Distance Measures for Prototype Based Classification

Michael Biehl[1(✉)], Barbara Hammer[2], and Thomas Villmann[3]

[1] Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, P.O. Box 407, 9700 Groningen, The Netherlands
m.biehl@rug.nl
[2] CITEC Centre of Excellence, Bielefeld University, Universitätsstr. 21-33, 33594 Bielefeld, Germany
[3] Department of Mathematics, University of Applied Sciences Mittweida, Technikumplatz 17, 09648 Mittweida, Germany

Abstract. The basic concepts of distance based classification are introduced in terms of clear-cut example systems. The classical k-Nearest-Neigbhor (kNN) classifier serves as the starting point of the discussion. Learning Vector Quantization (LVQ) is introduced, which represents the reference data by a few prototypes. This requires a data driven training process; examples of heuristic and cost function based prescriptions are presented. While the most popular measure of dissimilarity in this context is the Euclidean distance, this choice is frequently made without justification. Alternative distances can yield better performance in practical problems. Several examples are discussed, including more general Minkowski metrics and statistical divergences for the comparison of, e.g., histogram data. Furthermore, the framework of relevance learning in LVQ is presented. There, parameters of adaptive distance measures are optimized in the training phase. A practical application of Matrix Relevance LVQ in the context of tumor classification illustrates the approach.

1 Introduction

This contribution summarizes a tutorial talk which was meant as a first introduction to distance and prototype based machine learning techniques. Accordingly, our intention is not to give a complete overview of the field or to review all relevant literature. The paper may serve as a starting point for the interested reader to explore this practically relevant framework and active area of research.

The inference of classification schemes from previous observations, i.e. from labelled example data, is one of the core issues in machine learning [1–4]. A large variety of real world problems can be formulated as classification tasks. Examples include handwritten character recognition, medical diagnoses based on clinical data, pixel-wise segmentation and other image processing tasks, or fault detection in technical systems based on sensor data, to name only a few.

Throughout this contribution we assume that observations are given in terms of real-valued feature vectors in N dimensions. In general, the structure of the

L. Grandinetti et al. (Eds.): BrainComp 2013, LNCS 8603, pp. 100–116, 2014.
DOI: 10.1007/978-3-319-12084-3_9

data can be more complex and may require modified approaches, for instance the *pseudo-Euclidean* embedding of relational data. For this and other extensions of the concepts presented here, we refer the reader to [5,6] and references therein.

A variety of frameworks and training algorithms have been developed for the learning from examples, i.e. the data driven adaptation of parameters in the chosen classification model. They range from classical statistics based methods like Discriminant Analysis to the application of Multilayer Perceptrons or the prominent Support Vector Machine [1–4].

A particularly transparent approach is that of distance or similarity based classification [2,3,5]. Here, observations are directly compared with reference data or prototypes which have been determined in a training process from available examples. The similarity or, more correctly, *dis-similarity* is quantified in terms of a suitable distance measure.[1] The choice of appropriate measures is in the focus of this contribution. Most of the concepts discussed here can be applied in a much broader context, including supervised regression or the unsupervised clustering of data [5]. Here, however, we will limit the discussion to clear-cut classification problems and the use of prototype or reference data based classifiers.

In the next section we discuss two classical methods: the k-Nearest-Neighbor (kNN) approach [2,3,7] and Kohonen's Learning Vector Quantization (LVQ) [8,9] which – in their simplest versions – employ standard Euclidean distance. Mainly in terms of LVQ we discuss how to extend the framework to more general distance measures in Sect. 3.1. The use of divergences for the classification of histograms serves as one example. Section 4 presents the elegant framework of Relevance Learning Vector Quantization as an example for the use of adaptive distance measures. We conclude with a brief summary in Sect. 5.

2 Simple Classifiers Based on Euclidean Distances

When dealing with N-dimensional feature vectors, the use of Euclidean metrics for their pairwise comparison seems natural. In the following we discuss two classical methods which employ this measure in their simplest versions.

