Improving the H2MLVQ algorithm by the Cross Entropy Method

Abderrahmane Boubezoul, Sébastien Paris, and Mustapha Ouladsine
Laboratoire des Sciences de l'Information et des Systèmes
Domaine Universitaire de Saint-Jérôme, avenue Escadrille Normandie-Niemen,
13397 MARSEILLE CEDEX 20, France
email: {abderrahmane.boubezoul,sebastien.paris,mustapha.ouladsine}@lsis.org

Keywords: Generalized Learning Vector Quantization, Relevance Learning,
Cross Entropy method, Initialization sensitiveness

***Abstract*— This paper addresses the use of a stochastic optimization method called the Cross Entropy (CE) Method in the improvement of a recently proposed H2MLVQ (Harmonic to minimum LVQ) algorithm , this algorithm was proposed as an initialization insensitive variant of the well known Learning Vector Quantization (LVQ) algorithm. This paper has two aims, the first aim is the use of the Cross Entropy (CE) Method to tackle the initialization sensitiveness problem associated with the original (LVQ) algorithm and its variants and the second aim is to use a weighted norm instead of the Euclidean norm in order to select the most relevant features. The results in this paper indicate that the CE method can successfully be applied to this kind of problems and efficiently generate high quality solutions. Also, good competitive numerical results on several datasets are reported.

1 Introduction

Prototype based learning has been an ongoing research problem for last decades and it has been approached in various ways. It has been gaining more interest lately due to its ability to generate fast and intuitive classification models with good generalization capabilities. One prominent algorithm of Prototype based learning algorithms is a Learning Vector Quantization (LVQ) introduced by Kohonen ([2]), this algorithm and its variants have been intensively studied because of their robustness, adaptivity and efficiency. The idea of LVQ is to define class boundaries based on prototypes, a nearest neighbor rule and a winner-takes-it-all paradigm. The standard LVQ has some drawbacks: i) basically LVQ adjusts the prototypes using heuristic error correction rules, ii) it does not directly minimize an objective function, thus it cannot guarantee the convergence of the algorithm which leads to instability behavior especially in the case of overlapped data, iii) the results are strongly dependent on the initial positions of the prototypes. In order to improve the standard LVQ algorithm several modifications were proposed by the author him self (see[3]) and by other researchers.

A good description of the state of the art of Learning Vector Quantization and its variants is given in a recent survey (see [5]). Standard LVQ does not distinguish between more or less informative features due to the usage of the Euclidean distance, to improve that, extensions from various authors are suggested (see [4] [6] and [7]). The previous approaches obey to heuristic update for relevance and prototypes vectors are adapted using simple perceptron learning which may cause problems for non linear separable data. For these reasons another variant based on minimisation of cost function using stochastic gradient descent method was proposed by Sato and Yamada (see [14]). Hammer and *al* suggested to modify the GLVQ cost function by using weighted norm instead of Euclidean distance. This algorithm, called Generalized Relevances LVQ, showed in several tasks competitive results compared to SVM and it had been proved that GRLVQ can be considered as a large margin classifier (see [8]).

Although the GRLVQ algorithm guarantees convergence and shows better classification performances than other LVQ algorithms, it suffers from initialization sensitiveness due to the presence of numerous local minima incurred as a result of the use of gradient descent method especially for multi-modal problems. The same authors proposed another algorithm (Supervised Relevance Neural Gas SRNG) to tackle initialization sensitiveness (see[9]). They propose to combine the GRLVQ with the neighborhood oriented learning in the neural gas (NG). The algorithm mentioned above required choosing several parameters such as learning rate, size of an update neighborhood. A suitable choice of these parameters values may not always be evident and also changed from one data set to another. In this paper, we present an initialization insensitive H2MRLVQ which is based on the well-known and efficient cross entropy (CE) method [10]. We refer to this new algorithm as the Cross Entropy Method (CEMH2MRLVQ).

The rest of the paper is structured as follows. In section 2, we introduce the basics of classification and prototype learning; we review both the formulation of GLVQ and H2MLVQ. In section 3, we explain how the CE method can be considered as a stochastic global optimization procedure for H2MLVQ algorithm, In section 4, we present the results of numerical experiments using our proposed algorithm on some benchmark data sets and compare with the results obtained using the H2MLVQ and GLVQ stochastic

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

gradient descent method. We conclude in section 5.

2 Problem statement

Machine learning can be formulated as searching for the most adequate model describing a given data. A learning machine may be seen as a function $f(\boldsymbol{\mathcal{X}};\boldsymbol{\theta})$ which transforms objects from the the input space $\boldsymbol{\mathcal{X}}$ to the output space $\boldsymbol{\mathcal{Y}}$: $f(\boldsymbol{\mathcal{X}};\boldsymbol{\theta}) : \boldsymbol{\mathcal{X}} \longrightarrow \boldsymbol{\mathcal{Y}}$.

