



Application of the cross entropy method to the GLVQ algorithm

Abderrahmane Boubezoul, Sébastien Paris*, Mustapha Ouladsine

Laboratory of Sciences of Information's and of System, LSIS UMR6168, University Paul Cézanne, Aix-Marseille III Av Escadrille de Normandie Niemen, 13397 Marseille Cedex 20, France

ARTICLE INFO

Article history:

Received 22 December 2006

Received in revised form 7 March 2008

Accepted 19 March 2008

Keywords:

Generalized learning vector quantization

Cross entropy method

Initialization sensitiveness

ABSTRACT

This paper discusses an alternative approach to parameter optimization of well-known prototype-based learning algorithms (minimizing an objective function via gradient search). The proposed approach considers a stochastic optimization called the cross entropy method (CE method). The CE method is used to tackle efficiently the initialization sensitiveness problem associated with the original generalized learning vector quantization (GLVQ) algorithm and its variants. Results presented in this paper indicate that the CE method can be successfully applied to this kind of problem on real-world data sets. As far as known by the authors, it is the first use of the CE method in prototype-based learning.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

In statistical pattern recognition, a wide variety of classification techniques have been proposed based on minimization of the classification error or a generalized risk functions. The research field of neural networks (NN) has attracted the attention of many scientists in the last decade. A huge variety of artificial NN have been developed based on many different learning techniques and topologies. One prominent class of NN is prototype-based learning, which generates fast and intuitive classification models with good generalization capabilities. In the last decades, a learning vector quantization (LVQ) prototype-based algorithm introduced by Kohonen [1] 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) basic 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, including Kohonen [2], Pregenzer et al. [3], Somervuo and Kohonen [4], Bojer et al. [5], Hammer and Villman [6], to name but a few. In this paper we will be interested in approaches where learning is related to the minimization of an appropriate cost function (see Refs. [7,6,8]).

Sato and Yamada [7] proposed an algorithm based on minimization of continuous and differentiable objective function with respect to the classifier parameters (prototype vectors) by stochastic gradient descent. This algorithm is considered as the most relevant contribution in LVQ improvement. However, it is crucially dependent on the prototype's initial position. To overcome this drawback, Hammer et al. [9] proposed an efficient combination between generalized learning vector quantization (GLVQ), the neural gas algorithm, and scaling the input dimensions according to their relevance (SRNG). Bojer et al. [10] proposed a dynamic variant to overcome the limitation due to initialization. This algorithm proposes an online adaptation of the prototypes to tackle the initialization sensitiveness associated with the GLVQ algorithm. This adaptation is carried out by modifying the prototype set of the classifier according to its recognition performance and by adding new prototypes. The algorithm starts by initializing one prototype in the center of the clusters for each class. The growing method adds new prototypes in those local regions where the misclassification error is greater. Additional prototypes are added iteratively during training if the cost function has reached a plateau and a significant number of errors still exist. This algorithm is efficient to implement and shows superior performances to other variants of the LVQ algorithm. However, this algorithm has some drawbacks, one of them consists in adding prototypes without deleting, especially for confusing ones or those prototypes that do not form class borders. One bad effect of this drawback is an insufficient generalization capability. Crammer et al. [11] showed that using too many prototypes might result in poor performance. To overcome this problem of overfitting, the authors advise to use a test set in monitoring training. The second drawback of GLVQ is its use of a deterministic iterative algorithm to minimize the objective function.

In general, this procedure does not guarantee getting a global minimum. Iterative techniques only ensure convergence to a local

* Corresponding author. Tel.: +33 491 05 60 66.

E-mail address: sebastien.paris@lsis.org (S. Paris).

minimum of the objective function. Another variant for making GLVQ insensitive to the initialization was proposed by Qin and Suganthan [8] called the harmonic to minimum LVQ algorithm (H2MLVQ). The authors introduce a transition procedure from harmonic average distance measure to minimum squared Euclidean distance measure into the original GLVQ cost function. Both SRNG and H2MLVQ have a common property in that they require choosing some hyper-parameters whose values have to be properly chosen, such as learning rate, size of an update neighborhood (SRNG) and a strategy to accommodate the latter during learning. An appropriate choice of these parameters' values may not always be easy and varies from one data set to another to achieve good performances. In this paper, we present an initialization insensitive GLVQ which is based on the well-known cross entropy (CE) method [12]. This method has been applied to clustering and vector quantization problem (see Kroese et al. [13]). From now on, we refer to this new algorithm as the CE method GLVQ (CEMGLVQ).

