Neurocomputing 147 (2015) 96–106

Contents lists available at ScienceDirect

Neurocomputing

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

Efficient approximations of robust soft learning vector quantization for non-vectorial data

Daniela Hofmann *, Andrej Gisbrecht, Barbara Hammer

CITEC Center of Excellence, Bielefeld University, Germany

ARTICLE INFO

Article history:
Received 26 March 2013
Received in revised form
20 November 2013
Accepted 30 November 2013
Available online 9 June 2014

Keywords:
Classification
RSLVQ
Kernel
Nyström
Sparse

ABSTRACT

Due to its intuitive learning algorithms and classification behavior, learning vector quantization (LVQ) enjoys a wide popularity in diverse application domains. In recent years, the classical heuristic schemes have been accompanied by variants which can be motivated by a statistical framework such as robust soft LVQ (RSLVQ). In its original form, LVQ and RSLVQ can be applied to vectorial data only, making it unsuitable for complex data sets described in terms of pairwise relations only. In this contribution, we address kernel RSLVQ which extends its applicability to data which are described by a general Gram matrix. While leading to state of the art results, this extension has the drawback that models are no longer sparse, and quadratic training complexity is encountered due to the dependency of the method on the full Gram matrix. In this contribution, we investigate the performance of a speed-up of training by means of low rank approximations of the Gram matrix, and we investigate how sparse models can be enforced in this context. It turns out that an efficient Nyström approximation can be used if data are intrinsically low dimensional, a property which can be efficiently checked by sampling the variance of the approximation prior to training. Further, all models enable sparse approximations of comparable quality as the full models using simple geometric approximation schemes only. We demonstrate the behavior of these approximations in a couple of benchmarks.

1. Introduction

Learning vector quantization (LVQ) as proposed by Kohonen [17] more than 20 years ago still constitutes a popular and widely used classification scheme, in particular due to its intuitive training algorithm and classification behavior. The fact that the classifier represents its classification prescription in a compact way in terms of a small number of prototypical representatives enables its applicability in particular in the medical domain, where human insight is often crucial, or in online learning scenarios such as online vision systems where a compact representation of the already gathered information is required for further adaptation [1,2,16,8,15]. While original LVQ has been proposed on heuristic grounds, mimicking learning paradigms in biological systems, quite a few variants have been proposed in the last years which can be derived based on mathematical cost functions. Notably, generalized LVQ [23] relies on a cost function which can be linked to large margin classifiers [24], enabling a particularly robust classification scheme. As an alternative, robust soft LVQ (RSLVQ) models the data in terms of a mixture of Gaussians in a probabilistic framework. Training can be derived thereof as likelihood ratio optimization [26]. Interestingly, both variants yield to training algorithms which are very similar to original LVQ2.1 as proposed by Kohonen [17]. The formulation as cost function allows to easily integrate a larger flexibility into the prescriptions such as the concept of metric learning [24,26].

Note that LVQ schemes are in some sense complementary to popular classification schemes as provided e.g. using support vector machines (SVM): while both techniques constitute large margin approaches thus providing excellent generalization ability, one of the strengths of SVM is its very robust behavior due to a convex cost function with unique solutions. LVQ, on the contrary, typically possesses local optima, and optimization using gradient techniques is usually necessary. However, while SVM represents models in terms of support vectors, which constitute points at the boundary, the number of which typically scales with the size of the training set, LVQ represents solutions in terms of few typically prototypes only, resulting in an improved interpretability and classification time. On the down-side, SVM can often represent the boundaries in more detail because of its focus on the boundaries, while LVQ classifiers stay with more simple models. Because of the need of interpretable models in domains such as biomedical applications where the ultimate responsibility lies with the human applicant, however, sparse interpretable models

* Corresponding author.
E-mail address: dhofmann@techfak.uni-bielefeld.de (D. Hofmann).

http://dx.doi.org/10.1016/j.neucom.2013.11.044
0925-2312/

such as LVQ classifiers enjoy an increasing popularity among practitioners.

In this contribution, we will focus on the approach robust soft LVQ as proposed in [26] since it offers an intuitive representation of data in terms of a mixture of labeled Gaussians. Being a prototype based approach, LVQ provides a direct interface for the applicant, who can directly inspect the prototypes in the same way as data. Regarding the crucial impact of interpretability of the given models in many fields, this fact constitutes an important benefit of LVQ classifiers [28].

In many application areas, data sets are becoming more and more complex and additional structural information is often available. Examples include chemical structures, biological networks, social network data, graph structures, dedicated images, and heterogeneous web data. Often, dedicated similarity measures have been developed to compare such data; popular examples for widely used dissimilarity or similarity measures for such objects are dynamic time warping for time series, alignment for biological sequences or text, divergences for distributions, functional metrics for functional data such as spectral data, graphs or tree kernels for structured objects, and many more. These data are no longer explicitly represented as Euclidean vectors, rather, pairwise similarities or dissimilarities are available.

LVQ in its original form has been proposed for vectorial data only, since it heavily relies on the possibility to pick prototypes as members of the data space and to adapt these representatives smoothly by means of vectorial updates triggered by the data. Hence LVQ is not directly applicable to complex domains where data are represented in terms of pairwise relations only.

