Efficient Rejection Strategies for Prototype-based Classification

L. Fischer[a,1,*], B. Hammer[b,2], H. Wersing[a]

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

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.

Keywords: prototype-based, classification, global, rejection

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, handwritten digit classification, etc. 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

*Corresponding author

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

[1]LF acknowledges funding by the CoR-Lab Research Institute for Cognition and Robotics and acknowledges the financial support from Honda Research Institute Europe.

[2]BH gratefully acknowledges funding by the CITEC center of excellence.

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 article [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 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 [12, 13, 14, 15, 16, 11].

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 in which 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 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) is used for classification:

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

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 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 & 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 is derived thereof as an optimisation of the data log likelihood:

$$E = \sum_i \log p(y_i|\mathbf{x}_i, W) = \sum_i \log \frac{p(\mathbf{x}_i, y_i|W)}{p(\mathbf{x}_i|W)}$$

$p(\mathbf{x}_i|W) = \sum_j p(\mathbf{w}_j)p(\mathbf{x}_i|\mathbf{w}_j)$ is a mixture of Gaussians with uniform prior probability $p(\mathbf{w}_j)$ and Gaussian probability $p(\mathbf{x}_i|\mathbf{w}_j)$ centred in $\mathbf{w}_j$ which is isotropic with fixed variance and equal for all prototypes or, more generally, a general (possibly adaptive) covariance matrix. The probability $p(\mathbf{x}_i, y_i|W) = \sum_j \delta_{c(\mathbf{x}_i)}^{c(\mathbf{w}_j)} p(\mathbf{w}_j)p(\mathbf{x}_i|\mathbf{w}_j)$ (δ_i^j is the Kronecker delta) restricts to mixture components with correct labelling. Relying on a probability model, RSLVQ provides an explicit certainty value $p(y|\mathbf{x}, W)$ for every pair $\mathbf{x}$ and y, paying the price of a higher computational complexity for training.

3. Global Reject Option

A reject option relaxes the constraint on a classifier to provide a class label for every input. We will consider reject options which are based on certainty measures. Given a certainty measure

$$r : \mathbb{R}^n \to \mathbb{R}, \mathbf{x} \mapsto r(\mathbf{x}) \tag{3}$$

for a data point $\mathbf{x}$ and a threshold $\theta \in \mathbb{R}$, a simple reject option is to reject $\mathbf{x}$ iff

$$r(\mathbf{x}) < \theta \ . \tag{4}$$

If a data point is rejected no classification will take place and the decision is postponed. We denote such a reject option as 'global reject option' because the threshold θ is chosen uniformly for the whole input space. Extensions to local threshold strategies are possible, but out of the scope of this article [35].

As mentioned in [11], uncertainty can have two different reasons: data points being outliers, or data points being located in ambiguous regions. As we will discuss, certainty measures can take these two causes into account to varying degrees. Further, certainty measures differ according to their scaling, allowing a uniform threshold θ iff $r(\mathbf{x})$ is normalized, and they differ according to their computational complexity and on-line computability, i. e. efficiency. We will focus on different possible choices for natural certainty measures in the following section.

3.1. Certainty Measures

Common choices for a rejection measure r are based on estimated probabilities or on heuristics. Measures based on probabilities often either require a probabilistic classification model [5, 36] or a probabilistic model on top of the trained classifier to estimate the probabilities [11, 29]. Both approaches are computationally expensive. Heuristic measures can be based on distances [14, 7, 8, 37] or on the neighbouring class labels [28]. In the following we introduce two probabilistic measures based on a probabilistic classification model and several heuristic measures based on distances.

Note we use d as symbol for all metrics. The definition of d for the used algorithms can be found in the list below:

- GLVQ: $d(\mathbf{w}_j, \mathbf{x}) = (\mathbf{x} - \mathbf{w}_j)^T(\mathbf{x} - \mathbf{w}_j)$
- GMLVQ: $d(\mathbf{w}_j, \mathbf{x}) = (\mathbf{x} - \mathbf{w}_j)^T\Lambda(\mathbf{x} - \mathbf{w}_j)$
- LGMLVQ: $d_j(\mathbf{w}_j, \mathbf{x}) = (\mathbf{x} - \mathbf{w}_j)^T\Lambda_j(\mathbf{x} - \mathbf{w}_j)$ for each prototype $\mathbf{w}_j$

Bayes. The Bayes classifier provides class probabilities for each class provided the data distribution is known. The reject option corresponding to the certainty measure

$$r_{\text{Bayes}}(\mathbf{x}) = \max_{1 \leq j \leq Z} p(j|\mathbf{x}) \tag{5}$$

is optimal in the sense of an error-reject trade-off [5]. We will use it as ground truth for an artificial data set with known underlying distribution. Figure 1 shows the contour lines of Bayes (5) for an artificial two class problem with known class densities. In general, the class probabilities are unknown, such that this optimum Bayes reject option can serve as Gold standard for artificially designed settings with ground truth only.