2.1 Nearest-Neighbor Classifiers

Arguably the simplest and by far most popular distance based scheme for vectorial data is the k-Nearest-Neighbor (kNN) classifier [2,3,7]. In this classical approach, a given set of P vectors in N-dim. feature space is stored together with labels which indicate their known assignment to one of the C classes:

$$\{\,\mathbf{x}^\mu, y(\mathbf{x}^\mu) = y^\mu\,)\}_{\mu=1}^{P} \quad \text{where} \quad \mathbf{x}^\mu \in \mathbb{R}^N \quad \text{and} \quad y^\mu \in \{1, 2, \ldots, C\}. \tag{1}$$

[1] In this article, we use the term *distance* in its general sense, not necessarily implying symmetry or other metric properties.

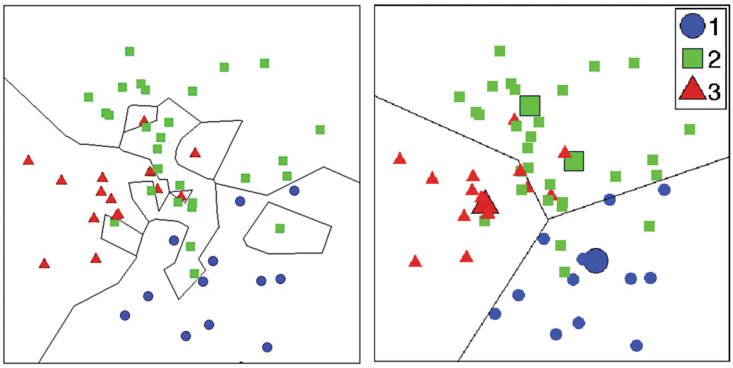

Fig. 1. Illustration of simple classification schemes based on Euclidean distance. Both panels display the same three class data set and decision boundaries represented by solid lines. **Left:** Nearest-Neighbor classifier. **Right:** Nearest Prototype classification, prototypes are marked by larger symbols as indicated by the legend.

An arbitrary feature vector $\mathbf{x}$ is classified according to the distances from the reference samples. In the most basic 1NN scheme, its (squared) Euclidean distance

$$d(\mathbf{x}, \mathbf{x}^\mu) = (\mathbf{x} - \mathbf{x}^\mu)^2 = \sum_{j=1}^{N} \left(x_j - x_j^\mu\right)^2 \tag{2}$$

from all reference samples $\mathbf{x}^\mu$ is computed and the data point is assigned to the class of the *Nearest Neighbor*:

$$y(\mathbf{x}) = y^* = y(\mathbf{x}^*) \quad \text{with} \quad \mathbf{x}^* = \underset{\mathbf{x}^\mu}{\operatorname{argmin}} \left\{ d(\mathbf{x}, \mathbf{x}^\mu) \right\}_{\mu=1}^{P} . \tag{3}$$

In the more general kNN classifier the assignment, Eq. (3), is replaced by a voting among the k closest reference vectors.

The kNN classifier is straightforward to implement and requires no further analysis of the example data in a training phase. Furthermore, theoretical considerations show that kNN schemes can achieve close to Bayes optimal classification if k is selected appropriately [2,3]. As a consequence, the kNN classifier continues to be applied in a variety of practical contexts and often serves as a baseline for comparison with other methods. Figure 1 (left panel) illustrates the 1NN classifier for an artificial three class data set in two dimensions. The prescription (3) results in a piece-wise linear tessellation of feature space.

Two major drawbacks of the approach are evident:

(I) For large data sets, the method involves considerable computational effort in the working phase. The naive implementation of (3) requires the evaluation of P distances and the identification of their minimum for each novel classification. Although clever search and sorting strategies can reduce the computational complexity [3], the basic problem persists for large data sets.

