Neural Networks 26 (2012) 159–173

Contents lists available at SciVerse ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Limited Rank Matrix Learning, discriminative dimension reduction and visualization

Kerstin Bunte [a,*], Petra Schneider [b], Barbara Hammer [c], Frank-Michael Schleif [c], Thomas Villmann [d], Michael Biehl [a]

[a] University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science, The Netherlands

[b] University of Birmingham, School of Clinical and Experimental Medicine, United Kingdom

[c] Bielefeld University, Center of Excellence-Cognitive Interaction Technology CITEC, Germany

[d] University of Applied Sciences Mittweida, Department of MPI, Germany

ARTICLE INFO

Article history:
Received 10 October 2010
Received in revised form 13 September 2011
Accepted 7 October 2011

Keywords:
Learning Vector Quantization
Classification
Visualization
Adaptive metrics
Dimension reduction

ABSTRACT

We present an extension of the recently introduced Generalized Matrix Learning Vector Quantization algorithm. In the original scheme, adaptive square matrices of relevance factors parameterize a discriminative distance measure. We extend the scheme to matrices of limited rank corresponding to low-dimensional representations of the data. This allows to incorporate prior knowledge of the intrinsic dimension and to reduce the number of adaptive parameters efficiently.

In particular, for very large dimensional data, the limitation of the rank can reduce computation time and memory requirements significantly. Furthermore, two- or three-dimensional representations constitute an efficient visualization method for labeled data sets. The identification of a suitable projection is not treated as a pre-processing step but as an integral part of the supervised training. Several real world data sets serve as an illustration and demonstrate the usefulness of the suggested method.

1. Introduction

Learning Vector Quantization (LVQ) (Kohonen, 2001) and its variants constitute a popular family of supervised, prototype-based classifiers. These algorithms have been employed successfully in a variety of scientific and commercial applications, including image analysis, bioinformatics, robotics, etc. (Biehl, Ghosh, & Hammer, 2007; Bojer, Hammer, Schunk, & von Toschanowitz, 2001; Bunte, Biehl, Petkov, & Jonkman, 2009; Bunte, Hammer, Schneider, & Biehl, 2009; Bunte, Hammer, Wismüller, & Biehl, 2010a; Hammer, Strickert, & Villmann, 2005a; Hammer & Villmann, 2002; Schneider, Biehl, & Hammer, 2009; Villmann, Merenyi, & Hammer, 2003). The method is easy to implement and its complexity is controlled by the user in a straightforward way. LVQ can be applied to multi-class problems without further complication and the resulting classifiers can be interpreted intuitively. This is due to the fact that the classification of data points is based on distances to typical representatives, i.e. prototypes, which are identified in feature space.

Numerous modifications of Kohonen's original, heuristic formulation of LVQ have been suggested in the literature, aiming at better convergence properties and generalization behavior. For instance, Sato and Yamada (1996) propose an algorithm, termed Generalized Learning Vector Quantization (GLVQ), which updates prototypes by means of gradient descent with respect to a heuristically motivated cost function. Recently, also kernelized versions have been proposed (Schleif, Villmann, Hammer, Schneider, & Biehl, 2010). A key issue in all LVQ algorithms, with or without an underlying cost function, is the choice of an appropriate similarity or distance measure. Most frequently, standard Euclidean or Minkowski metrics are employed, which are not necessarily appropriate for the given problem and data set. The fact that features can have very different meaning and magnitude in heterogeneous data, is accounted for in so-called relevance learning schemes (Bojer et al., 2001; Hammer, Strickert, & Villmann, 2005b; Hammer & Villmann, 2002) which employ adaptive scaling factors for each dimension in feature space.

An important extension of this concept has been introduced in Schneider et al. (2009): in the so-called Generalized Matrix LVQ (GMLVQ) a full matrix of relevances is used, which can account for correlations between different features. An adaptive self-affine transformation Ω of feature space identifies the coordinate system which is most suitable for the given classification task. The original formulation of GMLVQ employs symmetric squared matrices. In the simplest case, one matrix is taken to define a global distance measure. Extensions to class-wise or local matrices, attached to

* Corresponding author.
E-mail address: k.bunte@rug.nl (K. Bunte).
URL: http://www.cs.rug.nl/~kbunte/ (K. Bunte).

