Neurocomputing 169 (2015) 334–342

Contents lists available at ScienceDirect

Neurocomputing

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

Efficient rejection strategies for prototype-based classification

L. Fischer [a,b,*], B. Hammer [b], H. Wersing [a]

[a] HONDA Research Institute Europe GmbH, Carl-Legien-Str. 30, 63065 Offenbach, Germany
[b] Bielefeld University, Universitätsstr. 25, 33615 Bielefeld, Germany

ARTICLE INFO

Article history:
Received 26 June 2014
Received in revised form
26 September 2014
Accepted 18 October 2014
Available online 4 April 2015

Keywords:
Prototype-based
Classification
Global
Rejection

ABSTRACT

Due to intuitive training algorithms and model representation, prototype-based models are popular in settings where on-line learning and model interpretability play a major role. In such cases, a crucial property of a classifier is not only which class to predict, but also if a reliable decision is possible in the first place, or whether it is better to reject a decision. While strong theoretical results for optimum reject options in the case of known probability distributions or estimations thereof are available, there do not exist well-accepted reject strategies for deterministic prototype-based classifiers. In this contribution, we present simple and efficient reject options for prototype-based classification, and we evaluate their performance on artificial and benchmark data sets using the example of learning vector quantization. We demonstrate that the proposed reject options improve the accuracy in most cases, and their performance is comparable to an optimal reject option of the Bayes classifier in cases where the latter is available. Further, we show that the results are comparable to a well established reject option for support vector machines in cases where learning vector quantization classifiers are suitable for the given classification task, even providing better results in some cases.

1. Introduction

The digitalisation of many domains has turned automated classification algorithms into a standard tool in diverse application areas such as fraud detection, image recognition, and handwritten digit classification. Dramatically improved sensor technology and the increasing availability of high quality digital information carries the promise of radically new possibilities offered by machine learning technology in high impact domains such as personalised medicine [1]. In biomedical applications or safety-related fields, however, a wrong classification can severely affect the applicability of a classifier. The reliability of a classification constitutes a critical property of any method used in such domains [2,3]. In these fields, the reliability of classification results is as important as the accuracy of a classifier. It is often better to refuse the classification of a given data point rather than to predict a class with uncertain assignment [4]. In case of doubt, data can then be analysed by a human expert or it can be marked for further tests instead of a direct, uncertain classification.

Due to this demand, there exists an extensive literature of how to extend classification rules by reject options in an optimum way. The classical work of Chow [5] formalises the underlying learning scenario in terms of a loss function where the costs of a reject can be lower than the costs of a misclassification depending on the actual circumstances. In such cases, an optimum reject option can be derived with respect to these costs, provided class probabilities are known. Since the latter is usually not the case, the approach [6] studies the setting of plugin-rules for an estimation of the class probabilities. Consistent rules can be derived provided the probability estimation is of sufficient quality and no density mass accumulates in regions of the reject boundary. While providing a very elegant theoretical framework, the results are not fully satisfactory for a wide range of applications: first, the technology requires an estimation of the underlying class probabilities, which is often difficult in practice. For this reason, many approaches center around possibilities to reliably estimate class probabilities from given classifiers such as support vector machines (SVM), see e.g. the approaches [7,8] for technologies to approximately turn two-class or multiple-class SVMs, respectively, into fully probabilistic models. These methods, however, assign additional computational burden to the classifier and do not always allow reliable results. Second, the resulting loss function is no longer convex and hence its optimisation can become problematic. See e.g. the approaches [9,10] to approximate the setting by convex loss functions.

Due to these problems, there has been a strong interest how to devise reject strategies which can directly be used for a given (deterministic) classifier. As discussed in the paper [11], there are two main reasons for an uncertain classification: (i) ambiguous regions, e.g. points lie near class borders or (ii) outliers which are

* Corresponding author.
E-mail addresses: lfischer@cor-lab.uni-bielefeld.de (L. Fischer), bhammer@techfak.uni-bielefeld.de (B. Hammer), heiko.wersing@honda-ri.de (H. Wersing).

http://dx.doi.org/10.1016/j.neucom.2014.10.092
0925-2312/

caused by noise in the data or which are examples of a new type that is not yet represented by the actual model. Based on such considerations, quite a few heuristic reject strategies which capture these causes have been proposed [11–16].

Prototype-based classification constitutes a powerful machine learning scheme that has the advantages of an intuitive model understanding and sparse representation [17], leading to very interesting results e.g. in the biomedical domain [18]. One of the most popular examples for a supervised prototype-based model is offered by learning vector quantisation (LVQ) [19] for multi-class classification tasks. Due to the representation of models in terms of prototypes, this approach is particularly suited for on-line scenarios [20] or lifelong learning [21]. While classical LVQ models have been introduced on heuristic grounds, modern variants are based on cost-function models like generalized LVQ (GLVQ) [22], or robust soft LVQ (RSLVQ) [23]. This enables a principled treatment to guarantee the generalization performance and learning convergence of the resulting classifier [24,25]. Interestingly, prototype-based models provide a particularly efficient framework to integrate the powerful concept of metric learning such as presented in the overview [26]. Prototype models offer efficient metric parametrisation strategies by their decomposition of the data space into homogeneous receptive fields, see [25,27], for example. In this contribution, we will focus on different LVQ schemes, and we will investigate different efficient reject strategies which can be directly combined with classical, powerful LVQ classifiers.