The rest of the paper is organized as follows. In Section 2, we introduce the basics of classification and prototype learning based on parameter optimization. Especially we review the GLVQ formulation. In Section 3, we present the CE method as a global optimization procedure and we reformulate it in the context of the GLVQ algorithm. In Section 4, we present the results of numerical experiments using our new method on some benchmark data sets and compare results with up-to-date top scoring algorithms. Finally, we will conclude and propose perspectives in Section 5.

2. Problem statement

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

The data domain \mathcal{X} and the set of target values \mathcal{Y} are determined by the definition of the problem for which the $f(\mathbf{x}; \theta)$ 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 $f(\mathbf{x}; \theta)$ usually depends on some adaptive parameters θ sometimes also called free parameters. In this context, learning can be seen as the process of searching the parameters θ that solves a given task.

In supervised learning, the algorithm learns from the data set \mathcal{S} often called the "training set" and consists of N samples, (\mathbf{x}, y) pairs, drawn independently identically from a probability distribution $p(\mathbf{x}, y)$. We define the collected data by $\mathcal{S} \triangleq \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$ where $\mathbf{x}_i \triangleq [x_{i1}, x_{i2}, \dots, x_{id}]^T$ and $\mathbf{y} \triangleq \{y_i\}_{i=1, \dots, N}$, $y_i \in \{1, \dots, L\}$. N and L denote the number of records in the data set and the number of classes, respectively. In real applications, $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ is a multi-dimensional vector.

In Bayesian decision theory, the optimal classifier is obtained by minimizing the risk function (overall loss) (see Ref. [14]):

$$\mathfrak{R}(\theta) \triangleq \mathbb{E}[L(f(\mathbf{x}; \theta) | \mathbf{x})] = \int L(f(\mathbf{x}; \theta) | \mathbf{x}) \cdot p(\mathbf{x}) \, d\mathbf{x}, \tag{1}$$

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

$$L(f(\mathbf{x}; \theta) | \mathbf{x}) \triangleq \int \ell(f(\mathbf{x}; \theta) | y) \cdot p(y | \mathbf{x}) \, dy, \tag{2}$$

where $\ell(f(\mathbf{x}; \theta) | y)$ denotes the individual loss.

By rewriting Eq. (1), we have

$$\mathfrak{R}(\theta) = \sum_{k=1}^L p(y = k) \int \ell(f(\mathbf{x}; \theta) | y) \cdot p(\mathbf{x} | y = k) \, d\mathbf{x}. \tag{3}$$

We assume that the training data are drawn from some underlying distribution $p(\mathbf{x} | y)$, which is in practice unknown or not trivial to

estimate. In real applications, only a finite number of samples are available. The risk functional $\mathfrak{R}(\theta)$ is usually replaced by the empirical risk computed as follows:

$$\mathfrak{R}_{emp}(\theta) = \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^L \ell(f(\mathbf{x}_i; \theta) | y_i = k) \mathbf{1}(y_i = k), \tag{4}$$

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

2.1. GLVQ

It has been shown by Diamantini and Spalvieri [15] that traditional LVQ (LVQ1), proposed by Kohonen, does not minimize neither explicit risk function nor the Bayes risk. Juang and Katagiri [16] proposed a learning scheme called minimum classification error (MCE), which minimizes the expected loss in Bayesian decision theory by a gradient-descent procedure (generalized probabilistic descent, etc.). The MCE criterion is defined using specific discriminant functions. Let us consider $\mathbf{w}_{kj} \in \mathcal{X}$ is the j th prototype among P_k vectors associated with class $k = 1, \dots, L$ and let $\theta_k \triangleq \{\mathbf{w}_{kj} | j = 1, \dots, P_k\}$ be the collection of all prototypes of the class k . The collection of all prototypes representing all classes is defined as $\theta \triangleq \bigcup_{k=1}^L \theta_k$, and $P = \sum_{l=1}^L P_l$ denotes the total number of prototypes.