The data domain $\boldsymbol{\mathcal{X}}$ and the set of target values $\boldsymbol{\mathcal{Y}}$ are determined by the definition of the problem for which $f(\boldsymbol{\mathcal{X}};\boldsymbol{\theta})$ is constructed. The output of the function can be a continuous value (for regression application) or can predict a class label of the input object (for classification application). The learning model $f(\boldsymbol{\mathcal{X}};\boldsymbol{\theta})$ usually depends on some adaptive parameters $\boldsymbol{\theta}$ sometimes also called free parameters. In this context, learning can be seen as a process in which a learning algorithm searches for these parameters $\boldsymbol{\theta}$, which solve a given task.

In supervised learning, the algorithm learns form the data set $\boldsymbol{\mathcal{S}}$ often called the "training set" and consists of N samples, $(\boldsymbol{x}, y)$ pairs, drawn independently identically from a probability distribution $p(\boldsymbol{x}, y)$. We define the collected data by $\boldsymbol{\mathcal{S}} \triangleq \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)\}$ where $\boldsymbol{x}_i \triangleq [x_{1i}, x_{2i}, \ldots, x_{di}]^T \in \boldsymbol{\mathcal{X}}$ and $\boldsymbol{y} \triangleq \{y_i\}_{i=1,\ldots,N}$, $y_i \in \{1, \ldots, L\}$. N, L denote the number of records in the data set and the number of classes respectively. In real applications $\boldsymbol{x}_i \in \boldsymbol{\mathcal{X}} \subset \mathbb{R}^d$ is a multidimensional real vector.

In the Bayes decision theory the sanity behavior of the classifier is usually measured by classification error, the so-called overall loss (see [11]):

$$\Re(\boldsymbol{\theta}) \triangleq \mathbb{E}_X \left[L(f(\boldsymbol{x};\boldsymbol{\theta})|\boldsymbol{x})\right] = \int_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} L(f(\boldsymbol{x};\boldsymbol{\theta})|\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}, \tag{1}$$

where $L(f(\boldsymbol{x};\boldsymbol{\theta})|\boldsymbol{x})$ is the expected loss defined by:

$$L(f(\boldsymbol{x};\boldsymbol{\theta})|\boldsymbol{x}) \triangleq \int_{y\in\mathcal{Y}} \ell(f(\boldsymbol{x};\boldsymbol{\theta})|y) p(y|\boldsymbol{x}) dy, \tag{2}$$

where $\ell(f(\boldsymbol{x};\boldsymbol{\theta})|y)$ denotes the individual loss. By rewriting eq.(1), we have:

$$\Re(\boldsymbol{\theta}) = \int_{\boldsymbol{x},y\in\boldsymbol{\mathcal{X}}\times\boldsymbol{\mathcal{Y}}} \ell(f(\boldsymbol{x};\boldsymbol{\theta})|y) p(\boldsymbol{x}, y) d\boldsymbol{x} dy, \tag{3}$$

more precisely

$$\Re(\boldsymbol{\theta}) = \sum_{k=1}^{L} p(y=k) \int_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} \ell(f(\boldsymbol{x};\boldsymbol{\theta})|y) p(\boldsymbol{x}|y=k) d\boldsymbol{x}. \tag{4}$$

We assume training data drawn by some underlying joint distribution $p(\boldsymbol{x}, y)$ which is in practice unknown. In real applications, only a finite number of samples are available. The loss functional $\Re(\boldsymbol{\theta})$ is usually replaced by the so-called empirical loss functional computed as follows:

$$\Re_{emp}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{L} \ell(f(\boldsymbol{x}_i;\boldsymbol{\theta})|y_i = k) \mathbf{1}(y_i = k), \tag{5}$$

$\mathbf{1}(.)$ is an indicator function such that $\mathbf{1}(true) = 1$ if the condition between the parentheses is satisfied or $\mathbf{1}(false) = 0$ if not.

2.1 GLVQ algorithm

It has been shown by Diamantini in [12] that traditional LVQ (LVQ1) proposed by Kohonen does not minimize neither explicit risk function, nor the Bayes risk. In [13] Juang proposed a learning scheme called minimum classification error (MCE) approach that minimizes the expected loss in Bayes decision theory by a gradient-descent procedure (Generalized Probabilistic Descent). The MCE criterion is defined using specific discriminant functions.