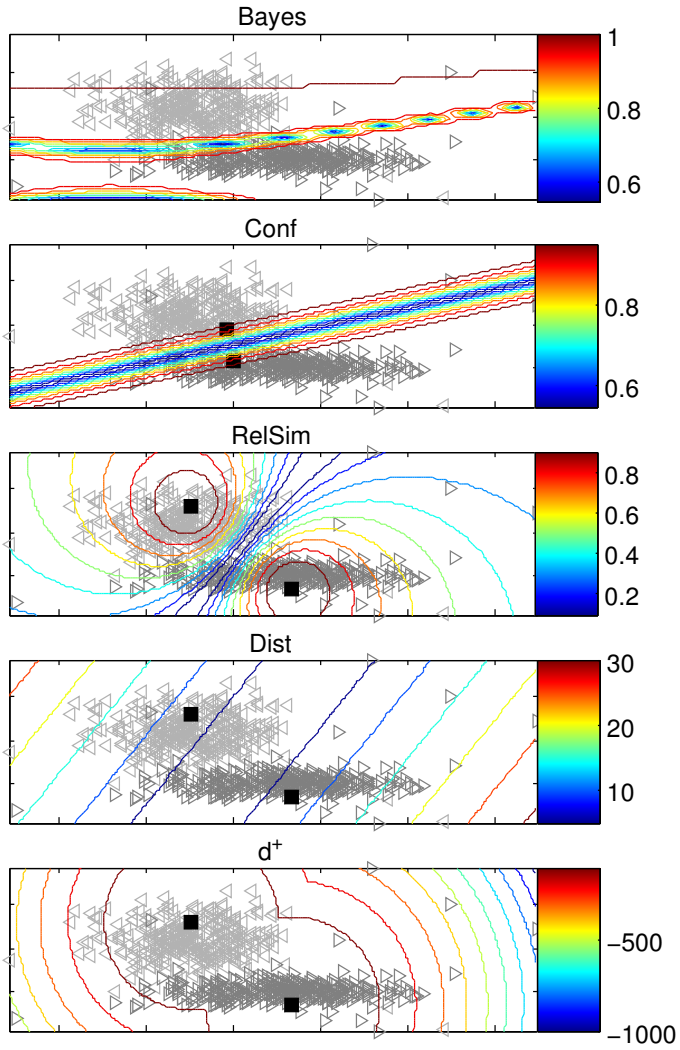

Figure 1: [37] Contour lines of the measures for artificial 2D data. The heading of a plot indicates which measure is used. This two class problem consists of data of two Gaussians (symbols: $\triangleleft, \triangleright$). Black squares are GLVQ/RSLVQ prototypes.

Conf. Classifiers based on probabilistic models such as RSLVQ provide a certainty value of the classification:

$$r_{\text{Conf}}(\mathbf{x}) = \max_{1 \le j \le Z} \hat{p}(j|\mathbf{x}) \tag{6}$$

with the estimated probability $\hat{p}(\cdot)$ as obtained during training. We can use this certainty measure for every setting where a probability value is obtained while training. Hence these settings fall under the framework of plugin-rules as investigated in [6]. One problem is caused by the fact that it is often unclear how good the empirical estimation $\hat{p}(\cdot)$ resembles the underlying probability. This is particularly problematic in supervised settings where the objective is often modelled as a good classification accuracy of the model rather than an exact estimation of the probability values.

This measure is normalized and, depending on the probability model, it takes into account ambiguous regions (Fig. 1). Note that it does not necessarily reject outliers, depending on the quality of the empirical estimation. A severe drawback is that this measure can only be used for probabilistic models such as RSLVQ, and its accuracy relies on the (often problematic) quality of the density estimation. In our case, we are faced with the higher computational complexity of RSLVQ as compared to its deterministic counterparts GLVQ or GMLVQ. In the following, Conf serves as baseline for an evaluation whether simple geometric measures can reach (or even outperform) the quality of an explicit probabilistic modelling.

RelSim. The relative similarity is a GLVQ cost function (2) related measure which is a slight modification of the $\mu(\mathbf{x})$ rejection, first mentioned in [22] where the argument of the Φ function in (2) is denoted as $\mu(\mathbf{x})$. RelSim [37] takes the distance of the closest prototype (BMU) d^+ and the distance of closest prototype of a different class d^- for a new unlabelled data point into account. This means the prototype which belongs to d^+ defines the class label of this unlabelled data point if it is not rejected. The measure calculates values according to:

$$r_{\text{RelSim}}(\mathbf{x}) = \frac{d^- - d^+}{d^- + d^+} \ . \tag{7}$$

The relation $r_{\text{RelSim}}(\mathbf{x}) = -\mu(\mathbf{x})$ is valid for the function $\mu(\mathbf{x})$ in [22] in the case of a GLVQ classifier. The measure ranges in the interval $(0, 1)$ where values near 1 indicate a certain classification and values near 0 are an indicator for uncertain class labels.

The values of d^+ and d^- are already calculated by the used algorithm and therefore no additional computational costs are caused. Furthermore RelSim (7) depends only on the stored prototypes W and the new unlabelled data point $\mathbf{x}$. Therefore no additional storage is needed, and the technique is well suited for on-line computation. Figure 1 shows the contour lines of RelSim (7)

for an artificial two class problem with trained prototypes by the GLVQ. The values near the class border are low. This means the measure correctly detects ambiguous rejection. In addition, as can be seen from the circular contour lines (in the Euclidean metric), a rejection of outliers is included. Therefore, this measure seems a good compromise between an efficient measure and a richness of its representation.

Dist. As certainty measure, we consider the uniqueness of the classification as measured by the distance of a point to the closest decision boundary of the classifier. The distance of a point $\mathbf{x}$ to the hyperplane separating the receptive fields of $\mathbf{w}^+$ and $\mathbf{w}^-$ is given by

$$r_{\mathrm{Dist}}(\mathbf{x}) = \frac{|d^+ - d^-|}{2\|\mathbf{w}^+ - \mathbf{w}^-\|^2} \ . \tag{8}$$

provided every class is modelled by only one prototype. Figure 1 shows the contour lines of Dist (8) for an artificial two class problem with trained prototypes by the GLVQ.