0893-6080/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2011.10.001

individual prototypes, are technically straightforward and allow for the parameterization of more complex decision boundaries.

Here we present and discuss an important modification: the use of rectangular transformation matrices Ω. The corresponding relevance matrices are of bounded rank or, in other words, distances are evaluated in a space with reduced dimension. The motivation for considering this variation of GMLVQ is at least twofold: (a) prior knowledge about the intrinsic dimension of the data can be incorporated efficiently and (b) the number of free parameters in the learning problem may be reduced significantly.

Although unrestricted GMLVQ displays a tendency to reduce the rank of the relevance matrices in the training process, the advantages of restricting the rank explicitly are obvious. In particular for nominally very high-dimensional data, e.g. in image analysis or bioinformatics, unrestricted relevance matrices become intractable. In addition, optimization results can be poor when the search is performed in an unnecessarily large parameter space. Furthermore, the exact control of the rank allows for pre-defining the dimension of the intrinsic representation and is, for instance, suitable for the discriminative visualization of labeled data sets. In contrast with many other schemes that consider dimension reduction as a pre-processing step, our method performs the training of prototypes and the identification of a suitable transformation simultaneously. Hence, both sub-tasks are guided by the ultimate goal of implementing the desired classification scheme.

Appropriate projections into two- or three-dimensional spaces can furthermore be used for efficient visualization of labeled data. Visualization enables to use the astonishing cognitive capabilities of humans for visual perception when extracting information from large data volumes. Structural characteristics can be captured almost instantly by humans, independent of the number of displayed points. Classical unsupervised dimension reduction techniques represent data points contained in a high dimensional data manifold by low dimensional counterparts in, for instance, two or three dimensions, while preserving as much information as possible. Since it is not clear in advance which parts of the data are relevant to the user, this problem is inherently ill-posed: depending on the specific data domain and the situation at hand, different aspects can be in the focus of attention. Prior knowledge, in the form of label information, can be used to formulate a well-defined objective in terms of the classification performance.

There exist a few classical dimensionality reducing visualization tools which take class labels into account: Classical Fisher linear discriminant analysis (LDA), the recently introduced local Fisher discriminant analysis (LFDA) (Sugiyama & Roweis, 2007), Neighborhood Component Analysis (NCA) (Goldberger, Roweis, Hinton, & Salakhutdinov, 2004), as well as partial least squares regression (PLS) offer supervised linear visualization techniques. Kernel techniques extend these settings to nonlinear projections (Baudat & Anouar, 2000; Ma, Qu, & Wong, 2007). Adaptive dissimilarity measures which modify the metric used for projection according to the given auxiliary information have been introduced in Kaski, Sinkkonen, J, and Peltonen (2001), Peltonen, Klami, and Kaski (2004), and Bunte et al. (2010a). The resulting metric can be integrated into various techniques such as SOM, MDS, or a recent information theoretic model for data visualization (Kaski et al., 2001; Peltonen et al., 2004; Venna, Peltonen, Nybo, Aidos, & Kaski, 2010). An ad hoc metric adaptation is used in Geng, Zhan, and Zhou (2005) to extend Isomap (Tenenbaum, Silva, & Langford, 2000) to class labels. Alternative approaches change the cost function of dimensionality reduction, for instance by using conditional probabilities, class-wise similarity matrices or introducing a covariance-based coloring matrix for the side information as proposed in Iwata et al. (2007), Memisevic and Hinton (2005), and Song, Smola, Borgwardt, and Gretton (2008).