Suppose that \mathbf{x} is the input vector fed to the system and $y = k$. The discriminant functions of the classes k and l , with $(l \neq k)$, are defined as follows:

- $g_k(\mathbf{x}; \theta) = -d_k(\mathbf{x}; \theta) \triangleq -\min_j \|\mathbf{x} - \mathbf{w}_{kj}\|^2$, where $d_k(\mathbf{x}; \theta)$ is the squared Euclidian distance between \mathbf{x} and the closest prototype belonging to the same class as \mathbf{x} .
- $g_l(\mathbf{x}; \theta) = -d_l(\mathbf{x}; \theta) \triangleq -\max_{l \neq k} \|\mathbf{x} - \mathbf{w}_{li}\|^2$.

If the following condition is satisfied, then \mathbf{x} is classified to class k : $g_k(\mathbf{x}; \theta) \geq g_l(\mathbf{x}; \theta)$ for all $k \neq l$.

A misclassification measure of the class k can be defined as follows:

$$\mu_k(\mathbf{x}; \theta) \triangleq -g_k(\mathbf{x}; \theta) + \left[\frac{1}{L-1} \sum_{j, j \neq k} g_j(\mathbf{x}; \theta) \right]^{1/\eta}, \tag{5}$$

where η is a positive constant. When $\eta \rightarrow \infty$, the misclassification measure becomes $\mu_k(\mathbf{x}; \theta) = -g_k(\mathbf{x}; \theta) + \max_{l \neq k} g_l(\mathbf{x}; \theta) = d_k(\mathbf{x}; \theta) - d_l(\mathbf{x}; \theta)$.

In a more general case, by considering both the closest genuine prototype and the incorrect prototype, Sato and Yamada proposed the generalized LVQ learning algorithm which ensures a better convergence of the prototypes. The new misclassification measure $f_k(\mathbf{x}; \theta)$ is defined by

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

where $d_l(\mathbf{x}; \theta)$ is the squared Euclidian distance between the input vector \mathbf{x} and its best matching prototype vector \mathbf{w}_{lr} with a different class label.

Introducing Eq. (6) in Eq. (4), the empirical loss minimized by the MCE criterion is rewritten as

$$\mathfrak{R}_{emp}(\theta) = \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^L \ell(f_k(\mathbf{x}_i; \theta)) \mathbf{1}(y_i = k), \tag{7}$$

where $\ell(f_k(\mathbf{x}_i; \theta)) = \ell(f(\mathbf{x}_i; \theta) | y_i = k)$.

Here, $\ell(f_k(\mathbf{x}; \theta))$ is a smoothed loss function instead of the 0–1 loss function in Bayesian decision theory, for example, the sigmoid function defined by

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

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

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

where $\mathbf{X} \triangleq \{\mathbf{x}_i\}_{i=1, \dots, N}$.

Although the GLVQ algorithm ensures convergence and exhibits better classification performances than other LVQ algorithms, it suffers from initialization sensitiveness, because the gradient-descent procedure may easily get trapped in local minimal states especially for multi-modal problems. To tackle this problem, we present the CEMGLVQ algorithm.

3. Cross entropy method

The general optimization problem can be viewed as a task of finding a best set of parameters that minimize a given objective function, i.e. $\min_{\theta \in \Theta} F(\theta)$, optimization problems are typically quite difficult to solve; in the context of combinatorial problems, they are often \mathcal{NP} -hard, where $\theta \in \Theta$ represents the vector of input variables, $F(\theta)$ is the scalar objective function and Θ is the constraint set. Many approaches exist to solve these kinds of problems (gradient-based procedures, random search, meta heuristics, model-based methods, etc.), see for example Fu et al. [17].

The choice of the optimization method is important in many learning problems. We would like to use optimization methods that can handle a large number of features, converge fast and return sparse classifiers. 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 Kroese et al. [18]). 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 (9) as a continuous multi-extremal optimization problem with constraints.

The CE method was introduced by Rubinstein [12] 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 Refs. [19–22]). The main ideas of the CE method are described in Rubinstein and Kroese [23] and Kroese et al. [18]. For this reason, in this paper we only present features of CE that are relevant to the problem hereby studied.