(II) More importantly, class boundaries can become very complex since every example is taken into account on an equal footing. The system is highly sensitive to single, potentially mislabelled examples or outliers. This bears the risk of over-fitting, i.e. the classifier can become too specific to the example set which may result in poor generalization performance with respect to novel data. The effect is clearly mildened when k neighbors are taken into account. However, too large k can yield overly smooth boundaries.

Both problems suggest to reduce the number of reference examples. The representation of the data set by a condensed set of examples was already considered in [10]. A variety of improved selection schemes have been proposed which aim at retaining relevant information contained in the data set, see e.g. [11] and references therein.

2.2 Learning Vector Quantization

Here we consider approaches which compute class representatives without restricting them to be elements of the training set. Each class is represented by at least one vector in a set of M labeled prototypes:

$$\left\{ \mathbf{w}^j, c^j \right\}_{j=1}^{M} \quad \text{where} \quad \mathbf{w}^j \in \mathbb{R}^N \quad \text{and} \quad c^j \in \{1, 2, \ldots C\}. \tag{4}$$

Together with the Euclidean measure, the prototypes parameterize piece-wise linear class boundaries. Similar to the 1NN classifier, a Nearest Prototype Scheme (NPC) assigns an arbitrary feature vector to class

$$y(\mathbf{x}) = c^* \quad \text{where } c^* \text{ is the label of } \mathbf{w}^* = \underset{\mathbf{w}^j}{\operatorname{argmin}} \left\{ d(\mathbf{x}, \mathbf{w}^j) \right\}_{j=1}^{M}. \tag{5}$$

The term *winner* is frequently used for the closest prototype $\mathbf{w}^*$ with respect to data point $\mathbf{x}$. More sophisticated voting rules, probabilistic or *soft* schemes can be devised, but here we limit the discussion to *crisp* classifiers.

The right panel of Fig. 1 illustrates the NPC scheme. The resulting decision boundaries are obviously much smoother than those of the corresponding 1NN classifier (left panel). The NPC scheme is less sensitive to details of the data set which is reflected by the fact that it misclassifies some of the training examples. In comparison to the 1NN scheme, this should result in superior generalization behavior in the presence of noisy examples and outliers.

Arguably the most attractive feature of prototype-based schemes is their interpretability [12]. Prototypes are defined in the feature space of observations and, hence, can be inspected by domain experts in the same way as the sample data. This is in contrast to Multilayer Perceptrons or other model parameterizations which are less transparent. Moreover, prototypes should be - in a sense - *typical* for their classes. Thus, the concept is complementary to, for instance, the Support Vector Machine approach [4] which puts emphasis on *atypical* samples close to the decision boundaries.

LVQ prototypes are determined from the example data by more or less sophisticated training procedures. A conceptually simple idea for their initialization is to compute the class-conditional means, which appears promising when classes are represented by single, more or less spherical clusters. In more realistic situations, LVQ prototypes could be initialized at random in feature space. More reasonably, their initial positions could be determined by means of class-wise unsupervised competitive learning [1–3] prior to the actual supervised training.

In the following we present two prominent prototype-based, iterative training schemes from the family of Learning Vector Quantization algorithms.

Kohonen's LVQ1. Kohonen's original Learning Vector Quantization algorithm [8,9,13], known as LVQ1, constitutes an intuitive, heuristic procedure for the computation of prototypes. It is reminiscent of competitive learning for the purpose of unsupervised Vector Quantization [2].