Note the distance between the prototypes $\|\mathbf{w}^+ - \mathbf{w}^-\|^2$ has to be calculated as the distances $d^{\pm}$.

For settings where more than one prototype per class is used, the underlying topology has to be estimated using e. g. Hebbian learning [38]. Then, (8) can be used for the pairs of prototypes that define the corresponding class border. An experimental evaluation has shown that the approximation of Dist based on $d^+, d^-, \mathbf{w}^+, \mathbf{w}^-$ (even if $\mathbf{w}^+$ and $\mathbf{w}^-$ do not define a class border) provides good results with less effort compared to the correct calculation. Therefore, we always use this approximation, avoiding additional computational burden.

Dist can be computed efficiently, but its range is not normalized. It depends on the stored prototypes W, the distance calculation, i. e. d^+, d^- and the new data point $\mathbf{x}$. This means no additional storage is needed and the needed values for the Dist calculation can be used directly without much additional computational effort.

Note that Dist and the reject option of the SVM [7] are closely related in case of a binary setting and one prototype per class in a prototype-based classification model since both models determine one separating hyperplane in this setting. Dist takes the *pure* distance of a data point to the hyperplane as rejection measure. The rejection of the SVM scales the distances to the

hyperplane with an adapted sigmoid function. This can cause a shifting of the classification border especially if the classes are unbalanced.

$\mathbf{d}^+$. Outliers can be identified by their distance to the closest prototype d^+ (Fig. 1). We use this information for an outlier-based certainty measure as basis for a reject option:

$$r_{d^+}(\mathbf{x}) = -d^+(\mathbf{x}) \ . \tag{9}$$

d^+ uses the stored prototypes W and the distance calculation. Therefore this measure is efficient. Note, that the measure d^+ is not normalized.

Comb. This measure combines the previous two reject options

$$r_{\text{Comb}}(\mathbf{x}) = (r_{\text{Dist}}(\mathbf{x}), r_{d^+}(\mathbf{x})) \tag{10}$$

leading to a reject strategy based on a threshold vector $\theta = (\theta_1, \theta_2)$: $\mathbf{x}$ is rejected iff

$$r_{\text{Dist}}(\mathbf{x}) < \theta_1 \text{ or } r_{d^+}(\mathbf{x}) < \theta_2 \ . \tag{11}$$

The measure takes into account ambiguity and outliers, but it requires two threshold parameters. For evaluation, we refer to the best combination of both thresholds determined via exhaustive search. This combination is no longer efficient since it requires a loop over the regime of threshold vectors, but it can excellently serve as a baseline for comparison.

3.2. Characteristics of the Measures

These measures display different principled properties, which we discuss in the following. Table 1 shows an overview of a few relevant formal properties. The top row lists the different measures whereby SVM refers to a rejection based on an SVM and standard rejection techniques as implemented in LibSVM [39]. The first column states the properties for comparison. The first row specifies requirements of the techniques. The measures RelSim, d^+, Dist and Comb do only rely on a set of prototypes W, whereas the measures Conf and Bayes need class probabilities or its estimations, respectively. For the reject option as provided by SVM the training set needs to be stored.

The next property specifies the co-domains of the measures. This is particularly interesting since it indicates whether a natural predefined choice of the threshold θ is possible (for a normalised co-domain) or not (if the co-domain is unlimited). Still, even for the same co-domains such as for

RelSim and Conf, it is unclear whether their interpretation coincides and hence similar thresholds have the same meaning.

The next property, comparable scaling, makes this more precise. It refers to the question whether the provided value displays the same range independent of the location of data points in the data space, or whether the scaling can severely change with different locations in the data space and different receptive fields. In the latter case, it is likely that global threshold strategies do not provide satisfactory results, but local threshold values have to be used. Provided measurements refer to probabilities such as for Conf and Bayes, a uniform scaling is present. For all other measures (RelSim, d^+, Dist, Comb) a uniform scaling is not guaranteed since relative distances can vary severely across the data space.

The next two lines refer to the type of rejects offered by the measures: Do they detect outliers and/or ambiguous regions, respectively?

A very interesting property is summarised in the final row: Can the measures be used in on-line settings, i. e. is it computable based on a finite number of parameters of the model? The proposed measures (RelSim, d^+, Dist, Comb, Conf) can be used in on-line settings because they only depend on the prototypes W. This means if the model is adapted in an on-line way the rejection measures takes these changes immediately into account because they are based on distances from data to prototypes only. For Bayes and SVM rejection this is not possible in an on-line way although there exist on-line training schemes of these algorithms: The rejection techniques require the whole training data for its computation, hence an update of the reject measure cannot be done in on-line settings with a finite amount of memory. Therefore a previous calculated rejection measure fits no longer to the permanent updated model of a Bayes or SVM classifier trained in an on-line way.

4. Experiments

After these theoretical considerations, we evaluate the results of the reject strategies for different data sets. In all cases, we use a 10-fold repeated cross-validation with ten repeats for RSLVQ, GLVQ, and (L)GMLVQ with one prototype per class. We compare our results with the results of the standard rejection measure of SVM [7, 8] implemented in the LibSVM toolbox [39].