The CE method can be viewed as an iterative method that involves two major steps until convergence is reached:

- (1) Generating samples according to $p(\cdot; \mathbf{v})$ and choosing the elite of these samples.
- (2) Updating the parameter \mathbf{v} of the distribution family on the basis of the elite samples, in order to produce a better samples in the next iteration.

Consider the following general cost function minimization problem. Let Θ be a constraint set and $\theta \in \Theta$ a given vector. Let us denote the wanted minimum of the function $F(\theta)$ by γ^* . The cost function minimization problem can be formulated by

$$\gamma^* = \min_{\theta \in \Theta} F(\theta). \quad (10)$$

- (1) Choose some $\hat{\mathbf{v}}_0 \in \mathcal{V}$. Set $t = 1$ (level counter).
- (2) Generate samples $\theta^1, \dots, \theta^V$ from the density $p(\cdot; \hat{\mathbf{v}}_{t-1})$ and compute the ρ -quantile $\hat{\gamma}_{t-1}$ of the sample scores.
- (3) Use the same samples to solve the stochastic program by:

$$\max_{\mathbf{v}} \widehat{D}(\mathbf{v}) = \max_{\mathbf{v}} \left\{ \frac{1}{V} \sum_{l=1}^V \mathbf{I}_{\{F(\theta^l) \leq \hat{\gamma}_{t-1}\}} \ln p(\theta^l; \mathbf{v}) \right\}. \quad (11)$$

Denote the solution by $\hat{\mathbf{v}}_t$.

- (4) If predefined stopping criterion is met, then **stop**; otherwise set $t = t + 1$ and reiterate from step 2.

Fig. 1. Proto-typical version of the CE method.

The CE method transforms the deterministic optimization problem (10) into a stochastic problem using a family of probability density functions (pdfs).

At each iteration t of the CE algorithm, random samples are drawn on the basis of $p(\cdot; \mathbf{v}_{t-1})$, then the new value of \mathbf{v}_t is updated according to \mathbf{v}_{t-1} and the best elite samples. The update formula is especially simple if $p(\cdot; \mathbf{v})$ belongs to natural exponential family (NEF) (Gaussian, truncated Gaussian, binomial, etc.). More details about the general CE algorithm are given in Kroese et al. [18].

We present a proto-typical version of the CE method in Fig. 1.

Instead of updating the parameter vector $\hat{\mathbf{v}}_{t-1}$ to $\hat{\mathbf{v}}_t$ the smoothed updating procedure is often used:

$$\hat{\mathbf{v}}_t = \alpha \tilde{\mathbf{v}}_t + (1 - \alpha) \hat{\mathbf{v}}_{t-1}, \quad (11)$$

with $0 \leq \alpha \leq 1$ and $\tilde{\mathbf{v}}_t$ is the solution of Eq. (11). This smoothed adaptation (11) is used to reduce the probability that some component $\hat{v}_{t,i}$ of $\hat{\mathbf{v}}_t$ will be zero or one at the first few iterations. Doing so may lead to premature convergence of the algorithm. As a result, the algorithm could converge to a wrong solution and get stuck in a local optimum.

3.1. The CE method for GLVQ optimization

We next present a CE method for solving the GLVQ optimization problem by viewing it as a continuous multi-extremal optimization problem. As pointed out above, to apply to the CE method to such a problem we have to specify the sampling distribution and the updating rules for its parameters. The choice of the sampling distribution is quite arbitrary. For a continuous optimization problem Kroese et al. [18] suggest to generate samples using the Gaussian, double exponential or beta distributions, for the simplicity of their parameter updates. In this paper, we consider the sampling distribution to be truncated Gaussian with independent components. That is, for each $\theta^l \triangleq \{\theta_{pq}^l\}_{p=1, \dots, d}^{q=1, \dots, P}$ (a $d \times P$ matrix) with $l = 1, \dots, V$, each component

θ_{pq}^l is drawn from a truncated Gaussian distribution such that $\theta_{pq}^l \sim \mathcal{N}^t(m_{pq}^t, (\sigma_{pq}^t)^2, \mathbf{a}_p, \mathbf{b}_p)$, where m_{pq}^t denotes the mean (average) of p qth components at iteration t , $(\sigma_{pq}^t)^2$ the variance and $\mathbf{a}_p, \mathbf{b}_p$ denote a lower and an upper bounding box for each dimension containing all data, respectively. In practice, we choose $\mathbf{a}_p = C \min_{j=1, \dots, N} \{x_{jp}\}$ and $\mathbf{b}_p = C \max_{j=1, \dots, N} \{x_{jp}\}$ with $C \geq 1$.

