Neurocomputing 141 (2014) 84–96

Contents lists available at ScienceDirect

Neurocomputing

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

Learning interpretable kernelized prototype-based models

Daniela Hofmann*, Frank-Michael Schleif, Benjamin Paaßen, Barbara Hammer

CITEC Center of Excellence, Bielefeld University, Germany

ARTICLE INFO

Article history:
Received 8 July 2013
Received in revised form
19 December 2013
Accepted 28 March 2014
Available online 13 April 2014

Keywords:
Kernel learning vector quantization
Proximity data
Sparsity
Interpretable models

ABSTRACT

Since they represent a model in terms of few typical representatives, prototype based learning such as learning vector quantization (LVQ) constitutes a directly interpretable machine learning technique. Recently, several LVQ schemes have been extended towards a kernelized or dissimilarity based version which can be applied if data are represented by pairwise similarities or dissimilarities only. This opens the way towards its application in domains where data are typically not represented in vectorial form. Albeit kernel LVQ still represents models by typical prototypes, interpretability is usually lost this way: since no vector space model is available, prototypes are represented indirectly in terms of combinations of data. In this contribution, we extend a recent kernel LVQ scheme by sparse approximations to overcome this problem: instead of the full coefficient vectors, few exemplars which represent the prototypes can be directly inspected by practitioners in the same way as data in this case. For this purpose, we investigate different possibilities to approximate a prototype by a sparse counterpart during or after training relying on different heuristics or approximation algorithms, respectively, in particular sparsity constraints while training, geometric approaches, orthogonal matching pursuit, and core techniques for the minimum enclosing ball problem. We discuss the behavior of these methods in several benchmark problems as concerns quality, sparsity, and interpretability, and we propose different measures how to quantitatively evaluate the performance of the approaches.

1. Introduction

Due to their intuitive learning and classification rule based on a winner-takes-all scheme, prototype-based techniques such as learning vector quantization (LVQ) enjoy a great popularity in diverse application domains ranging from telecommunication and robotics up to bioinformatics and data mining [32,4,20]. Apart from an only linear training time and its suitability for online scenarios, as demonstrated e.g. in [31,15], one of its benefits is given by the fact that models are represented in terms of few prototypes which can be inspected by practitioners in the same way as data. Hence this inherent representation scheme lends itself as an intuitive interface to the model, unlike many black box alternatives in machine learning which offer state-of-the-art results but, usually, do not provide a justification why a certain classification takes place [1]. In complex settings where the overall task is not necessarily clear a priori or in settings where the human has to take responsibility for a subsequent action, interpretability becomes crucial: here, human insight is often the only way to further specify a priorly unclear training setting or to substantiate mere observations by causalities. Due to this reason, there is an increasing demand of interpretable models which provide a human understandable interface to their decisions besides excellent classification accuracy in areas such as biomedical data analysis or interactive data inspection [56].

Recently, quite a few approaches have addressed the interpretability of powerful machine learning algorithms, including, for example, intelligent approximation techniques and feature selection mechanisms for SVM, blind signal separation, enhanced score methods, or visualization techniques [44,54,8,53,23]. One prominent example, for which interpretability is guaranteed per the design of the model, is offered by prototype based techniques such as learning vector quantization (LVQ) or generalizations thereof as proposed in [48,50,32,7]. LVQ relies on prototypical class representatives as model parameters. Decisions are taken based on the distance of a data point and the prototypes by means of a winner-takes-all rule. Interestingly, some LVQ techniques can be easily enhanced such that they provide an inherent low dimensional visualization of their decisions [11], or an extension of the models by directly interpretable relevance terms is possible [49,48]. Further, very strong learning theoretical guarantees substantiate LVQ algorithms as classification models with excellent generalization behavior [3,5,49].

Classical LVQ methods are restricted to vectorial data such that they cannot be applied if data are non-vectorial and represented in terms of pairwise similarities or dissimilarities. Examples for such

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

http://dx.doi.org/10.1016/j.neucom.2014.03.003
0925-2312/

settings include structured data such as graphs, trees, sequence data, XML, or the like [17,19,46]. Often, these data can be addressed by means of a dedicated similarity measure or kernel, including e.g. sequence alignment, the normalized compression distance, graph kernels, or similar [19,13,12,40,26,30,34,35]. As such, the similarity or dissimilarity measure can serve as a canonical interface of the model towards the given data set, as is the case e.g. in popular kernel approaches.

Several extensions of prototype methods to general distances or kernels have recently been proposed, see e.g. [33,14,24,9,42,29,18,27,38,41]. The key problem which is addressed in these approaches is the definition of a space where prototypes can be represented since no embedding vector space is explicitly available for this purpose. Some of these approaches restrict the prototype locations to exemplars, i.e. data points, and adapt prototypes within this discrete set. Alternatives rely on an implicit embedding of the data in a kernel space, or, more generally, pseudo-Euclidean space or Krein space, in which vector operations can be done [39]. Concrete learning algorithms usually provide means of how to perform this embedding implicitly by means of kernelization or relationalization. This technique results in methods which have squared complexity as opposed to cubic complexity for an explicit embedding result. Interestingly, approximation techniques as proposed in [21,24,47] can improve the complexity to linear time. While exemplar based techniques often suffer from the restricted numerical flexibility, relational or kernel approaches in particular have obtained results which are competitive to state-of-the-art alternatives such as SVM [29,25].

