe the Energy SRNG (ESRNG) classificator, a kernel method which uses the maximization of the IE as a criteria for computing the relevances of input features. This adaptive relevance determination is used in combination with the SRNG model, providing an alternative way for determining the relevances. After introducing the SRNG notations and the relevance determination using IE, we define the ESRNG algorithm and compare it to other algorithms of this family.

W. Duch et al. (Eds.): ICANN 2005, LNCS 3697, pp. 601–606, 2005.

2 SRNG

Assume that a clustering of data into M classes, $c_1, \ldots, c_M$, is implemented and a set of training data is available: $X = \{(\boldsymbol{x}_i, c_i) \subset \mathbb{R}^n \times \{1, \ldots, M\} \mid i = 1, \ldots, N\}$. The training vectors $\boldsymbol{x}_i$ have n components $[x_{i1}, \ldots, x_{in}]$. A subset of reference vectors from $\mathbb{R}^n$ are assigned to each class. Denote the set of all reference vectors by $W = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K\}$. The components of a vector $\boldsymbol{w}_j$ are $[w_{j1}, \ldots, w_{jn}]$.

The NG algorithm optimizes a cost function which uses the rank $r_j(\boldsymbol{x}_i, W)$ of the reference vector $\boldsymbol{w}_j$ for a given input $\boldsymbol{x}_i$ [1], [4]:

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

where $h_\gamma(r_j(\boldsymbol{x}_i, W)) = e^{-r_j(\boldsymbol{x}_i, W)/\gamma}$, $C(\gamma) = \sum_{r=0}^{K-1} h_\gamma(r)$, and γ is a parameter which gives the neighborhood range. The rank $r_j(\boldsymbol{x}_i, W)$ of the reference vector $\boldsymbol{w}_j$ for the input vector $\boldsymbol{x}_i$ is the number of reference vectors that are in the relation $\|\boldsymbol{x}_i - \boldsymbol{w}_k\| \leq \|\boldsymbol{x}_i - \boldsymbol{w}_j\|$, where $j, k \in \{1, \ldots, K\}$ and $j \neq k$. The neighborhood ranking of the reference vectors is updated each time a training vector is applied to the input of the neural network.

The GRLVQ algorithm uses a squared weighted distance between an input vector $\boldsymbol{x}_i$ and a reference vector $\boldsymbol{w}_j$, $D_{ij}^2 = \sum_{k=1}^n \lambda_k (x_{ik} - w_{jk})^2$, where $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]$ is the relevance vector, with $\lambda_i \geq 0$, $i = 1, \ldots, n$, $\sum_{i=1}^n \lambda_i = 1$. The Supervised Relevance NG (SRNG) can be obtained [1] by including the NG idea in the GRLVQ algorithm. The cost function optimized by this algorithm is:

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

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

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

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

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

for all reference vectors from $W^{\boldsymbol{x}_i}$. In these relations, η and η_1 are two positive constants. We 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.

3 Relevance Determination Using Informational Energy

Onicescu's IE [7], [8] is defined by: $E(Y) = \int_{-\infty}^{+\infty} p^2(\boldsymbol{y})d\boldsymbol{y}$, where Y is a continuous random variable with probability density function $p(\boldsymbol{y})$. The conditional information energy between Y and a discrete random variable C is: $E(Y|C) = \int_{\boldsymbol{y}} \sum_{p=1}^{M} p(c_p)p^2(\boldsymbol{y}|c_p)d\boldsymbol{y}$.

The unilateral dependence measure $o(Y, X) = E(Y|X) - E(Y)$, defined in [9], quantifies the amount of information contained in random variable X about random variable Y.

The ESRNG algorithm uses a vector of relevances obtained by maximizing $o(Y, X)$ with an ascending gradient method [6]. A transformation which makes the connection between the input vector and the class represented by the reference vector $\boldsymbol{w}_j$ is employed: $\boldsymbol{y}_i = \boldsymbol{\lambda I}(\boldsymbol{x}_i - \boldsymbol{w}_j)$. In this equation, $\boldsymbol{x}_i, i = 1, \ldots, N$, is the set of training vectors that belong to one of the $c_1, c_2, \ldots, c_M$ classes; $\boldsymbol{w}_j, j = 1, \ldots, P$, are the reference vectors of the classes; $\boldsymbol{\lambda}$ is the vector of relevances; $\boldsymbol{I}$ is the unity matrix. The values $\boldsymbol{y}_i, i = 1, \ldots, N$, are samples of the random variable Y.