In the last years, a few approaches have been developed which extend LVQ schemes or, more generally, prototype based approaches beyond the vectorial setting. Thereby, most techniques rely on an underlying cost function for which an alternative optimization scheme in the non-vectorial setting is proposed. As an example, unsupervised prototype based methods can rely on exemplars, i.e. they restrict the location of prototypes to the position of given data points, where dissimilarities are well defined. Training takes place in a discrete space, partially relying on appropriate assignment probability to achieve greater robustness, see e.g. the approaches [18,7,4]. These techniques, however, have the drawback that a smooth adaptation of prototypes is no longer possible and problems can occur especially if the given data are sparse. More general smooth adaptation is offered by relational extensions such as relational neural gas or relational learning vector quantization [12]. Kernelization constitutes another possibility such as proposed for neural gas, self-organizing maps, or different variants of learning vector quantization [3,22]. Recently, a kernel variant of RSLVQ has been proposed which matches the classification performance of support vector machines in a variety of benchmarks [14]. By formalizing the interface to the data as a general similarity or dissimilarity matrix, complex structures can be dealt with, relying on dedicated structure kernels or an explicit Gram matrix, for example [21,10,9].

In this contribution, we will focus on kernel RSLVQ (KRSLVQ) which will be extensively tested for benchmark data sets in comparison to popular alternatives such as k-nearest neighbor classifiers and the support vector machine. KRSLVQ allows to priorly specify the model complexity, i.e. number of prototypes which represent the classifier.

Kernel RSLVQ, unlike RSLVQ, represents prototypes implicitly by means of a linear combination of data in kernel space. This has two drawbacks: on one hand, prototypes are no longer directly interpretable, since the vector of linear coefficients is usually not sparse. Hence, in theory, all data points can contribute to the prototype. On the other hand, an adaptation step does no longer scale linearly with the number of data points, rather, quadratic complexity is required. This makes the technique infeasible if large data sets are considered. In this contribution, we propose two different approximation schemes and we investigate the effect of these techniques in a variety of benchmarks [13]. First, we consider the Nyström approximation of Gram matrices which has been proposed in the context of SVMs in [29]. It constitutes a low rank approximation of the matrix based on a small subsample of the data. Assuming a fixed size of the subsample, a linear adaptation technique results. This approximation technique accounts for an efficient update, but prototypes are still distributed. As an alternative, we investigate an approximation of prototypes in terms of their k closest exemplars after or while training. This way, sparse models are obtained, albeit the technique still displays quadratic complexity. The effects of these approximations on the accuracy are tested in a couple of benchmarks.

Now we first review RSLVQ and its kernel variant. We explain the Nyström approximation and its incorporation into kernel RSLVQ. Afterwards, we explain different sparse approximations of the prototypes. We test the performance using benchmarks similar to [6].

2. Kernel robust soft learning vector quantization

Robust soft LVQ has been proposed in [26] as a probabilistic counterpart to Learning vector quantization [17]. It models data by a mixture of Gaussians and derives learning thereof by means of a maximization of the log likelihood ratio of the given data. In the limit of small bandwidth, a learning rule which is similar to LVQ2.1 is obtained.

Assume that data $\xi_k \in \mathbb{R}^n$ are given accompanied by labels y_k. A RSLVQ network represents a mixture distribution, which is determined by m prototypes $w_j \in \mathbb{R}^n$, where the labels of prototypes $c(w_j)$ are fixed. σ_j denotes the bandwidth. Then, mixture component j induces the probability

$$p(\xi|j) = \text{const}_j \cdot \exp(f(\xi, w_j, \sigma_j^2)) \tag{1}$$

with normalization constant const_j and function f

$$f(\xi, w_j, \sigma_j^2) = -\|\xi - w_j\|^2/\sigma_j^2. \tag{2}$$

The probability of a data point ξ is given by the mixture

$$p(\xi|W) = \sum_j P(j) \cdot p(\xi|j) \tag{3}$$

with prior probability $P(j)$ of mixture j and parameters W of the model. The probability of a data point ξ and a given label y is

$$p(\xi, y|W) = \sum_{c(w_j)=y} P(j) \cdot p(\xi|j). \tag{4}$$

Learning aims at an optimization of the log likelihood ratio

$$L = \sum_k \log \frac{p(\xi_k, y_k|W)}{p(\xi_k|W)}. \tag{5}$$

A stochastic gradient ascent yields the following update rules, given a data point (ξ_k, y_k)

$$\Delta w_j = \alpha \cdot \begin{cases} (P_y(j|\xi_k) - P(j|\xi_k)) \cdot \text{const}_j \cdot \partial f(\xi_k, w_j, \sigma_j^2)/\partial w_j & \text{if } c(w_j) = y_k \\ -P(j|\xi_k) \cdot \text{const}_j \cdot \partial f(\xi_k, w_j, \sigma_j^2)/\partial w_j & \text{if } c(w_j) \neq y_k \end{cases} \tag{6}$$

$\alpha > 0$ is the learning rate. The probabilities are defined as

$$P_y(j|\xi_k) = \frac{P(j)\exp(f(\xi_k, w_j, \sigma_j^2))}{\sum_{c(w_j)=y_j} P(j)\exp(f(\xi_k, w_j, \sigma_j^2))} \tag{7}$$