ESANN'2007 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2007, d-side publi., ISBN 2-930307-07-2.

Relevance Matrices in LVQ

Petra Schneider[1], Michael Biehl[1] and Barbara Hammer[2]

1- University of Groningen - Mathematics and Computing Science
P.O. Box 800, 9700 AV Groningen - The Netherlands

2- Clausthal University of Technology - Institute of Computer Science
Julius Albert Strasse 4, 38678 Clausthal-Zellerfeld - Germany

Abstract. We propose a new matrix learning scheme to extend Generalized Relevance Learning Vector Quantization (GRLVQ). By introducing a full matrix of relevance factors in the distance measure, correlations between different features and their importance for the classification scheme can be taken into account. In comparison to the weighted euclidean metric used for GRLVQ, this metric is more powerful to represent the internal structure of the data appropriately while maintaining its excellent generalization ability as large margin optimizer. The algorithm is tested and compared to alternative LVQ schemes using an artificial dataset and the image segmentation data from the UCI repository.

1 Introduction

Learning vector quantization (LVQ) as introduced by Kohonen constitutes a particularly intuitive and simple though powerful classification scheme [10] which is very appealing for several reasons: the method is easy to implement; the complexity of the resulting classifier can be controlled by the user; the classifier can naturally deal with multiclass problems; and, unlike many alternative neural classification schemes, the resulting classifier is human understandable. Original LVQ, however, suffers from several drawbacks such as slow convergence and instability. An exact investigation of these properties is quite complex [2] and a variety of alternatives have been proposed, see e.g. [10].

One major drawback consists in the dependency on the euclidean metric. A variety of extensions of metric based approaches such as k-nearest neighbor or k-means clustering to more general metrics including an adaptive diagonal metric or a full matrix exist [5, 6]. For LVQ type algorithms, generalized relevance learning vector quantization [9] constitutes a more powerful alternative which includes adaptive relevance factors into training. This allows to scale the axes, i.e. a better adaptation towards clusters with axes-parallel ellipsoidal shapes. Here we introduce a more general version which includes a full adaptive matrix for every prototype, i.e. the possibility to adapt to arbitrary local ellipsoids which correspond to locally correlated input dimensions. We show that this general method leads to efficient and powerful classifiers with excellent generalization ability, as substantiated by a theoretical counterpart as well as two experiments.

2 Generalized metric LVQ

LVQ aims at approximating a clustering by prototypes. Assume training data $(\boldsymbol{\xi}_i, y_i) \in \mathbb{R}^N \times \{1, \ldots, C\}$ are given, N denoting the data dimensionality and C the number of different classes. A LVQ network consists of a number of prototypes which are characterized by their location in the weight space $\boldsymbol{w}_i \in \mathbb{R}^N$ and their class label $c(\boldsymbol{w}_i) \in \{1, \ldots, C\}$. Classification takes place by a winner takes

all scheme. For this purpose, a (possibly parameterized) similarity measure d^λ is fixed for $\mathbb{R}^N$. Often, the standard euclidean metric is chosen. A data point $\boldsymbol{\xi} \in \mathbb{R}^N$ is mapped to the class label $c(\boldsymbol{\xi}) = c(\boldsymbol{w}_i)$ of the prototype i for which $d^\lambda(\boldsymbol{w}_i, \boldsymbol{\xi}) \leq d^\lambda(\boldsymbol{w}_j, \boldsymbol{\xi})$ holds for every $j \neq i$ (breaking ties arbitrarily).

Learning aims at determining weight locations for the prototypes such that the given training data are mapped to their corresponding class labels. A very flexible learning approach has been introduced in [9]. It is derived as a minimization of the cost function

$$\sum_i \Phi\left(\frac{d_J^\lambda - d_K^\lambda}{d_J^\lambda + d_K^\lambda}\right) \tag{1}$$

where Φ is a monotonic function, e.g. the identity or the logistic function, $d_J^\lambda = d^\lambda(\boldsymbol{w}_J, \boldsymbol{\xi}_i)$ is the distance of data point $\boldsymbol{\xi}_i$ from the closest prototype $\boldsymbol{w}_J$ with the same class label y_i, and $d_K^\lambda = d^\lambda(\boldsymbol{w}_K, \boldsymbol{\xi}_i)$ is the distance from the closest prototype $\boldsymbol{w}_K$ with a different class label than y_i. Taking the derivatives with respect to the prototypes and metric parameters yields the adaptation rules.

The choice of the similarity measure as standard euclidean metric yields GLVQ. The squared *weighted* euclidean metric $d^\lambda(\boldsymbol{w}, \boldsymbol{\xi}) = \sum_i \lambda_i (w_i - \xi_i)^2$ where $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$ constitutes a powerful alternative, GRLVQ, particularly suited for high dimensional data with a different (but priorly not known) relevance of the input dimensions. Thereby, the relevance factors need not be global, but they can be attached to a single prototype, i.e. individual updates take place for the relevance factors λ^j, and $d^{\lambda_j}(\boldsymbol{w}_j, \boldsymbol{\xi}_i)$ is computed based on λ_j. This method, localized GRLVQ (LGRLVQ), has been investigated in [7].