Before we describe our method more formally in Section 3 we review GMLVQ in the following section. In Section 4, we apply the novel LiRaM LVQ to a benchmark problem and study the influence of the dimension reduction on the classification performance. We also compare the limited rank version to the naive approach of taking the first components of the full rank GMLVQ. We show that reducing the rank after training not only requires more memory and CPU time, but also yields inferior classification performance compared to LiRaM LVQ. In Section 5 we present example applications of our algorithm in the visualization of labeled data. We also compare with visualizations obtained by LFDA and NCA. We conclude by summarizing our findings and providing an outlook on perspective investigations.

2. Review of Generalized Matrix LVQ

In this section we briefly review the Generalized Matrix LVQ algorithm (Schneider et al., 2009). We will assume that training is based on n examples of the form $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^N \times \{1, \ldots, C\}$, where N is the dimension of feature vectors and C is the number of classes. Learning Vector Quantization (LVQ) parameterizes the classification by means of at least C prototypes, which are chosen as typical representatives of the respective classes. They are characterized by their location in feature space $\boldsymbol{w}_i \in \mathbb{R}^N$ and the respective class label $c(\boldsymbol{w}_i) \in \{1, \ldots, C\}$. Given a distance measure $d^{\Lambda}(\boldsymbol{w}, \boldsymbol{x})$ in $\mathbb{R}^N$ parameterized by Λ, the classification is done according to a "winner takes all" or "nearest prototype" scheme: Any data point $\boldsymbol{x} \in \mathbb{R}^N$ is assigned to the class label $c(\boldsymbol{w}_i)$ of the closest prototype i with $d^{\Lambda}(\boldsymbol{w}_i, \boldsymbol{x}) \leq d^{\Lambda}(\boldsymbol{w}_j, \boldsymbol{x})$ for all $j \neq i$.

Frequently, learning corresponds to an iterative procedure which presents a single example at a time and which moves prototypes closer to (away from) data points representing the same (a different) class. In Sato and Yamada (1996) a very flexible approach is introduced, in which the training algorithm is guided by the minimization of a cost function

$$f = \sum_i \Phi(\mu) = \sum_i \Phi\left(\frac{d_J^{\Lambda} - d_K^{\Lambda}}{d_J^{\Lambda} + d_K^{\Lambda}}\right), \tag{1}$$

where the quantities

$$d_J^{\Lambda} = d^{\Lambda}(\boldsymbol{w}_J, \boldsymbol{x}_i) \quad \text{with } c(\boldsymbol{w}_J) = c(\boldsymbol{x}_i) \tag{2}$$

$$d_K^{\Lambda} = d^{\Lambda}(\boldsymbol{w}_K, \boldsymbol{x}_i) \quad \text{with } c(\boldsymbol{w}_K) \neq c(\boldsymbol{x}_i) \tag{3}$$

correspond to the distances of the feature vector $\boldsymbol{x}_i$ from the closest *correct (wrong)* prototype $\boldsymbol{w}_J(\boldsymbol{w}_K)$, respectively. In Eq. (1), Φ is a monotonic function, e.g. the logistic function or the identity $\Phi(x) = x$ which we will consider throughout the following.

In GMLVQ the distance measure is specified by an $(N \times N)$ matrix, which can adapt to correlations of different features. It is of the form of a Mahalanobis distance

$$d^{\Lambda}(\boldsymbol{w}, \boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{w})^{\top} \Lambda (\boldsymbol{x} - \boldsymbol{w}) \tag{4}$$

with $\Lambda \in \mathbb{R}^{N \times N}$. The matrix Λ is assumed to be positive (semi-) definite. Hence, the measure corresponds to a (squared) Euclidean distance in an appropriately transformed space and we can substitute

$$\Lambda = \Omega^{\top} \Omega \quad \text{with } \Omega \in \mathbb{R}^{N \times N} \tag{5}$$

and, hence

$$d^{\Lambda}(\boldsymbol{w}, \boldsymbol{x}) = [\Omega \ (\boldsymbol{x} - \boldsymbol{w})]^2 \tag{6}$$

with an arbitrary matrix Ω. Specific restrictions may be imposed on Ω without loss of generality. Note that, for instance, every positive symmetric Λ has a symmetric root Ω with $\Lambda = \Omega^2$.