For kernel LVQ schemes, one important property of prototype-based techniques is lost: prototypes are no longer given as explicit points in the data space, rather, an indirect representation as a linear combination of an underlying (usually not explicitly given) feature space is used. Thus, interpretability of the models, one of the main benefits of LVQ techniques, is no longer given. In this contribution, we address the question how to get around this problem by means of sparse approximations of prototypes. In this case, prototypes are represented by one or few exemplars only, whereby the latter can be directly inspected by practitioners in the same way as data. At the same time, training benefits from the larger flexibility of a continuous adaptation space as provided by the full model.

The principle of sparsity constitutes a common paradigm in nature-inspired learning, as discussed e.g. in the seminal work [37]. Interestingly, apart from an improved complexity, sparsity can often serve as a catalyzer for the extraction of semantically meaningful entities from data. In our case, the basic entities are represented by the data itself, and the task is to approximate given prototypes by sparse counterparts, thereby minimizing the loss of accuracy. It is well known that the problem of finding smallest subsets of coefficients such that a set of linear equations can still be fulfilled constitutes an NP hard problem, being directly related to NP-complete subset selection. Because of this fact, approximation techniques have to be considered, one popular approach being e.g. a l_1-relaxation of the problem [16] as used in LASSO.

In this contribution, we propose a few possibilities to approximate prototypes in a classical LVQ scheme by sparse approximations, thereby partially relying on classical solutions, but also taking into account simple heuristics which are motivated by the underlying geometrical background. Thereby, we propose one technique which emphasizes sparsity already while training, comparing this to two mathematical approximation schemes of the representation, namely classical orthogonal matching pursuit [10] and core techniques to approximately solve the minimum enclosing ball problem for the receptive fields of prototypes. As an alternative, we investigate two simple heuristics: an approximation of the prototypes by their closest exemplars, and a simple numerical rounding of the coefficient vector obtained by full training. We investigate the performance of these different techniques as concerns their classification accuracy and degree of sparsity. As one quantitative measure which can be related to the model interpretability, we use Rissanen's description length principle in a supervised setting as well as the overall data entropy to judge the representativity of prototypes in an unsupervised perspective [43].

Now we first introduce robust soft learning vector quantization (RSLVQ) as a LVQ scheme based on a statistical model where training can be derived as likelihood ratio optimization [50], and its extension towards general kernels [25,29]. Afterwards, we introduce different sparse approximation schemes for the representation of prototypes. We test the approaches using different benchmarks from similarity based learning [12] and evaluate the degree of sparsity obtained in the diverse approaches as well as their accuracy. We conclude with an interpretation of the results in the light of the data signature.

2. Kernel robust soft learning vector quantization

LVQ as originally proposed by Kohonen constitutes a very intuitive classifier which bases its decision on a winner-takes-all scheme and its learning rule on variants of Hebbian learning. Original LVQ 1 is surprisingly good in typical model situations as investigated e.g. in [5], but its adaptation rule is based on heuristic grounds only and cannot be interpreted as direct optimization of a valid cost function [6]. One of the first proposals of an underlying cost function related to large margin maximization can be found in [45], see e.g. [28,49] for a corresponding proof. The alternative proposal presented in [50] takes the perspective of generative models by relying on a mixture of Gaussians. A learning rule similar to LVQ2.1 can be derived thereof as likelihood ratio maximization.

Formally, assume that data $\xi_i \in \mathbb{R}^n$ are labeled y_i. A trained RSLVQ network represents a mixture distribution characterized by m prototypes $w_j \in \mathbb{R}^n$. The labels of the prototypes $c(w_j)$ are fixed. σ_j denotes the bandwidth. Mixture component j induces $p(\xi|j) = \text{const}_j \cdot \exp(f(\xi, w_j, \sigma_j^2))$ with normalization constant const_j and function $f(\xi, w_j, \sigma_j^2) = -\|\xi - w_j\|^2/\sigma_j^2$. The probability of data point ξ is defined as mixture $p(\xi|W) = \sum_j P(j) \cdot p(\xi|j)$ with prior $P(j)$ and parameters W of the model. The probability of a data point ξ and a given label y is $p(\xi, y|W) = \sum_{c(w_j)=y} P(j) \cdot p(\xi|j)$. Learning aims at an optimization of the log likelihood ratio

$$L = \sum_i \log \frac{p(\xi_i, y_i|W)}{p(\xi_i|W)}.$$

For optimization, usually a stochastic gradient ascent is used which yields update rules similar to LVQ2.1 provided class priors are equal, see [50] for details.

Given a novel data point ξ, its class label is the most likely label y corresponding to a maximum value $p(y|\xi, W) \sim p(\xi, y|W)$. For typical settings, this rule can be approximated by the standard winner-takes-all rule. We refer to the data ξ_i which are closest to a given prototype w_j as the receptive field R_j of the prototype.

In this standard form, RSLVQ can be used to classify Euclidean vectors only. Often, data are presented in more general form, representing pairwise similarities or dissimilarities of the data. Depending on whether the underlying similarity corresponds to an Euclidean feature space, an implicit underlying vector space is present in the case of kernel variants of prototype based techniques (see e.g. [9,29,42,41,47,57]), or a more general Krein space is present in relational variants (see e.g. [24,38,25]). Here we consider a recent kernelized version of RSLVQ model [50,29,25]. We assume a fixed kernel k corresponding to a feature map Φ. We set