	RelSim	d^+	Dist	Comb	Conf	Bayes	SVM
Requirements	W	W	W	W	$\hat{p}(j\|\mathbf{x})$	$p(j\|\mathbf{x})$	data
Co-domain	$(0,1)$	$(-\infty,0)$	$(0,\infty)$	$(-\infty,\infty)$	$(0,1)$	$(0,1)$	$(-1,1)$
Comparable Scaling	no	no	no	no	yes	yes	no
Outliers	yes	yes	no	yes	no	no	no
Ambiguity	yes	no	yes	yes	yes	yes	yes
On-line	yes	yes	yes	yes	yes	no	no

Table 1: Properties of the analysed measures

4.1. Evaluation Scheme

As already mentioned, we use one global threshold θ for rejection. We evaluate the performance by means of the resulting accuracy reject curves (ARC) [40]. The latter is defined as follows: For a given value θ the data decompose into two sets $X = X_\theta \cup R$. The set R contains the rejected data points; X_θ contains the remaining data points which are classified. For an increasing threshold θ starting from no reject ($\theta = \min_i r(\mathbf{x}_i)$, original model) to full reject ($\theta = \max_i r(\mathbf{x}_i)$, no data point is classified any more) the cardinality of R increases whereas the cardinality of X_θ decreases. In the ARC, the relative size of $|X_\theta|/|X|$ versus the accuracy on X_θ is reported by means of a variation of the threshold θ in the interval $[\min_i r(\mathbf{x}_i), \max_i r(\mathbf{x}_i)]$.

In Fig. 2 to 4, we display the ARC averaged over 100 runs per data set and rejection measure. For numerical reasons, we do not display the point for $|X_\theta| = 0$ where no point is classified. Note that the single curves can end in different points with maximum threshold value. To ensure a reliable display, we only report those points where at least 80% of the repeats deliver a value.

4.2. Artificial and Benchmark Data

We report experiments on one artificial data set with known ground truth for the Bayes optimal rejection and four benchmarks.

Gaussian clusters. This data set contains two artificially generated overlapping 2D Gaussian clusters. These are overlaid with uniform noise. (parameters: $\mu_x = (-4, 4.5),\ \mu_y = (4, 0.5),\ \sigma_x = (5.2, 7.1),\ \sigma_y = (2.5, 2.1)$)

Tecator data. The tecator data set consists of 215 spectra with 100 spectral bands ranging from 850 nm to 1,050 nm [41]. The task is to predict the fat content of the probes, which is turned into a two class classification problem to predict a high/low fat content by binning into two classes of equal size.

Image Segmentation. The image segmentation data set consists of 2,310 data points representing small patches from outdoor images with 7 different classes with equal distribution such as brick-face, sky, ... [42]. Each data point consists of 19 real-valued image descriptors.

Haberman. The Haberman survival data set contains 306 instances from two classes indicating the survival for more than 5 years after breast cancer surgery [42]. Data are represented by three attributes related to the age, the year, and the number of positive axillary nodes detected.

COIL-20. The Columbia Object Image Database Library (COIL-20) consists of gray scaled images of twenty objects [43]. The objects are rotated in 5° steps, so that there are 72 images per object. The data set contains 1,440 data points which are 16,384 dimensional. We use PCA [44] to reduce the dimensionality to 30. The task is to classify each single object.

4.3. Results

We report the effect of the different reject strategies for the different models RSLVQ, GLVQ, and (L)GMLVQ where applicable: We can combine RSLVQ with Conf since the former provides explicit probabilities. All techniques can be combined with d^+, Dist, and Comb, since these measures depend on the provided prototypes only. For GLVQ and GMLVQ, the measure RelSim is already computed while training. For the artificial data set, the ground truth in the form of the data distribution is available, such that we can compare to the optimum Bayes decision.

4.3.1. Experiments on Artificial Data

Figure 2 displays the results for the Gaussian clusters data set. Note that errors mostly stem from ambiguity in the overlapping region, such that a reject option due to outliers is less efficient for this setting. We can observe that the probabilistic model RSLVQ together with a confidence estimation well resembles the optimum reject strategy of a Bayes classifier. GLVQ does not reach the performance of the Bayes classifier because it relies on the standard Euclidean distance. Hence it cannot account for the different

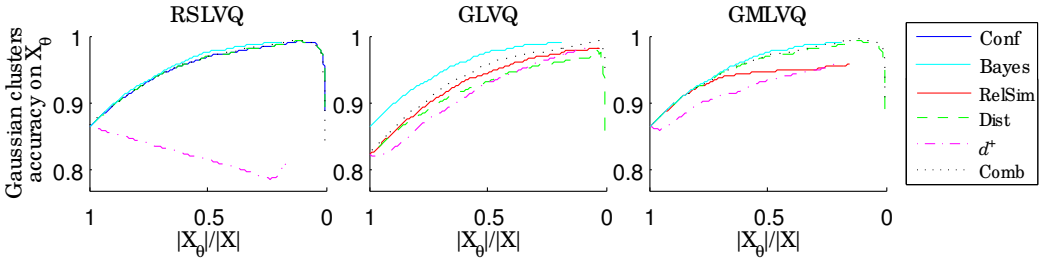

Figure 2: [37] Accuracy reject curves for different reject options when applying RSLVQ, GLVQ and GMLVQ models trained on Gaussian clusters. We display the relative size of the remaining data points X_θ vs. the accuracy of the classifier on this set.

standard variations in the two axes of the two Gaussians. Matrix adaptation improves this behaviour, and RelSim as well as Dist and Comb reach the performance of the optimum Bayes reject in the (important) regime of up to 25 % rejected data points. For more rejects, the accuracy of RelSim drops due to its reject of outliers. Overall, this setting shows that the reject options as proposed in this contribution which rely on the distance to the decision boundary or the confidence value are well suited for a close to optimum reject in the interesting regime, provided the underlying prototype-based model is sufficiently flexible to capture the nature of the data.