In LVQ1, single training examples are presented, for instance in randomized order. Upon presentation of example $\{\mathbf{x}, y\}$, the currently closest prototype $\mathbf{w}^*$ is determined in analogy to Eq. (5). Only the winner is updated according to

$$\mathbf{w}^* \leftarrow \mathbf{w}^* + \eta \ \Psi(c^*, y) \ (\mathbf{x} - \mathbf{w}^*) \quad \text{where} \quad \Psi(c, y) = \begin{cases} +1 \text{ if } & c = y \\ -1 \text{ if } & c \neq y. \end{cases} \tag{6}$$

Here, the learning rate $\eta > 0$ controls the step size. Note that Eq. (6) could be re-written as

$$\mathbf{w}^* \leftarrow \mathbf{w}^* - \eta \ \Psi(c^*, y) \ \frac{\partial}{\partial \mathbf{w}^*} \left[\frac{1}{2} (\mathbf{x} - \mathbf{w}^*)^2 \right], \tag{7}$$

formally. The *Winner Takes All* (WTA) prescription moves the prototype $\mathbf{w}^*$ closer to or away from the feature vector if the class labels agree or disagree, respectively. As a consequence, the sample $\mathbf{x}$ – or other feature vectors in its vicinity – will be classified correctly with higher probability after the update. Intuitively, after repeated presentation of the data set, prototypes approach positions which should be typical for the corresponding classes.

A number of variations of the basic scheme have been suggested in the literature, aiming at better generalization ability or more stable convergence behavior, e.g. [9,14–17]. Several modifications update more than one prototype at a time, e.g. LVQ2.1 or LVQ3, or employ adaptive learning rates as for instance the so-called Optimized LVQ (OLVQ) [9]. However, the essential features of LVQ1 – competitive learning and label-dependent updates – are present in all versions of LVQ.

Generalized Learning Vector Quantization. Cost function based approaches [14–17] have attracted particular attention. First of all, convergence properties can be studied analytically in terms of their objective function. Moreover, cost functions allow for systematic extensions of the training schemes, for instance by including adaptive *hyperparameters* in the optimization [18,19].

Here we focus on the so–called Generalized Learning Vector Quantization (GLVQ) as introduced by Sato and Yamada [14,15]. The suggested cost function is given by a sum over examples:

$$E = \sum_{\mu=1}^{P} e^{\mu} \quad \text{with} \quad e^{\mu} = \Phi\left(\frac{d(\mathbf{w}^J, \mathbf{x}^\mu) - d(\mathbf{w}^K, \mathbf{x}^\mu)}{d(\mathbf{w}^J, \mathbf{x}^\mu) + d(\mathbf{w}^K, \mathbf{x}^\mu)}\right), \tag{8}$$

where $\mathbf{w}^J$ and $\mathbf{w}^K$ denote the *closest correct* and *closest incorrect* prototype, respectively, for a particular example $\{\mathbf{x}^\mu, y^\mu\}$. Formally,

$$\begin{aligned} \mathbf{w}^J &= \underset{\mathbf{w}^j}{\operatorname{argmin}} \left\{ d(\mathbf{x}^\mu, \mathbf{w}^j) \mid c^j = y^\mu \right\}_{j=1}^{M} \\ \mathbf{w}^K &= \underset{\mathbf{w}^j}{\operatorname{argmin}} \left\{ d(\mathbf{x}^\mu, \mathbf{w}^j) \mid c^j \neq y^\mu \right\}_{j=1}^{M} . \end{aligned} \tag{9}$$

Popular choices for the monotonically increasing function $\Phi(z)$ in Eq. (8) are the identity $\Phi(z) = z$ or a sigmoidal like $\Phi(z) = 1/[1 + exp(-\gamma\, z)]$ where $\gamma > 0$ controls the *steepness* in the origin [14,20]. Its argument obeys $-1 \leq z \leq 1$, negative values $z < 0$ indicate that the corresponding training example is correctly classified. Note that for large γ the cost function approximates the number of misclassified training data, while for small steepness the minimization of E corresponds to maximizing the margin-like quantities $\left[d(\mathbf{w}^K, \mathbf{x}^\mu) - d(\mathbf{w}^J, \mathbf{x}^\mu)\right]$.