The CE method parameters $(\mathbf{M}^t \triangleq \{m_{pq}^t\}, (\Sigma^t)^2 \triangleq \{(\sigma_{pq}^t)^2\})$ are updated via the sample mean and sample standard deviation of elite samples (samples that give lowest scores); i.e. those that minimize the $F_{GLVQ}(\mathbf{X}; \theta)$ function. The algorithm is summarized in Fig. 2.

There are many possible variations in the standard stopping criterion; we use the stopping criterion to stop when the best value over t iterations does not decrease by more than a specified tolerance.

To prevent the algorithm from being trapped in local optima, Rubinstein and Kroese [23] proposed to use dynamic smoothing (12)

- (1) Choose some initial $\{\mathcal{M}^0, \Sigma^0\}$ for $p = 1, \dots, d, q = 1, \dots, P$. Set $t = 1$ (level counter).
- (2) Draw samples $\mathbf{W}^l \sim \mathcal{N}(\mathcal{M}^{(t-1)}, \Sigma^{(t-1)}, \mathbf{a}_p, \mathbf{b}_p)$, $l = 1, \dots, V$.
- (3) Compute $S^t = F_{GLVQ}(\mathbf{X}; \theta^l)$ scores by applying (9) $\forall l$.
- (4) Sort S^t in ascending order and denote by \mathcal{I} the set of corresponding indices. Let us denote $(\bar{\mathcal{M}}^{(t-1)}, (\bar{\Sigma}^{(t-1)})^2)$ the mean and the variance of the best $[\rho V]$ prototypes elite samples of $\{\mathbf{W}^{\mathcal{I}(l)}, l = 1, \dots, [\rho V]$ respectively.
- (5) $\bar{\mathcal{M}}^t = \alpha \bar{\mathcal{M}}^t + (1 - \alpha) \bar{\mathcal{M}}^{t-1}$, $\bar{\Sigma}^t = \beta_t \bar{\Sigma}^t + (1 - \beta_t) \bar{\Sigma}^{(t-1)}$
- (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.

Fig. 2. CE method for GLVQ optimization: CEMGLVQ.

Table 1
Comparison of GLVQ cost function of and CEMGLVQ on *Iris* data set

	Min	Max	Mean	Trials	CPU
GLVQ	50.54	50.56	50.55	12	31.8
CEMGLVQ	50.52	50.52	50.52	1	31.04

where at each iteration the variance (σ^2) is updated using a smoothing parameter:

$$\beta_t = \beta_0 - \beta_0 \left(1 - \frac{1}{t}\right)^c, \quad (12)$$

where c is a small integer (typically between 5 and 15) and β_0 is a large smoothing constant (typically between 0.8 and 0.99). By using β_t instead of α , the convergence to the degenerate case has polynomial speed instead of exponential. However, occasionally one or more components of the parameters' variances prematurely converge to zero, perhaps at a local optimum. One way to deal with this is by "injecting" extra variance into the sampling variance parameters, this modification was first proposed in Botev and Kroese [24].

According to Kroese et al. [18], the performance of the 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, to ensure the exploration of the entire solution space. One advantage of the CE method is that, as a global optimization method, it is very robust with respect to the initial positions of the prototypes. Provided that the initial standard deviations are chosen large enough, the initial means have little or no effect on the accuracy and the convergence speed of the algorithm.

In general, we choose the initial means and standard deviations such that the initial sampling distribution is fairly "uniform" over the smallest rectangle that contains the data points. Practically, this means that the initial standard deviations should not be too small, say equal to the width or height of this "bounding box".

4. Experimental results

In this section, we first applied GLVQ and CEMGLVQ algorithms to *Iris* data set [25]. One run of the CEMGLVQ and GLVQ take about 31.04 and 2.65 s, using a Matlab implementation on a 3.2 GHz computer, respectively. In order to make a fair comparison we run the GLVQ algorithm (12 times), so that the total time for all GLVQ runs is no less than the time taken by the CE algorithm (31.8 s).