3 Generalized matrix LVQ

Here, we introduce a more general choice of the similarity measure, a full matrix, which can account for arbitrary correlations of the dimensions. The metric has the form

$$d^\Lambda(\boldsymbol{w}, \boldsymbol{\xi}) \;=\; (\boldsymbol{\xi} - \boldsymbol{w})^T \, \Lambda \, (\boldsymbol{\xi} - \boldsymbol{w})$$

where Λ is a full matrix. Note that, this way, arbitrary euclidean metrics can be achieved by an appropriate choice of the parameters. In particular ellipsoidal clusters which are not axes parallel can be obtained. Such choices have already successfully been introduced in several *unsupervised* clustering methods, e.g. [6].

Note that the above similarity measure only leads to a squared distance if Λ is positive (semi-) definite. We can achieve this by substituting $\Lambda = \Omega\,\Omega^T$. As Λ is symmetric, we can assume that Ω itself is symmetric: $\Omega = \Omega^T$. To obtain the adaptation formulas we need to compute the derivatives of (1) with respect to $\boldsymbol{w}$ and Λ. We get the updates

$$\begin{aligned}
\Delta \boldsymbol{w}_J &= +\epsilon \cdot \phi'(\mu(\boldsymbol{\xi})) \cdot \mu^+(\boldsymbol{\xi}) \cdot \Omega\Omega \cdot (\boldsymbol{\xi} - \boldsymbol{w}_J) \\
\Delta \boldsymbol{w}_K &= -\epsilon \cdot \phi'(\mu(\boldsymbol{\xi})) \cdot \mu^-(\boldsymbol{\xi}) \cdot \Omega\Omega \cdot (\boldsymbol{\xi} - \boldsymbol{w}_K) \\
\Delta \Omega_{lm} &= -\epsilon \cdot \phi'(\mu(\boldsymbol{\xi})) \cdot \\
&\quad \Big(\mu^+(\boldsymbol{\xi}) \cdot \Big([\Omega(\boldsymbol{\xi} - \boldsymbol{w}_J)]_m (\xi_l - w_{J,l}) + [\Omega(\boldsymbol{\xi} - \boldsymbol{w}_J)]_l (\xi_m - w_{J,m}) \Big) \\
&\quad - \mu^-(\boldsymbol{\xi}) \cdot \Big([\Omega(\boldsymbol{\xi} - \boldsymbol{w}_K)]_m (\xi_l - w_{K,l}) + [\Omega(\boldsymbol{\xi} - \boldsymbol{w}_K)]_l (\xi_m - w_{K,m}) \Big) \Big)
\end{aligned}$$

for the prototypes and matrix elements Ω_{lm} with $\mu(\boldsymbol{\xi}) = (d_J^\Lambda - d_K^\Lambda)/(d_J^\Lambda + d_K^\Lambda)$, $\mu^+(\boldsymbol{\xi}) = 2 \cdot d_K^\Lambda/(d_J^\Lambda + d_K^\Lambda)^2$, and $\mu^-(\boldsymbol{\xi}) = 2 \cdot d_J^\Lambda/(d_J^\Lambda + d_K^\Lambda)^2$. (See [3] for the derivation of these formulas.) Thereby, the learning rate for the metric can be chosen independently of the learning rate for the prototypes. Note that Ω is symmetric because these updates are symmetric. After each update, Λ is normalized to prevent the algorithm from degeneration. We set $\sum_i \Lambda_{ii} = \sum_{i,j} \Omega_{ij}^2 = 1$ which fixes the sum of diagonal elements and, here, the sum of eigenvalues. We term this learning rule generalized matrix LVQ (GMLVQ).

Note that we can work with one full matrix which accounts for a transformation of the whole input space, or, alternatively, with local matrices Λ^j attached to the individual prototypes $\boldsymbol{w}_j$, such that general local ellipsoidal clusters can be obtained. We refer to this general version as localized GMLVQ (LGMLVQ).

4 Generalization ability

One of the benefits of prototype-based learning algorithms is that they show very good generalization ability also for high dimensional data as proved in [4, 8]. The basic insight consists in the fact that large margin generalization bounds can be derived for LVQ networks similar to learning theoretical results derived for SVM. Since LVQ networks are universal approximators provided enough neuron, a good overall behavior is guaranteed. This argument can be transferred to the matrix version. We shortly sketch the result and refer to [3] for details.