Let us consider $\boldsymbol{w}_{kj} \in \boldsymbol{\mathcal{X}}$ is the jth prototype among P_k vectors associated to class $k = 1, \ldots, L$ and $\boldsymbol{\theta}_k \triangleq \{\boldsymbol{w}_{kj} | j = 1, \ldots, P_k\} = \boldsymbol{W}_k$ is the collection of all prototypes of the class k. The collection of all prototypes representing all classes are defined as $\boldsymbol{\theta} \triangleq \bigcup_{k=1}^{L} \boldsymbol{\theta}_k = \boldsymbol{W}$ and $P = \sum_{l=1}^{L} P_l$ denotes the total number of prototypes.

As proposed by Sato and Yamada, the generalized LVQ learning algorithm use the following discriminant function $\mu_k(\boldsymbol{x};\boldsymbol{\theta})$ defined by:

$$\mu_k(\boldsymbol{x};\boldsymbol{\theta}) \triangleq \frac{d_k(\boldsymbol{x};\boldsymbol{\theta}) - d_l(\boldsymbol{x};\boldsymbol{\theta})}{d_k(\boldsymbol{x};\boldsymbol{\theta}) + d_l(\boldsymbol{x};\boldsymbol{\theta})}, \tag{6}$$

where $d_k(\boldsymbol{x};\boldsymbol{\theta})$ is the square euclidian distance between $\boldsymbol{x}$ and the closest prototype belonging to the same class of $\boldsymbol{x}$ and $d_l(\boldsymbol{x};\boldsymbol{\theta})$ is the squared Euclidean distance between the input vector $\boldsymbol{x}$ and its best matching prototype vector $\boldsymbol{w}_{lr}$ with a different class label. This discriminant function $\mu_k(\boldsymbol{x};\boldsymbol{\theta}) \in [-1, 1]$ ensure that if $\boldsymbol{x}$ is correctly classified to class k then $\mu_k(\boldsymbol{x};\boldsymbol{\theta}) \leq 0$.

Introducing (6) in (5), the empirical loss minimized by the MCE criterion is rewritten by:

$$\Re_{emp}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{L} \ell(\mu_k(\boldsymbol{x}_i;\boldsymbol{\theta})) \mathbf{1}(y_i = k), \tag{7}$$

where $\ell(\mu_k(\boldsymbol{x}_i;\boldsymbol{\theta})) = \ell(f(\boldsymbol{x}_i;\boldsymbol{\theta})|y = k)$. and $\mathbf{1}(.)$ is an indicator function such that $\mathbf{1}(true) = 1$ if the condition between the parentheses is satisfied or $\mathbf{1}(false) = 0$ if not. $\ell(\mu_k(\boldsymbol{x};\boldsymbol{\theta}))$ is a smoothed loss function instead of the $0-1$ loss function of the Bayes decision theory, for example the sigmoid function defined by:

$$\ell(z;\xi) \triangleq \frac{1}{1 + e^{-\xi z}}, \tag{8}$$

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

where ξ is a scalar (which usually increases with time). When $t \rightarrow \infty$ the loss function will be identical to empirical loss in bayes decision theory. From (7), the general GLVQ cost function is expressed as:

$$F_{GLVQ}(\boldsymbol{X};\boldsymbol{\theta}) = \sum_{i=1}^{N} \ell(\mu_k(\boldsymbol{x}_i;\boldsymbol{\theta})) = \sum_{i=1}^{N} \frac{1}{1+e^{-\xi(t)\mu_k(\boldsymbol{x}_i;\boldsymbol{\theta})}}, \tag{9}$$

where $\boldsymbol{X} \triangleq \{\boldsymbol{x}_i\}_{i=1,\ldots,N}$. Although the GLVQ algorithm ensures convergence and exhibits better classification performance than other LVQ algorithms.

2.2 H2MLVQ

In the H2MLVQ not only the two best prototypes (with the same and different labels as the input sample) are updated, but all prototypes with different degrees according to their nearness to the input sample (see [1]). The authors proposed to replace the distances d_k and d_l by harmonic average distances d_k^H and d_l^H, respectively, as follows:

$$d_k^H = \frac{N_k d_{kmin}}{1+\sum_{j=1,j\neq kmin}^{N_k}(d_{kmin}/\|x_i-w_j\|^2)^t}, \qquad d_l^H = \frac{N_l d_{lmin}}{1+\sum_{j=1,j\neq lmin}^{N_l}(d_{lmin}/\|x_i-w_j\|^2)^t}, \tag{10}$$

where N_k, N_l are the number of prototypes with the same and different label with the sample $\boldsymbol{x}_i$, respectively, t gradually changes from 1 to 0 as training proceeds. The misclassification measure $\mu_k(\boldsymbol{X},\boldsymbol{\theta})$ is defined as:

$$\mu_k(\boldsymbol{x};\boldsymbol{\theta}) \triangleq \frac{d_k^H(\boldsymbol{x};\boldsymbol{\theta}) - d_l^H(\boldsymbol{x};\boldsymbol{\theta})}{d_k^H(\boldsymbol{x};\boldsymbol{\theta}) + d_l^H(\boldsymbol{x};\boldsymbol{\theta})}. \tag{11}$$