One possible strategy to optimize E for a given data set is *stochastic gradient descent* based on single example presentation [1,2,21,22]. The update step for the winning prototypes $\mathbf{w}^J, \mathbf{w}^K$, given a particular example $\{\mathbf{x}, y\}$, reads

$$\begin{aligned} \mathbf{w}^J &\leftarrow \mathbf{w}^J - \eta \frac{\partial}{\partial \mathbf{w}^J} \Phi(e) = \mathbf{w}^J + \eta\, \Phi'(e) \frac{4 d_K}{(d_J + d_K)^2} \left(\mathbf{x} - \mathbf{w}^J\right) \\ \mathbf{w}^K &\leftarrow \mathbf{w}^K - \eta \frac{\partial}{\partial \mathbf{w}^K} \Phi(e) = \mathbf{w}^K - \eta\, \Phi'(e) \frac{4 d_J}{(d_J + d_K)^2} \left(\mathbf{x} - \mathbf{w}^K\right) \end{aligned} \tag{10}$$

where the abbreviation $d^L = d(\mathbf{w}^L, \mathbf{x})$ for the squared Euclidean distances has been introduced.

Note that in contrast to GLVQ, LVQ1 cannot be interpreted as a stochastic gradient descent, although Eq. (7) involves the gradient of $d(\mathbf{w}^*, \mathbf{x})$. Formal integration yields the function

$$\frac{1}{2} \sum_{\mu=1}^{P} \Psi(c^*, y^\mu)\, (\mathbf{x}^\mu - \mathbf{w}^*)^2$$

which is not differentiable at class borders. Crossing the decision boundary, a different prototype becomes the winner and the sign of Ψ changes discontinuously.

3 Extensions to General Distance Measures

Occasionally it is argued that all distance based methods are bound to fail in high-dimensional feature space due to the so-called *curse of dimensionality* and the related *concentration of norms*, see [23] for a general discussion thereof. The problem is evident in the context of, e.g., density estimation or histogram based techniques. However, we would like to emphasize that the argument does not necessarily carry over to the *comparison* of distances. Consider, for instance, the difference of two squared Euclidean distances

$$d(\mathbf{x}, \mathbf{x}^a) - d(\mathbf{x}, \mathbf{x}^b) = 2\,\mathbf{x} \cdot (\mathbf{x}^b - \mathbf{x}^a) + (\mathbf{x}^a)^2 - (\mathbf{x}^b)^2 \tag{11}$$

which involves the projection of $\mathbf{x}$ into the low-dimensional subspace spanned by reference vectors $\mathbf{x}^a, \mathbf{x}^b$. The *concentration of norms* suggests, indeed, that the last two terms approximately cancel each other in high dimensions, while the first remains non-trivial. Moreover, in the context of LVQ, $\mathbf{x}^a, \mathbf{x}^b$ in (11) are replaced by prototypes which have been determined as *low noise* representatives of the data set.

Euclidean distance appears to be a natural measure and is by far the most popular choice in practice. However, one should be aware that other measures may be more suitable, depending on the nature of the data set at hand [24]. Both the kNN and the LVQ framework facilitate the use of alternative distance measures in a rather straightforward fashion as outlined in the following.

3.1 Example Metrics and More General Measures

Frequently, distances $d(\mathbf{x}, \mathbf{y})$ are required to satisfy the metric properties

$$d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}, \quad d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x}), \quad d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}). \tag{12}$$

However, in the prototype based or kNN classification of a query $\mathbf{x}$, these conditions can be relaxed. For example, the NPC with prototypes $\{\mathbf{w}^j\}$ is still well defined with a non-symmetric measure as long as only one of the two choices, $d(\mathbf{x}, \mathbf{w}^j)$ or $d(\mathbf{w}^j, \mathbf{x})$, is used consistently. Distances between different prototypes or between two data points are never considered explicitly in the scheme.

A large variety of distance measures can be employed for classification tasks. Discretized data, for instance, can be compared by means of the Hamming distance or more general string metrics. Specific measures have been devised for *functional data* where the order of the observed features is relevant, see [25,26] for examples.