We consider a LGMLVQ network given by P prototypes $\boldsymbol{w}_i$ with inputs $|\boldsymbol{\xi}| \leq B$ for some $B > 0$ (hence also $|\boldsymbol{w}_i| \leq B$) and the case of a binary classification, i.e. labels 1 or -1. Classification takes place by a winner takes all rule

$$\boldsymbol{\xi} \mapsto c(\boldsymbol{w}_i) \text{ where } (\boldsymbol{\xi} - \boldsymbol{w}_i)^T \Lambda^i (\boldsymbol{\xi} - \boldsymbol{w}_i) \leq (\boldsymbol{\xi} - \boldsymbol{w}_j)^T \Lambda^j (\boldsymbol{\xi} - \boldsymbol{w}_j) \, \forall j \neq i \qquad (2)$$

with positive semidefinite matrix Λ^i with $\sum_l \Lambda_{ll}^i = 1$. A network corresponds to a function in $\mathcal{F} := \{f : \mathbb{R}^N \to \{-1, 1\} \mid f \text{ is given by (2) for some } \Lambda^i, \boldsymbol{w}_i\}$.

Assume some unknown underlying probability measure P is given on $\mathbb{R}^N \times \{-1, 1\}$. The goal of learning is to find a function $f \in \mathcal{F}$ such that the generalization error

$$E_P(f) := P(y \neq f(y))$$

is as small as possible. P is not known during training; instead, examples $(\boldsymbol{\xi}_i, y_i)$, $i = 1, \ldots, m$, are available, which are independent and identically distributed according to P. Training aims at minimizing the empirical error

$$\hat{E}_m(f) := \sum_{i=1}^{m} |\{y_i \neq f(\boldsymbol{\xi}_i)\}|/m \, .$$

The learning algorithm generalizes to unseen data if $\hat{E}_m(f)$ is representative for $E_P(f)$ for large enough m and every f. Assume a pattern $(\boldsymbol{\xi}, y)$ is classified by a GMLVQ network which implements the function f. We define the margin

$$M_f(\boldsymbol{\xi}, y) = -d_J^{\Lambda^J} + d_K^{\Lambda^K}$$

whereby $d_J^{\Lambda^J}$ refers to the distance from the closest prototype with class y and $d_K^{\Lambda^K}$ refers to the distance from the closest prototype with class different from

y. Note that LGMLVQ implicitely maximizes this term since it constitutes the nominator in the cost function (1). Following [1], we define the loss

$$L : \mathbb{R} \to \mathbb{R}, t \mapsto \begin{cases} 1 & \text{if } t \leq 0 \\ 1 - t/\rho & \text{if } 0 < t \leq \rho \\ 0 & \text{otherwise} \end{cases}$$

where $\rho > 0$ is some fixed value. The term

$$\hat{E}_m^L(f) := \sum_{i=1}^{m} L(M_f(\boldsymbol{\xi}_i, y_i))/m$$

accumulates the number of errors for a given data set, and, in addition, also punishes all correct classifications with a margin smaller than ρ. It is possible to correlate the generalization error and this modified empirical error by a dimensionality independent bound: the inequality

$$E_P(f) \leq \hat{E}_m^L(f) + \mathcal{O}\left(\frac{P^2(B^3 + \sqrt{\ln(1/\delta)})}{\rho\sqrt{m}} + \sqrt{\frac{\ln(1/\delta)}{m}}\right) \tag{3}$$

holds with probability $1 - \delta$ for every P and f (see [3] for the derivation). This bound is independent of the dimensionality of the data. Rather, it involves the margin ρ which is the nominator of the cost function, hence optimized during LGMLVQ training. Note that, in this formalism, the error of the algorithm can be uniformly bounded in terms of the training error (involving classifications with margin smaller than ρ) and a term which depends on some model parameters and ρ, but not on the input dimensionality and the matrix size. Usually, an appropriate ρ which yields good overall bounds is not known beforehand. In [3], a possibility to extend this result to a posterior parameter ρ is presented.

5 Experiments

Artificial Data. In a first experiment, the algorithm is applied to a two-dimensional artificial dataset consisting of two ellipsoidal classes as depicted in Fig.1(a). Two Gaussians are generated with mean values $\mu_1 = [1.5, 0.0]$ and $\mu_2 = [-1.5, 0.0]$, respectively, and variance $\sigma_{1,2} = [0.5, 3.0]$, and then rotated about the origin by the angles $\varphi_1 = \pi/4$ and $\varphi_2 = -\pi/6$, respectively. Training and test set consist of 300 resp. 600 datapoints per class. We test the standard euclidean metric (GLVQ), an adaptive diagonal metric (GRLVQ), individual adaptive diagonal metrics for each prototype (LGRLVQ), an adaptive matrix (GMLVQ), and individual adaptive matrices for every prototype (LGMLVQ). Relevance or matrix learning is done after an initial phase consisting of 500 epochs prototype adaptation and training is done for several 1000 epochs. In all experiments, learning rates are annealed during training, and they are chosen smaller for the matrix elements (initial rates ranging from 0.01 to 0.0001).

The classification accuracies on the training and test set are summarized in Tab.1. The position of the resulting prototypes and decision boundaries are shown in Fig.1(b)-(f). GMLVQ determines one single direction in feature space which is used for classification. The resulting matrix Ω projects the data onto