While probabilistic classification models like Gaussian mixture models or Bayes classifiers directly provide a reject option based on their class probabilities, deterministic models such as prototype-based approaches often do not. Only few methods in the literature address prototype-based reject options without estimating probabilities [14,28,13] thereby lacking a comparison to other well established reject options. Common approaches for rejection usually rely on an estimation of class probabilities on top of a classifier to enable an optimum rejection following the approaches [5,6], see e.g. [11,29,30,7,8].

In this contribution we will propose several simple, efficient prototype-based reject options: we will consider reject options based on the distance of the point to the classification boundary, based on the indication of the point being an outlier, a combination of both, as well as a simple direct measurement inspired by the GLVQ cost function, which we will dub 'relative similarity'. In addition, we will consider the behaviour of the probabilistic model RSLVQ together with an optimum reject as specified by probabilistic plugin-rules. We will compare their performance to the optimal reject option of the Bayes classifier in a case where the latter is available. Further, we will compare their performance to a well established reject option of the support vector machine (SVM) [7,8]. We will demonstrate that the proposed reject options can have the same performance and even provide better results in some cases. In particular, the relative similarity seems an excellent compromise between a reliable reject measurement and its efficient computation.

2. Prototype-based classification

We are interested in classification scenarios in $\mathbb{R}^n$ with Z classes, enumerated as $\{1,\ldots,Z\}$. Prototype-based classifiers are defined as follows: a set W of prototypes $(\mathbf{w}_j, c(\mathbf{w}_j)) \in \mathbb{R}^n \times \{1,\ldots,Z\}$, $j \in \{1,\ldots,w\}$ is specified which should represent the data and its underlying classes in a proper way. Every prototype $\mathbf{w}$ is equipped with a class label $c(\mathbf{w})$. Then, given a new data point, the winner takes all scheme (WTA) that is used for classification

$$c(\mathbf{x}) = c(\mathbf{w}_l) \quad \text{with } l = \arg\min_{\mathbf{w}_j \in W} d(\mathbf{w}_j, \mathbf{x}) \tag{1}$$

where d is a distance measure, often the standard Euclidean distance. Hence the closest prototype $\mathbf{w}_l$, the winner, determines the class label of a new data point $\mathbf{x}$; it is also called the best matching unit (BMU). Training aims at an optimum determination of prototype locations given a set X of training data $(\mathbf{x}_i, y_i) \in \mathbb{R}^n \times \{1,\ldots,Z\}$.

Note that prototype-based models are very similar to k-nearest neighbour [31] (k-NN) classifiers due to their strong dependency on similarity calculations. A k-NN classifier simply stores all training points as 'prototypes' and predicts a label according to the closest ($k=1$) or the k closest units. In contrast, prototype-based training models aim at a sparser representation of data by a predefined number of prototypes. Training techniques can be divided into methods which are based on heuristics or alternatives which are derived from an explicit cost function. Original LVQ as proposed by Kohonen relies on the heuristic Hebbian learning paradigm [19], for example, with surprisingly good results in typical model situations, see [32].

Here, we will focus on extensions of LVQ which are derived from explicit cost functions such as generalized LVQ (GLVQ) [22] and robust soft LVQ (RSLVQ) [23]. These techniques have the advantage that convergence guarantees directly follow from their derivation. Further, an extension to more complex scenarios such as powerful metric adaptation is directly possible based on the formal objective function [25], the generalized matrix LVQ (GMLVQ). In addition to the local version of the GMLVQ, the LGMLVQ [25] is used in one experiment. This algorithm uses one local metric per prototype.

2.1. GLVQ and GMLVQ and its local version

Sato and Yamada [22] generalize the LVQ rule based on the formalisation as cost minimisation with the cost function

$$E = \sum_i \Phi\left(\frac{d^+(\mathbf{x}_i) - d^-(\mathbf{x}_i)}{d^+(\mathbf{x}_i) + d^-(\mathbf{x}_i)}\right). \tag{2}$$

The resulting model is dubbed generalised LVQ (GLVQ). The function Φ has to be monotonic increasing, e.g. the logistic function. $d^\pm$ is the distance to the closest prototype $\mathbf{w}^\pm$ of the correct/incorrect class for a data point $\mathbf{x}_i$. GLVQ optimizes the location of prototypes by means of a stochastic gradient descent based on this cost function (2), see e.g. [33] for a proof of its validity at the boundaries of receptive fields. A generalization of GLVQ towards an algorithm with metric adaptation has been published under the acronym GMLVQ [25], which is a short hand notation for generalized matrix LVQ. This takes into account a positive semi-definite matrix Λ in the general quadratic form which replaces the metric d of the GLVQ, i.e. $d(\mathbf{w}_j, \mathbf{x}) = (\mathbf{x}-\mathbf{w}_j)^T \Lambda (\mathbf{x}-\mathbf{w}_j)$. The local version, the LGMLVQ, uses a single metric $d_j(\mathbf{w}_j, \mathbf{x}) = (\mathbf{x}-\mathbf{w}_j)^T \Lambda_j (\mathbf{x}-\mathbf{w}_j)$ for each prototype $\mathbf{w}_j$.

The cost function (2) strongly correlates to the classification error since a data point is classified correctly iff the nominator of the cost function is smaller than zero. Further, the nominator can be linked to the hypothesis margin of the classifier which influences its generalization ability [25]. Note that the value of the fraction ranges in the interval $(-1,1)$ with -1 indicating a certain classification because d^+ is much smaller than d^-. Due to its excellent performance in practice [34], we will consider a reject option related to these costs in the following.

2.2. RSLVQ

Robust soft learning vector quantization [23] is based on the assumption of a Gaussian mixture model with labelled types. Training