From Table 1 we can see that our algorithm gives slightly lower cost function value than the GLVQ algorithm. This result gives us an idea about the strength of the CE algorithm to find globally optimization solutions.

We applied the CEMGLVQ algorithm with some machine-learning algorithms (GLVQ, H2MGLVQ) to show the effectiveness of the proposed method. Following standard procedure in experiments with

Table 2

List of real-world data sets used for comparison between algorithms

Name	# Features	# Patterns	# Classes
Liver	6	345	2
Pima	8	768	2
Glass	9	214	6
WBC	10	699	2
Waveform*†	21	500	3

Data sets indicated by † contain features with only noise, while those indicated by * contain intrinsic within-class multi-modal structure.

Table 3

Recognition rates on real data sets

Name	GLVQ (%)	CEMGLVQ (%)	H2MGLVQ (%)
Liver	58.85 ± 1.82	66.98 ± 4.30	65.30 ± 8.10
Pima	72.13 ± 2.87	74.87 ± 4.33	72.90 ± 4.07
Glass	60.07 ± 8.07	69.17 ± 6.49	68.72 ± 9.73
WBC	95.85 ± 2.04	96.28 ± 1.80	96.42 ± 1.54
Waveform*†	92.00 ± 6.74	93.50 ± 5.79	93.00 ± 7.88

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 indicated in boldface.

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 procedure (see Ref. [26]).

We tested these algorithms in real-world data sets taken from the public UCI repository (see Table 2). For comparison, we used the same data as in Qin and Suganthan [8] (*Liver-disorders* and *Glass*). We used different data sets *PIMA* (Pima Indians Diabetes), *WBC* (Wisconsin breast cancer) which cover a broad spectrum of properties occurring frequently in real-world applications. In particular, we used the waveform data set because it contains features with only noise and intrinsic within-class multi-modal structure.

The parameters in the GLVQ algorithms were set as follows:

- GLVQ: learning rate $\varepsilon = 0.05$, $\xi = 0.05$.
- H2MGLVQ: $\xi = 0.05$, $\varepsilon_k = 0.05$, $\varepsilon_l = 0.01(0.05/0.01)^{k/(K-1)}$, where k and K are current epoch and number of iterations, respectively. For these two algorithms, the prototypes for different classes were randomly initialized around the center of the corresponding classes. Number of prototypes per class and the values of parameters are chosen as described in Qin and Suganthan [8].
- For the CEMGLVQ: $V = 500$, $\xi = 0.05$, $\rho = 0.006$, $\beta_0 = 0.8$, $h = 5.10e - 6$, $q = 12$, $\varepsilon = 5.10e - 5$ and $c = 10$, $C = 1.2$. It is found empirically that when α is between 0.6 and 0.9, it gives best results; in our case we chose $\alpha = 0.6$.

For all algorithms, the maximum number of iterations is set to $K = 2000$. We based our comparison on three criteria: the error rate, the variance and optimal value of the cost function, the results are presented in Tables 3 and 4. We see in Table 3 that our algorithm outperforms other algorithms and shows smaller variance, which confirms the accuracy of the proposed method and that the CE Method can be viewed as a variance reduction method [18]. In Table 4 we see that the CEMGLVQ gives the lowest value of the cost function.

Compared to other algorithms based on gradient descent, the CEMGLVQ is more demanding on computational cost of running. The theoretical complexity of CE is an open problem still under investigation. This complexity is partially dependent on the studied problem (see Ref. [27]). The computational complexity of our algorithm is around $O(dNPV)$ in each iteration. The architecture of the CE

Table 4
Optimal cost function values

Name	GLVQ	CEMGLVQ
Liver	154.56 ± 0.26	154.42 ± 0.27
Pima	342.76 ± 2.04	342.33 ± 0.20
Glass	95.39 ± 0.27	95.30 ± 0.26
WBC	308.20 ± 0.15	308.17 ± 0.16
Waveform*†	89.34 ± 0.056	89.05 ± 0.024

The format of the numbers in this table is $M \pm S$, where M is the mean cost function value, S is the standard deviation.

method algorithm being inherently parallel, it requires to evaluate a cost function, consequently the CE method can be accelerated by parallelizing all these evaluations.