The H2MLVQ cost function is expressed as:

$$F_{H2MLVQ}(\boldsymbol{X},\boldsymbol{\theta}) = \sum_{i=1}^{N} \frac{1}{1+e^{-\xi(t)\mu_k(\boldsymbol{x}_i,\boldsymbol{\theta})}}. \tag{12}$$

2.3 Attribute weighing and H2MLVQ algorithm

In general, prototype based classifiers are dependant on the metric which is used to measure proximity, the performance of this type of algorithms depends essentially on that choice. The usual choice is the Euclidian metric which is not always appropriate because it supposes that all the attributes contribute equally in the classification. We will use the same idea as Hammer and *al* in [5], they suggested to replace the Euclidean metric by Weighted Euclidean metric by introducing input weights $\boldsymbol{\lambda} \triangleq [\lambda_1, \lambda_2, \ldots, \lambda_d]^T \in \mathbb{R}^d$, $\lambda_i \geq 0$ to allow a different scaling of the inputs. They substitute the Euclidean metric by its scaled variant in the GLVQ formulation

$$d_\lambda(\boldsymbol{a},\boldsymbol{b}) \triangleq (\boldsymbol{a}-\boldsymbol{b})^T \boldsymbol{\lambda}\mathbf{I}(\boldsymbol{a}-\boldsymbol{b}), \tag{13}$$

where $(\boldsymbol{a},\boldsymbol{b}) \in \boldsymbol{\mathcal{X}}$ are two column vectors, $\mathbf{I}$ is identity matrix.

From (6) the new formulation of the misclassification measure $\mu_k^\lambda(\boldsymbol{x};\boldsymbol{\theta})$ can be expressed as

$$\mu_k^\lambda(\boldsymbol{x};\boldsymbol{\theta}) = \frac{d_k^\lambda(\boldsymbol{x};\boldsymbol{\theta}) - d_l^\lambda(\boldsymbol{x};\boldsymbol{\theta})}{d_k^\lambda(\boldsymbol{x};\boldsymbol{\theta}) + d_l^\lambda(\boldsymbol{x};\boldsymbol{\theta})}, \tag{14}$$

where $d_k^\lambda(\boldsymbol{x};\boldsymbol{\theta})$ and $d_l^\lambda(\boldsymbol{x};\boldsymbol{\theta})$ are now the corresponding weighted distances. Now the H2MRLVQ cost function is computed as follows

$$F_{H2MRLVQ}(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{i=1}^{N} \frac{1}{1+e^{-\xi(t)\mu_k^\lambda(\boldsymbol{x}_i;\boldsymbol{\theta})}}. \tag{15}$$

In the next section we will present the Cross Entropy Method to tackle the initialization sensitiveness problem and to estimate the attribute's weights.

3 Cross Entropy Method

The general optimization problem introduced in the GLVQ and H2MLVQ formulation can be viewed as a task of finding a best set of parameters $\boldsymbol{W}$ and $\boldsymbol{\lambda}$ that minimize the corresponding objective function. These optimization problems are typically quite difficult to solve; *i.e.* $\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \{F(\boldsymbol{\theta})\}$, where $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ represents the vector of input variables, $F(\boldsymbol{\theta})$ is the scalar objective function and $\boldsymbol{\Theta}$ is the constraint set. Many approaches exist to solve this kind of problems (Gradient based procedures, random search, Meta heuristics, model-based methods,...) (see [16]).

As an alternative to the stochastic gradient descent algorithm we consider the CE approach because it offers good results for multi-extremal functional optimization (see [17]). This method requires neither special form of the objective function and its derivatives nor heuristic assumptions.

We view the problem of minimization of cost function (12) as a continuous multi-extremal optimization problem with constraints. The cross-entropy (CE) method was introduced by Rubinstein (see [10]) as an efficient method for the estimation of rare-event probabilities and has been successfully applied to a great variety of research areas (see for example ([18],[19],[15],[20]). The main ideas of the CE method are described in [21] and [15]. For this reason, in this paper we only present features of CE that are relevant to the problem hereby studied.