We obtain the relevance values by an iteratively updating approach:

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

Considering the M class labels as samples of a discrete random variable denoted by C, we have: $o(Y, C) = E(Y|C) - E(Y)$. The conditional information energy can be reformulated as a dependence of the squared mutual probability density $E(Y|C) = \sum_{p=1}^{M} p(c_p) \int_{\boldsymbol{y}} p^2(\boldsymbol{y}|c_p)d\boldsymbol{y} = \sum_{p=1}^{M} \frac{1}{p(c_p)} \int_{\boldsymbol{y}} p^2(\boldsymbol{y}, c_p)d\boldsymbol{y}$.

This allows us to write $o(Y, C) = \sum_{p=1}^{M} \frac{1}{p(c_p)} \int_{\boldsymbol{y}} p^2(\boldsymbol{y}, c_p)d\boldsymbol{y} - \int_{\boldsymbol{y}} p^2(\boldsymbol{y})d\boldsymbol{y}$, which can easily estimated by using the Parzen windows with the Gaussian kernel $G(\boldsymbol{y} - \boldsymbol{y}_i, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{\|\boldsymbol{y}-\boldsymbol{y}_i\|^2}{2\sigma}}$.

The probability density $p(\boldsymbol{y})$ can be expressed [10] as $p(\boldsymbol{y}) = \frac{1}{N}\sum_{i=1}^{N} G(\boldsymbol{y} - \boldsymbol{y}_i, \sigma^2)$. We can write: $\int_{\boldsymbol{y}} p^2(\boldsymbol{y}, c_p)d\boldsymbol{y} = \frac{1}{N^2}\sum_{k=1}^{N_p}\sum_{l=1}^{N_p} G(\boldsymbol{y}_{pk} - \boldsymbol{y}_{pl}, 2\sigma^2)$ and $\int_{\boldsymbol{y}} p^2(\boldsymbol{y})d\boldsymbol{y} = \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N} G(\boldsymbol{y}_k - \boldsymbol{y}_l, 2\sigma^2)$, where $\boldsymbol{y}_{pk}$, $\boldsymbol{y}_{pl}$ are two training samples from class p, and $\boldsymbol{y}_k$, $\boldsymbol{y}_l$ are two training samples from any class. N_p is the number of the training samples from the class p.

We obtain

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

We use two consecutive samples $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ as classes representatives. This expression can only be evaluated when the two training vectors belong to different classes. In this case, we obtain:

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

4 The ESRNG as a Kernel Based Algorithm

When $\boldsymbol{y}_1 \neq \boldsymbol{y}_2$, we have $\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|^2 > 0$ and $G(0, 2\sigma^2) > G(\boldsymbol{y}_1 - \boldsymbol{y}_2, 2\sigma^2)$. This means $o(Y, C) > 0$ for all input vectors. Hence, this is a positive defined kernel.

The squared weighted distance between an input vector $\boldsymbol{x}_i$ and a reference vector $\boldsymbol{w}_j$ $D_{ij}^2 = \sum_{k=1}^{n} \lambda_k (x_{ik} - w_{jk})^2$ requires that $\lambda_k \geq 0$ for all $k = 1, \ldots, n$. In the case when at least one relevance value is negative, this condition can be realized by transforming the relevance vectors with $\lambda_k = \frac{e^{\lambda_k}}{\sum_{i=1}^{n} e^{\lambda_i}} + \epsilon$ or by scaling the relevance components $\lambda_k = \lambda_k + \min_{i=1,\ldots,n} \lambda_i + \epsilon$, where ϵ is a positive constant. We usually apply a transform of the relevance vector in order to keep its component's values in a reasonable domain.

Finally, we obtain:

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

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

The ESRNG 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 (3) and transform them.
3. Update the overall rank of each feature as an average over all previous steps.

Since we also obtain a ranking of the input vectors' components, this algorithm can be used not only in classification tasks, but also in feature selection.

The weighted Euclidean metric we use allows for a direct interpretation as kernelized NG if the relevances are fixed [1]. In this case, the relevances should not be updated after processing each input pattern. This may be achieved if we allow a preprocessing of the patterns, where the relevances are computed first.