The CEMGLVQ is slower than other algorithms; one could object and say if the aim of this work is to make the GLVQ algorithm insensitive to initialization of prototypes, we can overcome this weakness by using algorithms based on gradient descent like H2MLVQ, since this algorithm appears to work, but our algorithm also outperforms this version.

5. Conclusion

In this paper we considered solving the GLVQ initialization sensitivity problem. We formulated it as an 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 similar to other proposed methods.

The main benefit of the proposed method is its insensitivity to the choice of its hyperparameters, not the case for the other methods SRNG and H2MLVQ; for example, the learning rate has to be chosen for both and the size of an update neighborhood for SRNG. One question that comes to mind that we have to answer is why we chose the CE method and not another optimization method (simulated annealing (SA), genetic algorithms, etc.). In general, the CE algorithm is efficient and easy to implement. It is an iterative procedure, in each iteration samples are generated according to a certain probability distribution and then some elite samples are selected to update the distribution parameters. The CE can be seen as a stochastic method, which gives CE the ability to find more global optima than deterministic methods. Compared with other stochastic solvers like SA, CE is obviously more efficient because it focuses on a few high-performance samples (elite samples) among a large collection of samples and quickly converges good solutions. Unlike genetic algorithms the CE does not need to define unnatural operators. Finally, similar to SA the CE method is guaranteed to converge to the optimal solution in the large sample limit. In future work we will concentrate on the investigation of more appropriate cost function which do not only take into account the two best matching prototypes (form the correct and the wrong class) but prototypes cooperation (for example, H2MLVQ). The second drawback of GLVQ algorithm is its use of the Euclidian metric which is not always appropriate because it supposes that all the attributes contribute equally to the classification; we propose to replace the Euclidian metric by a weighted Euclidian metric by introducing input weights to allow a different scaling of the inputs as suggested by Hammer et al. [28]. As a future direction of this work, we propose the use of

a sophisticated version of CE by considering the fully adaptive CE (FACE) variant, where the parameters N and ρ are updated online.