4.3.2. Benchmark data

Figure 3 displays the average ARCs of the rejection measures for the benchmark data sets. Mere outlier detection d^+ does not work well on average, which can be attributed to the fact that most errors can be accounted for by ambiguities rather than unknown types. This principle might become more important in on-line scenarios where the underlying distribution is subject to trend.

Dist and Conf show similar results for the RSLVQ models, with an exception being the Coil data where dist is even superior to Conf. This finding indicates that the more efficient measure Dist can be sufficient for a reliable reject option, making the (more complex) estimation of probability values superfluous. For the Coil data, Dist is even superior, which indicates that the plugin estimation of the probability by the RSLVQ values is not a good estimator for these data.

GLVQ displays results which are mostly inferior to RSLVQ, while GMLVQ can reach the same or even better accuracy. This can be attributed to the fact

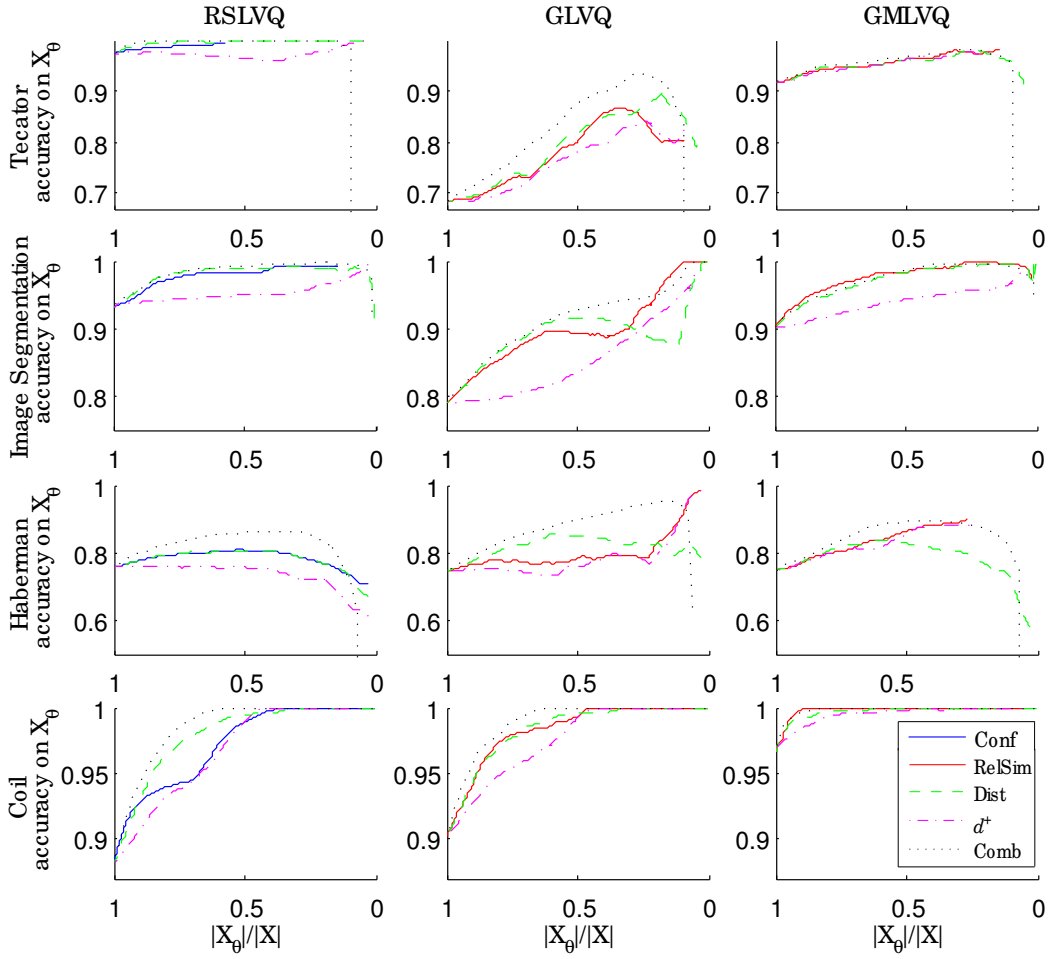

Figure 3: [37] Accuracy reject curves for several prototype-based classifiers trained on benchmark data sets.

that GLVQ does not process sufficient flexibility of data representation since it is restricted to isotropic isobars with uniform scaling over all prototypes, while RSLVQ can change the bandwidth as a meta parameter, and GMLVQ can even adapt the local metric according to the data. For GLVQ, Dist and RelSim mostly provide comparable results in the relevant regime of up to 25 % rejected data points. Interestingly, taking into account an optimum combination with outlier detection can improve this performance, albeit outlier detection alone (d^+) is not very performant. This combination, however, does not offer an

efficient technology since it requires an additional loop over possible threshold vectors.

For GMLVQ, RelSim and Dist both provide excellent results which are comparable to or even better than a fully probabilistic modelling as offered by RSLVQ and Conf. Hence it offers a good compromise between model accuracy and efficiency of the reject option. In addition, it has the benefit that an adaptation to on-line scenarios is easily possible, the measures depending on the position of the prototypes only.