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

The central idea of Cross Entropy (CE) is related to the association of a stochastic problem to the original optimization problem, called parameterized associated stochastic problem (ASP) characterized by a density function $p(.;\boldsymbol{v}), \boldsymbol{v} \in \boldsymbol{\mathcal{V}}$. The stochastic problem is solved by identifying the optimal importance sampling density p^* which minimizes the Kullback-Leibler distance (also called the cross-entropy) between p and p^*. The CE method can be viewed as an iterative method that involves two major steps until convergence is reached:

(1) Generating samples according to $p(.;\boldsymbol{v})$ and choosing the elite of these samples.

(2) Updating the parameter $\boldsymbol{v}$ of the distribution family on the basis of the elite samples, in order to produce a better solution in the next iteration.

In this paper we will give a brief introduction of CE, the reader can refer to [15] for a more detailed description of the CE method.

Consider the following general cost function minimization problem. Let $\boldsymbol{\Theta}$ be a constraint set and $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is a a given vector. Let us denote the desired minimum of the function $F(\boldsymbol{\theta})$ by $\gamma^\star$. The cost function minimization problem can be formulated by:

$$\gamma^\star = \min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \{F(\boldsymbol{\theta})\}. \tag{16}$$

In other words, we seek an optimal solution $\gamma^\star$ satisfying $F(\boldsymbol{\theta}^\star) \leq F(\boldsymbol{\theta}), \forall \boldsymbol{\theta} \in \boldsymbol{\Theta}$. The CE method transforms the deterministic optimisation problem (16) to stochastic problem using a family of probability density functions (pdfs) $p\left(.;\boldsymbol{v}\right)$ which depends on a reference parameter $\boldsymbol{v} \in \mathcal{V}$. Consequently, the associated stochastic estimation problem is

$$\begin{aligned} l(\gamma) &\triangleq \mathbb{P}_{\boldsymbol{v}}\left(F(\boldsymbol{\Theta}) \leq \gamma\right) = \mathbb{E}_{\boldsymbol{v}}\left[\boldsymbol{I}_{F(\boldsymbol{\Theta})\leq\gamma}\right] \\ &= \int \boldsymbol{I}_{F(\boldsymbol{\theta})\leq\gamma} p\left(.;\boldsymbol{v}\right) d\boldsymbol{\theta}. \end{aligned} \tag{17}$$

At each iteration t of the CE algorithm V random samples are drawn on the basis of $p\left(.;\boldsymbol{v}_{t-1}\right)$, then the new value of $\boldsymbol{v}_t$ is updated according to $\boldsymbol{v}_{t-1}$ and the best elite samples. The update formula is especially simple if $p\left(.;\boldsymbol{v}\right)$ is belonging to Natural Exponential Function (NEF)(Gaussian, Truncated Gaussian, Binomial,...). Instead of updating the parameter vector $\widehat{\boldsymbol{v}}_{t-1}$ to $\widehat{\boldsymbol{v}}_t$ directly the smoothed updating procedure is often used:

$$\widehat{\boldsymbol{v}}_t = \alpha\widetilde{\boldsymbol{v}}_t + (1-\alpha)\widehat{\boldsymbol{v}}_{t-1}, \tag{18}$$

with $0 \leq \alpha \leq 1$ and $\widetilde{\boldsymbol{v}}_t$ is the solution of (17). This smoothed adaptation (18) is used to reduce the probability of the algorithm to get stuck in a local minima.

3.1 Cross Entropy Method for H2MRLVQ optimisation

In order to link the H2MLVQ cost function minimization (15) with the CE theory, we define the parameter $\boldsymbol{\theta}$ as $\boldsymbol{\theta} \triangleq \{\boldsymbol{W}, \boldsymbol{\lambda}\}$. Both $\boldsymbol{W}$ and $\boldsymbol{\lambda}$ are sampled from two sampling distributions which belong to the NEF family in order to simplify parameters update.

In this paper we take Truncated Gaussian and Dirichlet distributions to estimate the prototypes and attribute's weights, respectively. That is, for each $\boldsymbol{W}^l \triangleq \{w^l_{pq}\}_{\substack{p=1,\ldots,d \\ q=1,\ldots,P}}$ (a $d \times P$ matrix) with $l = 1,\ldots,V$, each components w^l_{pq} are drawn from a Truncated Gaussian such $w^l_{pq} \sim \mathcal{N}t\left(m^t_{pq}, (\sigma^t_{pq})^2, \boldsymbol{a}_p, \boldsymbol{b}_p\right)$ where m^t_{pq} denotes the average of pq^{th} components at iteration t, $(\sigma^t_{pq})^2$ the variance and $\boldsymbol{a}_p, \boldsymbol{b}_p$ denote a lower and upper bounding box for each dimension containing all data respectively. In practice, we choose $\boldsymbol{a}_p = C \min_{j=1,\ldots,N}\{x_{jp}\}$ and $\boldsymbol{b}_p = C \max_{j=1,\ldots,N}\{x_{jp}\}$ with $C \geq 1$. To sample attribute's weights we use the Dirichlet distribution. Let $\boldsymbol{\lambda} \triangleq [\lambda_1, \lambda_2, \ldots, \lambda_d]^T$ a random vector whose elements sum to 1. The density function of the Dirichlet distribution with the parameter vector $\boldsymbol{\delta}$ is:

$$\boldsymbol{\lambda}(\boldsymbol{\delta}) \sim \mathcal{D}ir(\delta_1, \delta_2, \ldots, \delta_d) = \frac{\Gamma(\sum_d \delta_d)}{\prod_d \Gamma(\delta_d)} \prod_d \lambda_d^{\delta_d - 1}, \tag{19}$$

where $\lambda_d > 0$, $\sum_d \lambda_d = 1$. The parameters $\boldsymbol{\delta}$ of the Dirichlet distribution can be estimated by maximizing the log-likelihood function of given data. The log-likelihood is convex in $\boldsymbol{\delta}$ which guarantee a unique optimum. To estimate $\boldsymbol{\delta}$, we use the simple and efficient iterative gradient descent method described in minka's paper (see [23]). The parameters $\left(\boldsymbol{M}^t \triangleq \{m^t_{pq}\}, \boldsymbol{\delta}^t \triangleq \{\delta^t_p\}, \boldsymbol{\Sigma}^t \triangleq \{(\sigma^t_{pq})^2\}\right)$ at iteration t are updated *via* the sample mean and sample standard deviation of elite samples. The algorithm is summarized as follows:

According to Kroese in [17] the performance CE method is insensitive to the exact choice of parameters. The algorithm is quite robust under the choice of the initial parameters, provided that the initial variances are chosen large enough, ensuring at the beginning to cover all solutions space.

To prevent the algorithm from being trapped in local optima Kroese in [17] proposed to use dynamic smoothing (20) where at each iteration the variance $(\sigma^t_{pq})^2$ is updated using a smoothing parameter:

$$\beta_t = \beta_0 - \beta_0\left(1 - \frac{1}{t}\right)^c, \tag{20}$$

where c is a small integer (typically between 5 and 15) and β is a large smoothing constant (typically between 0.8 and

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

(1) Choose some initial $\{\boldsymbol{M}^0, \boldsymbol{\Sigma}^0, \boldsymbol{\delta}^0\}$ for $p = 1, \ldots, d, q = 1, \ldots, P$. Set $t = 1$ (level counter).
(2) Draw samples $\boldsymbol{W}^l \sim \mathcal{N}t\left(\boldsymbol{M}^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)}, \boldsymbol{a}_p, \boldsymbol{b}_p\right)$, $\boldsymbol{\lambda}^l \sim \mathcal{D}ir(\boldsymbol{\delta}^{(t-1)}), l = 1, \ldots, V$.
(3) Compute $S^l = F_{H2MRLVQ}(\boldsymbol{X}; \boldsymbol{\theta}^l)$ scores by applying eq.(15) $\forall l$.
(4) Sort S^l in ascending order and denote by $\mathcal{I}$ the set of corresponding indices. Let us denote $\left(\widetilde{m}_{pq}^{(t-1)}, (\widetilde{\sigma}_{pq}^{(t-1)})^2\right)$ the mean and the variance of the best $\lceil \rho V \rceil$ prototypes elite samples of $\{\boldsymbol{W}^{\mathcal{I}(l)}\}, l = 1, \ldots, \lceil \rho V \rceil$ respectively. The $\widetilde{\boldsymbol{\delta}}^t$ is the corresponding the Dirichlet parameter fitted on $\{\boldsymbol{\delta}^{\mathcal{I}(l)}\}$.
(5) $\widehat{\boldsymbol{M}}^t = \alpha \widetilde{\boldsymbol{M}}^t + (1-\alpha)\widehat{\boldsymbol{M}}^{t-1}$, $\widehat{\boldsymbol{\Sigma}}^t = \beta_t \widetilde{\boldsymbol{\Sigma}}^t + (1-\beta_t)\widehat{\boldsymbol{\Sigma}}^{(t-1)}$ and $\widehat{\boldsymbol{\delta}}^t = \alpha \widetilde{\boldsymbol{\delta}}^t + (1-\alpha)\widehat{\boldsymbol{\delta}}^{(t-1)}$
(6) If convergence is reached or $t = T$ (T denote the final iteration), then **stop**; otherwise set $t = t + 1$ and reiterate from step 2.

Figure 1: Cross Entropy Method for H2MRLVQ optimisation: CEMH2MRLVQ.

0.99). By using β instead of α the convergence to the degenerate case has polynomial speed instead of exponential.