In the following we outline how, quite generally, differentiable distance measures can be made use of in LVQ schemes. Then we briefly discuss three example families of measures which constitute important alternatives to the standard Euclidean choice.

Incorporation of Differentiable Distances in LVQ Schemes. In the working phase of kNN or prototype based classification, essentially any meaningful distance measure can be employed which is appropriate for the problem at hand. An important restriction applies, however, if gradient based training schemes like LVQ1 or GLVQ are used which require that the underlying distance is differentiable. Under this condition, a general LVQ1-like update can be written as

$$\mathbf{w}^* \leftarrow \mathbf{w}^* - \eta \ \Psi(c^*, y) \ \frac{\partial}{\partial \mathbf{w}^*} d(\mathbf{w}^*, \mathbf{x}) \tag{13}$$

in analogy with Eq. (7).

Similarly, the Euclidean distance in the GLVQ cost function (8) can be replaced by a more general, differentiable measure, yielding the update

$$\begin{aligned} \mathbf{w}^J &\leftarrow \mathbf{w}^J + \eta \, \Phi'(e) \, \frac{2 d_K}{(d_J + d_K)^2} \, \frac{\partial}{\partial \mathbf{w}^J} \, d(\mathbf{w}^J, \mathbf{x}) \\ \mathbf{w}^K &\leftarrow \mathbf{w}^K - \eta \ \Phi'(e) \, \frac{2 d_J}{(d_J + d_K)^2} \, \frac{\partial}{\partial \mathbf{w}^K} \, d(\mathbf{w}^K, \mathbf{x}) \end{aligned} \tag{14}$$

where the winners and all other quantities are defined as in (10). In the following we highlight a few families of differentiable distance measures which can be incorporated into LVQ in a straightforward way.

Minkowski Distances. A prominent class of distances corresponds to the so-called Minkowski measures

$$d_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{j=1}^{N} |x_j - y_j|^p \right)^{1/p} \tag{15}$$

with $p > 0$ which includes the standard Euclidean distance for $p = 2$ or the so–called Manhattan metric for $p = 1$. Note that (15) is a metric only for $p \geq 1$, while it violates (12) for $p < 1$. However, in the latter case, $(d_p(\mathbf{x}, \mathbf{y}))^p$ becomes a metric [27]. Note that the Euclidean distance can be determined using the inner product

$$\langle \mathbf{x}, \mathbf{y} \rangle = \textstyle\sum_{j=1}^{N} x_j \cdot y_j \tag{16}$$

by computing

$$d_2(\mathbf{x}, \mathbf{y}) = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle^2 - 2 \cdot \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{y}, \mathbf{y} \rangle^2}. \tag{17}$$

For $p \neq 2$ and $p \geq 1$, an analogous calculation can be done using semi-inner products [28,29]. The use of Minkowski metrics with $p \neq 2$ has proven advantageous in several practical applications, e.g. [30,31], which can be accompanied by appropriate dimensionality reduction schemes, e.g. principal component analysis (PCA) [32,33]. Minkowski distances are either differentiable or can be replaced by differentiable approximations, see [27] and references therein. Figure 2 illustrates the influence of the parameter p in (15). It displays the unit circles in two dimensions corresponding to different Minkowski distances.

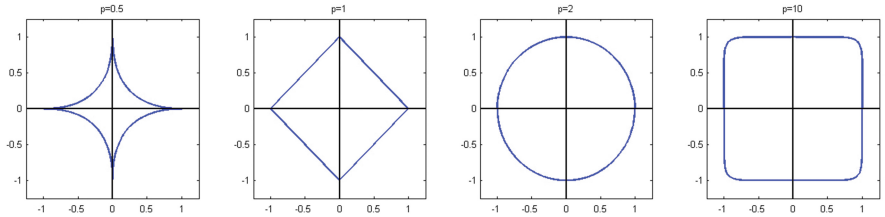

Fig. 2. *Unit circles* corresponding to Minkowski metrics, Eq. (15), in two dimensions with, from left to right, $p = 0.5$, $p = 1$ (Manhattan), $p = 2$ (Euclidean), and $p = 10$.