4.3.3. Comparison to SVM Rejection

We compare the results of the rejection measures Conf, Comb, Dist, and RelSim with a state of the art reject option on top of an SVM model. This enables us to compare the efficiency of the proposed reject options to alternative models which are not based on prototypes. For the SVM reject option for binary classes, a technology which rescales the distance to the boundary has been proposed in [7]: A sigmoid function is fitted against the binned distances of the training data points to the separating hyperplane such that probabilities which are estimated from the data are matched as closely as possible. This approach can be extended to multi-class settings by means of a pairwise coupling as proposed in [8].

Figure 4 displays the results of RSLVQ and GMLVQ as taken from Fig. 3 and results for the LGMLVQ in comparison to SVM for the relevant reject options and the data sets Tecator, Image Segmentation, Habermann, and Coil. For GMLVQ and RSLVQ all data but Habermann, SVM is capable of obtaining a better accuracy at the price of a more complex model: The average number of support vectors per model is 14.96 for Tecator, 265.81 for Image Segmentation, and 145.51 for Haberman. For the sake of completeness, we show the results for Coil, although the SVM reaches an accuracy close to 100%, hence the reject option cannot be evaluated in a meaningful way. The difference between the accuracy of the SVM and the LGMLVQ are very small because the latter is powerful due to its trained local metrics.

Interestingly, in all cases reject options decrease the difference of the accuracy provided by SVM and the accuracy of RSLVQ or GMLVQ. Hence the proposed reject options seem to provide a very suitable strategy as concerns the acquired performance. For the Haberman data set, the results are even superior as compared to SVM. Hence prototype-based methods such as (L)GMLVQ or RSLVQ together with efficient reject options such as Conf or RelSim offer a good compromise of a sparse classification model enhanced

with the possibility of reject, and a good classification accuracy.

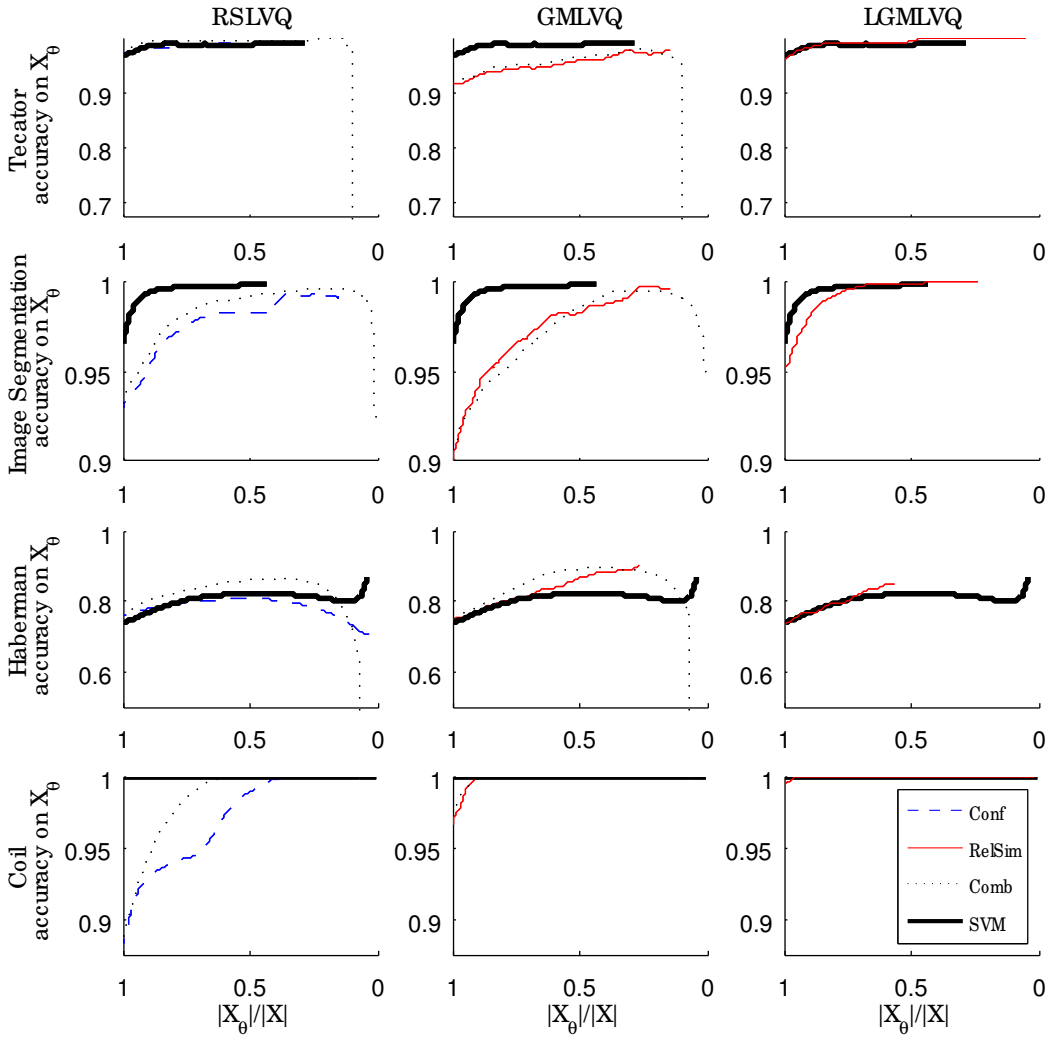

Figure 4: Comparison to SVM rejection

5. Conclusion

We have proposed and systematically compared several reject options for prototype-based techniques using the example of learning vector quantisation. In particular, we have proposed efficient geometric reject measures