4 Experimental results

In this section, we applied both the CEMH2MRLVQ algorithm with some machine learning algorithms (GLVQ, H2MLVQ) to show the effectiveness of the proposed method. Following standard procedure in experiments with other published works, each data set is initially normalized and algorithm parameters are selected as described in the papers related to the algorithms described above. To obtain meaningful comparisons against other published algorithms and to assess the effectiveness of the proposed algorithm, we used a stratified 10 fold cross-validation. We tested these algorithms in real world data sets taken from the public UCI repository [24] (see Table 1). For comparison we used the same data sets as in [1]. In particulary, we used glass data set because it contains features with intrinsic within-class multi-modal structure. The parameters in the algorithms were set as follows:

- GLVQ: $\xi = 0.5$, $\varepsilon_k = 0.05$, $\varepsilon_l = 0.01$ where ε_k and ε_l are learning rates for nearest correct and wrong prototype, respectively.
- H2MLVQ: $\xi = 0.1$, $\varepsilon_k = 0.05$, $\varepsilon_l = 0.01$. For these two algorithms, the prototypes for different classes were randomly initialized around the center of the corresponding classes. Number of prototypes per class is set to four as in [1].
- For the CEMH2MRLVQ: $V = 5N$ where N denote the number of train samples, $\xi = 0.1$, $\rho = 3.10e-3$, $\beta_0 = 0.95$, $h = 5.10e-6$, $q = 10$, $\epsilon = 5.10e-9$ et $c = 50$, $C = 1.2$. It is found empirically that when α is between 0.6 and 0.9 it gives best results in our case we choose $\alpha = 0.7$.

For all algorithms the number of iterations is set to $K = 600$. We based our comparison on two criteria: the error rate and optimal value of the cost function. Results are presented in Tables (2 and 3) respectively. We see in the Table 2 that our algorithm outperforms the GLVQ algorithm and shows slightly better performances than H2MLVQ. In Table 3 we see that the CEMH2MRLVQ gives the lowest value of the cost function which leads to high quality solutions of the optimization problem. Compared to other algorithms based on gradient descent, the CEMH2MRLVQ is more demanding on computational cost of running. The theoretical complexity of CE is an open problem still under investigation, this complexity is partially depending on the studied problem (see [22]). The computational complexity of our algorithm is around $O(dNPV)$ in each iteration. The architecture of CE method algorithm being inherently parallel, it requires to evaluate a cost function, consequently the CE method can be accelerated by parallelizing all these evaluations.

Table 1: List real world data sets used in comparaison between algorithms.

Name	# Features	# Patterns	# Classes
Liver	6	345	2
Glass	9	214	6

Table 2: Recognition rates on real data sets. The format of the numbers in this table is $M \pm S$, where M is the mean recognition rate, S is the standard deviation. For each data set, the best method is described by the boldface.

Name	GLVQ (%)	H2MLVQ (%)	CEMH2MRLVQ (%)
Liver	57.45±9.82	59.41±4.42	**62.29±5.04**
Glass	60.34±7.40	63.66±8.37	**66.55±7.18**

Table 3: Optimal Cost Function Values. The format of the numbers in this table is $M \pm S$, where M is the mean Cost Function Value, S is standard deviation.

Name	H2MLVQ	CEMH2MRLVQ
Liver	155.4469±0.076	**155.3483±0.0093**
Glass	97.8378±0.0139	**97.7194±0.0053**

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

5 Attribute weighing estimation

Feature ranking is hard to compare because of the differences in data interpretation. In this section, we will be concerned with the Iris data set. This is one of the best known databases in the pattern recognition literature. The data set contains three classes (Iris Setosa, Iris Versicolour, and Iris Virginica) of 50 instances each. There are four input attributes (sepal length, sepal width, petal length, and petal width). The relevance vector that resulted after the experiment is $[.1108\ .1220\ .2027\ .5645]$ this result points out that the most important features are the latest two features. Our ranking is similar to GRLVQ [5] results $[.2353\ .2193\ .2704\ .2750]$.

6 Conclusion

In this paper we considered solving the LVQ initialization sensitiveness problem. We formulated it as a stochastic optimization problem and applied the CE method to solve it. The suggested method was shown to be apt to deal well with such problems. It produced recognition rates that were fairly superior to other proposed methods. The main benefit of the proposed method is its insensitivity to the choice of its hyper parameters which is not the case of the other methods (learning rate and size of an update neighborhood (SRNG)). In general, the CE algorithm is efficient and easy to implement. The CE can be seen as a stochastic method, which gives CE the ability to find more global optima than deterministic methods.

References

[1] A.K. Qin and P.N. Suganthan, Initialization Insensitive LVQ Algorithm Based on Cost-Function Adaptation, Pattern Recognition, Vol. 38, pp. 773-776, 2005.