References

- [1] T. Kohonen, Learning vector quantization for pattern recognition, Technical Report TKK-F-A601, 1986.
- [2] T. Kohonen, Improved versions of learning vector quantization, INJCN, 1990.
- [3] M. Pregoner, D. Flotzinger, G. Pfurtscheller, Distinction sensitive learning vector quantization—a noise-insensitive classification method, in: IEEE International Joint Conference on Neural Networks, vol. 10, 1994, pp. 2890–2894.
- [4] P. Somervuo, T. Kohonen, Self-organizing maps and learning vector quantization for feature sequences, Neural Process. Lett. 10 (1999) 151–159.
- [5] T. Bojer, B. Hammer, D. Schunk, K.V. Toschanowitz, Relevance determination in learning vector quantization, in: The European Symposium on Artificial Neural Networks, 2001, pp. 271–276.
- [6] B. Hammer, T. Villman, Generalized relevance learning vector quantization, Neural Networks 15 (8–9) (2002) 1059–1068.
- [7] A.S. Sato, K. Yamada, A formulation of learning vector quantization using a new misclassification measure, in: The 14th International Conference on Pattern Recognition, 1998.
- [8] A. Qin, P. Suganthan, Initialization insensitive LVQ algorithm based on cost-function adaptation, Pattern Recognition 38 (5) (2005) 773–776.
- [9] B. Hammer, M. Strickert, T. Villmann, Supervised neural gas with general similarity measure, Neural Process. Lett. 21 (1) (2004) 21–44.
- [10] T. Bojer, B. Hammer, C. Koers, Monitoring technical systems with prototype based clustering, in: The European Symposium on Artificial Neural Networks, 2002, pp. 271–276.
- [11] K. Crammer, R. Gilad-Bachrach, A. Navot, N. Tishby, Margin analysis of the LVQ algorithm, in: NIPS'2002, 2002.
- [12] R. Rubinstein, Optimization of computer simulation models with rare events, Eur. J. Oper. Res. 99 (1997) 89–112.
- [13] D.P. Kroese, R.Y. Rubinstein, T. T., Application of the cross-entropy method to clustering and vector quantization, Eur. J. Oper. Res. 37 (2007) 137–157.
- [14] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley, New York, 2001.
- [15] C. Diamantini, A. Spalvieri, Certain facts about Kohonen's LVQ1 algorithm, IEEE Trans. Circuits Syst. I (47) (1996) 425–427.
- [16] B.H. Juang, S. Katagiri, Discriminative learning for minimum error classification, IEEE Trans. Signal Process. 40 (2) (1992) 3043–3054.
- [17] M.C. Fu, F.W. Glover, J. April, Simulation optimization: a review, new developments, and applications, in: The 37th Winter Simulation Conference (WSC2005), 2005, pp. 83–95.
- [18] D. Kroese, S. Porotsky, R. Rubinstein, The cross-entropy method for continuous multi-extremal optimization, Methodol. Comput. Appl. Probab. 8 (2006) 383–407.
- [19] O. Wittner, B.E. Helvik, Cross entropy guided ant-like agents finding dependable primary/backup path patterns in networks, in: Congress on Evolutionary Computation (CEC2002), 2002.
- [20] Z. Liu, A. Doucet, S.S. Singh, The cross-entropy method for blind multiuser detection, in: IEEE International Symposium on Information Theory, 2004.
- [21] P.D. Boer, D. Kroese, S. Mannor, R.Y. Rubinstein, A tutorial on the cross-entropy method, Ann. Oper. Res. (2005) 19–67.
- [22] K. Chepuri, T.H. de Mello, Solving the vehicle routing problem with stochastic demands using the cross entropy method, Ann. Oper. Res. 134 (2005) 153–181.
- [23] R.Y. Rubinstein, D.P. Kroese, The Cross-Entropy Method: A Unified Approach to Combinatorial Method, Monte-Carlo Simulation, Randomized Optimization and Machine Learning, Springer, Berlin, 2004.
- [24] Z. Botev, D. Kroese, Global likelihood optimization via the cross-entropy method, with an application to mixture models, in: Proceedings of the 2004 Winter Simulation Conference, 2004.
- [25] D.J. Newman, S. Hettich, C.L. Blake, C.J. Merz, Uci Repository of Machine Learning Databases. URL: (<http://www.ics.uci.edu/~mllearn/MLRepository.html>), 1998.
- [26] X. Zeng, T.R. Martinez, Distribution-balanced stratified cross-validation for accuracy estimation, J. Exp. Theor. Artif. Intell. 12 (2000) 1–12.
- [27] J. Wu, A. Chung, Cross entropy: a new solver for Markov random field modeling and applications to medical image segmentation, in: The Eighth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'05), 2005, pp. 229–237.
- [28] B. Hammer, M. Strickert, T. Villmann, Relevance LVQ versus svm, in: Seventh International Conference on Artificial Intelligence Soft Computing (ICAISC 2004), 2004, pp. 592–597.

About the author—ABDERRAHMANE BOUBEZOUL, born in Algeria on July 28, 1976, received the Diplôme d'Etude Approfondies in Réalité Virtuelle et Maitrise des Systèmes Complexes from the Evry Val d'Essone University (France) in 2004. He has been at the université d'aix-marseille 3 where he is a Ph.D. Student in Computer Sciences and Mathematics. His current work is about statistical signal processing and machine learning.

About the author—SÉBASTIEN PARIS, born in France on December 12, 1973, received the Diplôme d'Etude Approfondies in Telecommunication, Propagation and Teledetection from the Nice University (Sophia-Antipolis, France) in 1996, the title of "Docteur de l'Université" in 2000 from

the Université de Toulon et du Var (France). From May 2000 to July 2001, he worked on tracking and vehicle path's planning problems with particle filter, PCRB and cross-entropy methods at IRISA. From 2002 to 2005, he joined SOPRAGROUP. Since February 2005, he has been at the université d'aix-marseille 2 where he teaches Computer Sciences and Telecommunications. His current work is about statistical signal processing and machine learning.

About the author—MUSTAPHA OULADSINE received his Ph.D. in Nancy 1993 in the estimation and identification of nonlinear systems. In 2001 he joined the laboratory of sciences of information's and of systems in Marseille France. His research interests include nonlinear estimation and identification, neural networks, diagnosis and control and their applications in the vehicle and aeronautic domains, he published more than 50 technical papers.