for prototype-based approaches which have the potential of direct on-line applicability. We have compared these direct measures with statistical reject strategies which are based on a full (more demanding) probabilistic modelling, and with state of the art rejection for SVM on benchmark data sets. Interestingly these settings constitute typical representatives of popular classification paradigms: (L)GMLVQ as a popular LVQ scheme based on a cost function and motivated by large margin optimisation, incorporating the very powerful framework of metric learning in its model; RSLVQ as statistically motivated discriminative model; and, in comparison, SVM as discriminative large margin model which, unlike sparse prototype-based representations, relies on a representation of class boundaries in terms of support vectors.

We have demonstrated that efficient geometrically motivated measures (RelSim, Dist) can be used as efficient reject options, providing results which are comparable to optimum Bayes reject strategies where available, but releasing the burden of explicit statistical modelling. Interestingly, geometric measures reach the accuracy of fully probabilistic models used as plugin-rules. Further, the reject options approach the performance of SVM techniques equipped with state of the art reject options. In such settings, however, SVM usually displays a better overall accuracy for the full model due to its ability to use a flexible description of the class boundaries in terms of support vectors rather than a sparse prototype-based representation only. We would like to stress the fact that the proposed reject measures are not restricted to LVQ classifiers but they have a broader scope: On the one hand, the training technique is not relevant for the scenario, rather any prototype-based classifier can be enhanced accordingly, such as unsupervised techniques equipped with posterior labels. On the other hand, some of the concepts transfer to alternative models such as the distance to the class boundary. E. g. any model where one can define the closest distance to the class boundary of a data point one can apply a reject option based on this measure.

These findings open the way towards the design of efficient lifelong model adaptation for popular prototype-based classifiers such as (L)GMLVQ: The model complexity can easily be tailored on-line towards regions with a low certainty of the classification, e. g. introducing novel prototypes which are capable of representing novel aspects of the data.

References

[1] J. C. Weiss, S. Natarajan, P. L. Peissig, C. A. McCarty, D. Page, Machine Learning for Personalized Medicine: Predicting Primary Myocardial Infarction from Electronic Health Records., AI Magazine 33 (4) (2012) 33–45.

[2] C. Rudin, K. L. Wagstaff, Machine learning for science and society, Machine Learning 95 (1) (2014) 1–9.

[3] A. Vellido, J. D. Martin-Guerrero, P. J. G. Lisboa, Making machine learning models interpretable, in: ESANN, 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2012, pp. 163–172.

[4] B. Hanczar, E. R. Dougherty, Classification with reject option in gene expression data., Bioinformatics 24 (17) (2008) 1889–1895.

[5] C. K. Chow, On Optimum Recognition Error and Reject Tradeoff, in: IEEE Transactions in Information Theory, Vol. 16(1), 1970, pp. 41–16.

[6] R. Herbei, M. H. Wegkamp, Classification with reject option, Canadian Journal of Statistics 34 (4) (2006) 709–721. doi:10.1002/cjs.5550340410.

[7] J. C. Platt, Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods, in: Advances in Large Margin Classifiers, MIT Press, 1999.

[8] T.-F. Wu, C.-J. Lin, R. C. Weng, Probability Estimates for Multi-class Classification by Pairwise Coupling, Journal of Machine Learning Research 5 (2004) 975–1005.

[9] P. L. Bartlett, M. H. Wegkamp, Classification with a reject option using a hinge loss, Journal of Machine Learning Research 9 (2008) 1823–1840.

[10] M. Yuan, M. Wegkamp, Classification Methods with Reject Option Based on Convex Risk Minimization, Journal of Machine Learning Research 11 (2010) 111–130.

[11] A. Vailaya, A. K. Jain, Reject Option for VQ-Based Bayesian Classification, in: International Conference on Pattern Recognition (ICPR), 2000, pp. 2048–2051.

[12] G. Fumera, F. Roli, G. Giacinto, Reject option with multiple thresholds, Pattern Recognition 33 (12) (2000) 2099–2101.

[13] C. De Stefano, C. Sansone, M. Vento, To Reject or Not to Reject: That is the Question-An Answer in Case of Neural Classifiers, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 30 (1) (2000) 84–94. doi:10.1109/5326.827457.

[14] J. Suutala, S. Pirttikangas, J. Riekki, J. Röning, Reject-Optional LVQ-Based Two-Level Classifier to Improve Reliability in Footstep Identification, in: A. Ferscha, F. Mattern (Eds.), Pervasive, Vol. 3001 of Lecture Notes in Computer Science, Springer, 2004, pp. 182–187.

[15] G. Fumera, F. Roli, Support Vector Machines with Embedded Reject Option, in: International Workshop on Pattern Recognition with Support Vector Machines (SVM2002), Niagara Falls, Springer, 2002, pp. 68–82.

[16] L. P. Cordella, C. de Stefano, C. Sansone, M. Vento, An Adaptive Reject Option for LVQ Classifiers, in: International Conference on Image Analysis and Processing (ICIAP), 1995, pp. 68–73.