[2] T. Kohonen, Learning vector quantization for pattern recognition, Technical Report TKK-F-A601, Helsinki University ofTechnology, Finland, 1986.

[3] T. Kohonen, Improved versions of learning vector quantization, Proceedings of the IJCNN, San Diego, 1990.

[4] T. Kohonen,Bibliography on the Self-Organizing Map (SOM) and Learning Vector Quantization(LVQ), Neural Networks Research Centre Helsinki University of Technology, 2002.

[5] B. Hammer and T. Villman, Generalized relevance learning vector quantization, Neural Networks, Vol. 15, pp. 1059-1068, 2002.

[6] T. Bojer, B. Hammer, D. Schunk, and K. von Toschanowitz, Releavance determination in learning vector quantization, in Proceedings of the European Symposium on Artificial Neural networks (ESANN 2001), M. Vereysen, Ed, D-side plublications, pp. 271-276 ,2001.

[7] B. Hammer and T. Villman, 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, pp. 173-180, 2001.

[8] B. Hammer and M. Strickert and T. Villmann,Relevance LVQ versus SVM, Seventh International Conference Artificial Intelligence Soft Computing (ICAISC 2004), Lecture Notes in Computer Science, 592–597, 2004.

[9] B. Hammer, M. Strickert, T. Villmann,Supervised neural gas with general similarity measure, Neural Processing Letters, $http://www.informatik.uni-osnabrueck.de/barbara/papers/pub\_hammer.html$.

[10] R.Y. Rubinstein, Optimization of computer simulation models with rare events, European Journal of operational Research, pp. 89-112, Vol.99, 1997.

[11] R.O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, 1973.

[12] C. Diamantini and A. Spalvieri, Certain facts about Kohonen's LVQ1 algorithm, IEEE Transactions on Circuits and Systems, Vol.I 47, pp. 425-427, 1996.

[13] B.H. Juang and S. Katagiri, Discriminative learning for minimum error classification, IEEE Transactions on Signal Processing, Vol. 40 (2), pp. 3043-3054, 1992.

[14] A.S. Sato and K. Yamada, A formulation of learning vector quantization using a new misclassification measure, Proceedings of the 14th International Conference on Pattern Recognition, Brisbane, Australia, pp. 332-325, 1998.

[15] De Boer, P-T. and D.P. Kroese and S. Mannor and R. Y. Rubinstein, A Tutorial on the Cross-Entropy Method, Annals of Operations Research,Vol. 134, pp. 19-67, 2005.

[16] M.C. Fu, Fred W. Glover and J. April, Simulation Optimization: A Review, New Developements, And Applications, Proceedings of the 2005 Winter Simulation Conference, M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A.Joines, eds.

[17] Kroese, D. P. and R.Y. Rubinstein and S. Porotsky, The Cross-Entropy Method for Continuous Multi-extremal Optimization, Methodology and Computing in Applied Probability, Vol.8,pp. 383-407, 2006.

[18] O. Wittner and B.E. Helvik, Cross entropy guided ant-like agents finding dependable primary/backup path patterns in networks, In Proceedings of Congress on Evolutionary Computation (CEC2002), Honolulu, Hawaii, May 12-17th, 2002.

[19] Z. Liu, A. Doucet and S.S. Singh, The cross-entropy method for blind multiuser detection, In IEEE International Symposium on Information Theory, Chicago, 2004.

[20] K. Chepuri and T. Homem de Mello, Solving the vehicle routing problem with stochastic demands using

the cross entropy method, Annals of Operations Research, Vol. 134, pp. 153-181, 2005.

[21] R.Y. Rubinstein and D.P. Kroese, The Cross-Entropy method: A unified approach to combinatorial method, monte-carlo simulation and machine learning, Springer Verlag, 2004.

[22] J. Wu and A.C.S. Chung, Cross Entropy: A New Solver for Markov Random Field Modeling and Applications to Medical Image Segmentation, The Eighth International Conference on Medical Image Computing and Computer-Assisted Intervention 2005 (MICCAI05), Palm Springs, California, USA, LNCS 3749, 2005.

[23] T. Minka,Estimating a Dirichlet distribution, http://www.stat.cmu.edu/ minka/papers/dirichlet.pdf,2003.

[24] D.J. Newman and S. Hettich and C. L. Blake and C. J. Merz, UCI Repository of machine learning databases, http://www.ics.uci.edu/∼mlearn/MLRepository.html,1998.

[25] X. Zeng, Tony R. Martinez, Distribution-balanced stratified cross-validation for accuracy estimation, J. Exp. Theor. Artif. Intell, 12(1): 1-12, 2000.

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de