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
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 hyperparameters 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 (CEGLVQ).

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(x; θ), which transforms objects x from the input space Ξ to the output space Φ.

The data domain Ξ and the set of target values Φ are determined by the definition of the problem for which the f(x; θ) is constructed. The output of the function can be a continuous value (for regression applications) or can predict a class label of the input object (for classification applications). In Bayesian decision theory, the optimal classifier is obtained by minimizing the risk computed as follows:

\[ R(θ) = \mathbb{E}[L(f(x; θ); x)] = \int L(f(x; θ); x) \cdot p(x) \, dx, \]  

where L(f(x; θ); x) is the expected loss defined by

\[ L(f(x; θ); y) = \int \ell(f(x; θ); y) \cdot p(y| x) \, dx, \]  

where \( \ell(f(x; θ); y) \) denotes the individual loss.

By rewriting Eq. (1), we have

\[ R(θ) = \sum_{k=1}^{L} p(y = k) \int \ell(f(x; θ); y) \cdot p(y = k) \, dx. \]  

We assume that the training data are drawn from some underlying distribution \( p(x|y) \), which is in practice unknown or not trivial to estimate. In real applications, only a finite number of samples are available. The risk functional \( R(θ) \) is usually replaced by the empirical risk computed as follows:

\[ R_{\text{emp}}(θ) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{L} \ell(f(x_i; θ); y_i = k) \cdot 1(y_i = k), \]  

where \( 1(\cdot) \) is an indicator function such that \( 1(y = k) = 1 \) if the condition between the parentheses is satisfied or \( 1(y = k) = 0 \) 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 \( w_{kj} \in Ξ \) is the jth prototype among \( P_k \) vectors associated with class \( k = 1, \ldots, L \) and let \( θ_k \) be the collection of all prototypes of the class \( k \). The collection of all prototypes representing all classes is defined as \( θ \equiv \bigcup_{k=1}^{L} θ_k \), and \( P = \sum_{k=1}^{L} P_k \) denotes the total number of prototypes.

Suppose that \( x \) 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:

- \( g_k(x; θ) = -d_k(x; θ) \equiv -\min_{l \neq k} \|x - w_{jl}\|^2 \), where \( d_k(x; θ) \) is the squared Euclidean distance between \( x \) and the closest prototype belonging to the same class as \( x \).
- \( g_l(x; θ) = -d_l(x; θ) \equiv -\max_{l \neq k} \|x - w_{jl}\|^2 \).

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

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

\[ μ_k(x; θ) \equiv g_k(x; θ) + \left[ \frac{1}{L - 1} \sum_{j \neq k} g_j(x; θ) \right]^{1/η}, \]  

where \( η \) is a positive constant. When \( η \to \infty \), the misclassification measure becomes \( μ_k(x; θ) = -d_k(x; θ) + \max_{l \neq k} g_l(x; θ) = d_l(x; θ) - d_l(x; θ) \).

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 \( f_k(x; θ) \) is defined by

\[ f_k(x; θ) = d_k(x; θ) - d_l(x; θ), \]  

where \( d_l(x; θ) \) is the squared Euclidean distance between the input vector \( x \) and its best matching prototype vector \( w_{lr} \), with a different class label.

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

\[ R_{\text{emp}}(θ) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{L} \ell(f_k(x_i; θ); y_i = k), \]  

where \( \ell(f_k(x_i; θ); y_i = k) \) is the indicator function.
Here, $\ell(f_k(x; \theta))$ is a smoothed loss function instead of the 0−1 loss function in Bayesian decision theory, for example, the sigmoid function defined by
\[
\ell(z) = \frac{1}{1 + e^{-z}}.
\] where $z$ is a scalar (which usually increases with time). When $t \to \infty$, the loss function will be identical to empirical loss in Bayesian decision theory. From Eq. (7), the general GLVQ cost function is expressed as
\[
F_{\text{GLVQ}}(X; \theta) = \sum_{i=1}^{N} \frac{1}{N} \ell(f_k(x_i; \theta)).
\]
where $X = \{x_i\}_{i=1,...,N}$.

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 $\Theta$ 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 standard stopping criterion. 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 $\theta \in \Theta$, $\theta_p \sim \mathcal{N}(\theta_p, \sigma^2_{\theta_p})$ with $\theta_p = (\theta_{p1}, ..., \theta_{pM})$ with $M$ the variance and $\theta_p$ a lower and an upper bounding box for each dimension containing all respective values. In practice, we choose $\theta_{p1} = \min_{j=1,...,M} \theta_j$ and $\theta_{pM} = \max_{j=1,...,M} \theta_j$ with $C \geq 1$.

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 $p(\cdot; \theta_{t-1})$, then the new value of $\theta_t$ is updated according to $\theta_{t-1}$ and the best elite samples. The update formula is especially simple if $p(\cdot; \theta)$ 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 $\vec{\theta}_{t-1}$ to $\vec{\theta}_t$ the smoothed updating procedure is often used:
\[
\vec{\theta}_t = \alpha \vec{\theta}_t + (1 - \alpha) \vec{\theta}_{t-1},
\]
with $0 < \alpha < 1$ and $\vec{\theta}_t$ is the solution of Eq. (11). This smoothed adaptation (11) is used to reduce the probability that some component $\vec{\theta}_{t-1}$ of $\vec{\theta}$ 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 $\theta \in \Theta$, $\theta_p \sim \mathcal{N}(\theta_p, \sigma^2_{\theta_p})$ with $\theta_p = (\theta_{p1}, ..., \theta_{pM})$ with $M$ the variance and $\theta_p$ a lower and an upper bounding box for each dimension containing all respective values. In practice, we choose $\theta_{p1} = \min_{j=1,...,M} \theta_j$ and $\theta_{pM} = \max_{j=1,...,M} \theta_j$ with $C \geq 1$.

The CE method parameters $(\mathcal{M} \equiv \{|m_{\theta_p}|, \sigma^2_{\theta_p}\}, (\mathcal{Z}^2 \equiv \{|z_{\theta_p}^2|\})$ are updated via the sample mean and sample standard deviation of elite samples (samples that give lowest scores); i.e. those that minimize the $F_{\text{GLVQ}}(X; \theta)$ 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)
where at each iteration the variance \( (\sigma^2) \) is updated using a smoothing parameter:

\[
\beta_t = \beta_0 - \beta_0 \left( 1 - \frac{1}{T} \right)^c,
\]

where \( c \) is a small integer (typically between 5 and 15) and \( \beta_0 \) is a large smoothing constant (typically between 0.8 and 0.99). By using \( \beta_t \) instead of \( \sigma \), 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 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 \( \epsilon = 0.05, \xi = 0.05 \).
- H2MGLVQ: \( \xi = 0.05, \epsilon_k = 0.01/0.05/0.01/(K-1) \), 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: \( V = 500, \xi = 0.05, \rho = 0.006, \beta_0 = 0.8, h = 5.10e-6, q = 12, \epsilon = 5.10e-5 \) and \( c = 10, C = 1.2 \). It is found empirically that when \( x \) is between 0.6 and 0.9, it gives best results; in our case we chose \( x = 0.6 \).

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
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 $\rho$ are updated online.

References

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.