[17] M. Biehl, B. Hammer, P. Schneider, T. Villmann, Metric Learning for Prototype-Based Classification, in: M. Bianchini, M. Maggini, F. Scarselli, L. C. Jain (Eds.), Innovations in Neural Information Paradigms and Applications, Vol. 247 of Studies in Computational Intelligence, Springer, 2009, pp. 183–199.

[18] W. Arlt, M. Biehl, A. Taylor, S. Hahner, R. Libe, B. Hughes, P. Schneider, D. Smith, H. Stiekema, N. Krone, E. Porfiri, G. Opocher, J. Bertherat, F. Mantero, B. Allolio, M. Terzolo, P. Nightingale, C. Shackleton, X. Bertagna, M. Fassnacht, P. Stewart, Urine steroid metabolomics as a biomarker tool for detecting malignancy in adrenal tumors, Journal of Clinical Endocrinology and Metabolism 96 (2011) 3775–3784.

[19] T. Kohonen, Self-Organization and Associative Memory, Springer Series in Information Sciences, Springer-Verlag, third edition, 1989.

[20] A. Denecke, H. Wersing, J. J. Steil, E. Körner, Online Figure-Ground Segmentation with Adaptive Metrics in Generalized LVQ, Neurocomputing 72 (7-9) (2009) 1470–1482.

[21] S. Kirstein, H. Wersing, H.-M. Gross, E. Körner, A Life-Long Learning Vector Quantization Approach for Interactive Learning of Multiple Categories, Neural Networks 28 (2012) 90–105.

[22] A. Sato, K. Yamada, Generalized Learning Vector Quantization, in: Advances in Neural Information Processing Systems, Vol. 7, 1995, pp. 423–429.

[23] S. Seo, K. Obermayer, Soft Learning Lector Quantization., Neural Computation 15 (7) (2003) 1589–1604. doi:10.1162/089976603321891819.

[24] M. Biehl, A. Ghosh, B. Hammer, Dynamics and generalization ability of LVQ algorithms, The Journal of Machine Learning Research 8 (2007) 323–360.

[25] P. Schneider, M. Biehl, B. Hammer, Adaptive Relevance Matrices in Learning Vector Quantization, Neural Computation 21 (12) (2009) 3532–3561.

[26] A. Bellet, A. Habrard, M. Sebban, A Survey on Metric Learning for Feature Vectors and Structured Data, Tech. rep. (Jun. 2013). arXiv:1306.6709.

[27] P. Schneider, M. Biehl, B. Hammer, Distance Learning in Discriminative Vector Quantization, Neural Computation 21 (10) (2009) 2942–2969.

[28] R. Hu, S. J. Delany, B. M. Namee, Sampling with Confidence: Using k-NN Confidence Measures in Active Learning, in: Proceedings of the UKDS Workshop at 8th International Conference on Case-based Reasoning, ICCBR'09, 2009, pp. 181–192.

[29] E. Ishidera, D. Nishiwaki, A. Sato, A confidence value estimation method for handwritten Kanji character recognition and its application to candidate reduction, International Journal on Document Analysis and Recognition 6 (4) (2004) 263–270.

[30] P. R. Devarakota, B. Mirbach, B. Ottersten, Confidence Estimation in Classification Decision: A Method for Detecting Unseen Patterns, in: International Conference on Advances in Pattern Recognition (ICAPR 2007), 2006.

[31] T. M. Cover, P. E. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1) (1967) 21–27.

[32] M. Biehl, A. Ghosh, B. Hammer, Dynamics and generalization ability of LVQ algorithms, Journal of Machine Learning Research 8 (2007) 323–360.

[33] B. Hammer, M. Strickert, T. Villmann, Supervised Neural Gas with General Similarity Measure, Neural Processing Letters 21 (1) (2005) 21–44.

[34] M. Biehl, K. Bunte, P. Schneider, Analysis of flow cytometry data by matrix relevance learning vector quantization, PLoS ONE 8 (3) (2013) e59401. doi:10.1371/journal.pone.0059401.

[35] L. Fischer, B. Hammer, H. Wersing, Local Rejection Strategies for Learning Vector Quantization, in: ICANN, 24th International Conference on Artificial Neural Networks, 2014, pp. 563–570.

[36] L. Fischer, D. Nebel, T. Villmann, B. Hammer, H. Wersing, Rejection Strategies for Learning Vector Quantization – a Comparison of Probabilistic and Deterministic Approaches, in: WSOM, 10th Workshop on Self-Organizing Maps, 2014 accepted.

[37] L. Fischer, B. Hammer, H. Wersing, Rejection Strategies for Learning Vector Quantization, ESANN, 22nd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, i6doc.com, 2014, pp. 41–46.

[38] T. Martinetz, K. Schulten, Topology representing networks, Neural Networks 7 (1994) 507–522.

[39] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (2011) 27:1–27:27.

[40] M. S. A. Nadeem, J.-D. Zucker, B. Hanczar, Accuracy-Rejection Curves (ARCs) for Comparing Classification Methods with a Reject Option, in: International Workshop on Machine Learning in Systems Biology (MLSB), 2010, pp. 65–81.

[41] H. H. Thodberg, Tecator data set, contained in StatLib Datasets Archive (1995).

[42] K. Bache, M. Lichman, UCI machine learning repository (2013).

[43] S. A. Nene, S. K. Nayar, H. Murase, Columbia Object Image Library (COIL-20), Technical Report CUCS-005-96.

[44] L. J. P. van der Maaten, Matlab Toolbox for Dimensionality Reduction (March 2013).