5 Experiments

The classification results obtained by ESRGN, applied on three well known datasets (Iris, Ionosphere, and Vowel Recognition [11]), are compared in Table 1 with other experiments performed under similar conditions.

We used 6 reference vectors to classify the 150 vectors from the Iris database. The third component was ranked as most important and the least important was the second component, while the recognition rate was 97.33%. The 351 instances of the Ionosphere dataset were split into two subsets. For the first training 200 samples we used 8 reference vectors. The remaining 151 samples were used in the classification tests. We obtained a recognition rate of 94.40%. For the Vowel recognition database (Deterding data) we trained 59 reference vectors and we obtained a recognition accuracy of 47.61%. The second feature was found as most important, whereas the 7-th and 10-th features were ranked as the least important.

Figure 1 shows the average values of the feature relevances obtained with ESRNG experiments.

Table 1. 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%	94.40%

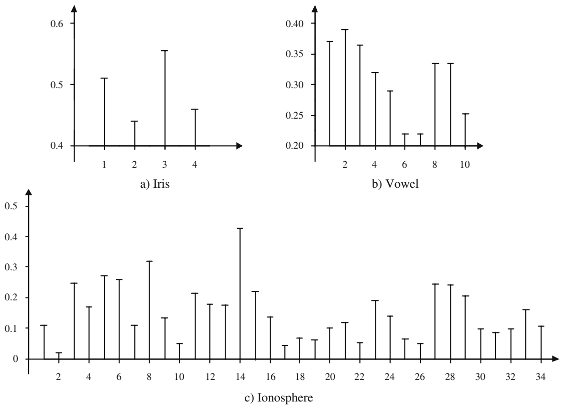

Fig. 1. The average values of the feature relevances obtained with ESRNG experiments

6 Conclusions

Our contribution is an information theory approximation of the relevances in the supervized NG algorithm. This method proves to be computationally effective and leads to good recognition rates.

Jenssen *et al.* [12] have recently proved that information theoretic learning based on Parzen windows density estimation is similar to kernel-based learning. Since the distance we use allows for a direct interpretation as kernelized NG, in our future work we will attempt to combine these two results.

References

1. Hammer, B. , Strickert, M. , Villmann, T.: Supervised neural gas with general similarity measure. Neural Process. Lett. **21** 1 (2005) 21-44
2. Bojer, T., Hammer, B., Schunk, D., von Toschanowitz, K.T.: Relevance Determination in Learning Vector Quantization. Proceedings of the European Symposium on Artificial Neural Networks (ESANN 2001) (2001) 271–276
3. Hammer, B., Villmann, T.: Generalized Relevance Learning Vector Quantization. Neural Networks **15** (2002) 1059–1068
4. Martinetz, T. M. , Berkovich, S. G. , Schulten, K. J.: Neural-gas network for vector quantization and its application to time-series prediction. IEEE Transactions on Neural Networks **4**(1993) 558-569
5. Andonie, R., Cataron, A.: An informational energy LVQ approach for feature ranking. Proceedings of the European Symposium on Artificial Neural Networks (ESANN 2004) (2004) 471–476
6. Cataron, A., Andonie R.: Energy generalized LVQ with relevance factors. Proceedings of the IEEE International Joint Conference on Neural Networks IJCNN 2004, Budapest, Hungary, July 26-29 (2004) 1421-1426
7. Onicescu, O.: Theorie de l'information. Energie informationelle. C. R. Acad. Sci. Paris, Ser. A-B **263** (1966) 841-842
8. Guiasu, S.: Information theory with applications. McGraw Hill New York (1977).
9. Andonie, R., Petrescu, F.: Interacting systems and informational energy. Foundation of Control Engineering **11** (1986) 53-59
10. Principe, J. C., Xu, D., Fisher III, J. W.: Information-theoretic learning. In Unsupervised Adaptive Filtering, S. Haykin, Wiley, New York (2000)
11. Blacke, K., Keogh, E., Merz, C. J.: UCI Repository of Machine Learning Databases. [Online]. Available: http://www.ics.uci. edu/∼mlearn/MLSummary.html (1998)
12. Jenssen, R., Erdogmus, D., Principe, J.C., Eltoft, T.: Towards a unification of information theoretic learning and kernel methods. IEEE Workshop on Machine Learning for Signal Processing, Sao Luis, Brazil, (2004)
13. Chow, T. W. S., Huang D.: Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information. IEEE Transactions on Neural Networks **16**(2005) 213–224