Divergences. In many practical problems, properties of the data are represented by histograms. Prominent examples are the characterization of images by color histograms or the *bag of words* representation for texts. In other domains, *intensity spectra* or other non-negative and normalizable functional data represent the objects of interest [34]. A large variety of statistical divergences are tailored for the comparison of positive measures or probability densities. Arguably, the non-symmetric Kullback-Leibler divergence is the most prominent example [35]. Here we exemplify the concept in terms of the symmetric Cauchy-Schwarz divergence

$$d_{CS}(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \log \left[\langle \mathbf{x}, \mathbf{x} \rangle \cdot \langle \mathbf{y}, \mathbf{y} \rangle \right] - \log \left[\langle \mathbf{x}, \mathbf{y} \rangle \right] \tag{18}$$

which is obviously differentiable [36]. Figure 3 illustrates how d_{CS} differs from the standard Euclidean distance: Three normalized 50-bin histograms are displayed which satisfy $(\mathbf{x}^a - \mathbf{x}^b)^2 = (\mathbf{x}^c - \mathbf{x}^b)^2$. However, according to the Cauchy-Schwarz measure, $d_{CS}(\mathbf{x}^a, \mathbf{x}^b) \approx 1/2\, d_{CS}(\mathbf{x}^c, \mathbf{x}^b)$, implying that the single peak $\mathbf{x}^a$ is considered to be closer to the broad unimodal $\mathbf{x}^b$ than the double peak histogram $\mathbf{x}^c$.

The incorporation of symmetric and non-symmetric, differentiable divergences into GLVQ training and classification is introduced in [37]. As an application example, the detection of Mosaic Disease in Cassava plants based on various image histograms is discussed there.

Kernel Distances. Kernel distances [38] can also be incorporated in prototype based learning and classification approaches, see e.g. [39,40]. The so-called kernel trick consists of an implicit, in general non-linear, mapping to a potentially infinite dimensional space. This mapping space is equipped with an inner product which can be calculated from original data in terms of a so–called kernel $\kappa(\mathbf{x}, \mathbf{y})$ [41,42]. The corresponding kernel distance is calculated as

$$d_\kappa(\mathbf{x}, \mathbf{y}) = \sqrt{\kappa(\mathbf{x}, \mathbf{x})^2 - 2 \cdot \kappa(\mathbf{x}, \mathbf{y}) + \kappa(\mathbf{y}, \mathbf{y})^2} \tag{19}$$

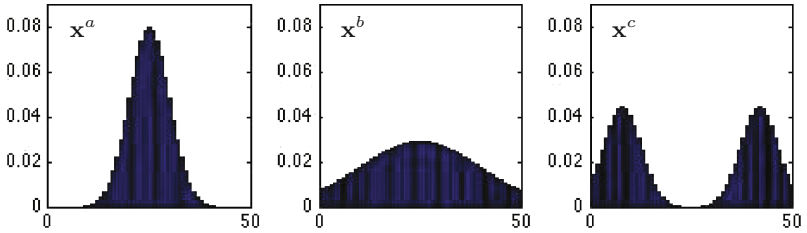

Fig. 3. Three normalized histograms $\mathbf{x}^a, \mathbf{x}^b, \mathbf{x}^c$ with 50 bins each. The pair-wise comparison in terms of Euclidean distance and Cauchy-Schwarz divergence, cf. Eq. (18), as discussed in Sect. 3.1

in complete analogy to the inner product based Euclidean distance calculation (17). A famous example is the Gaussian kernel

$$\kappa_G(\mathbf{x}, \mathbf{y}) = \exp\left(\frac{-\left(d_2(\mathbf{x}, \mathbf{y})\right)^2}{2\sigma^2}\right) \tag{20}$$

with the kernel width σ.

The application of kernel distances frequently translates non-linear complex classification tasks into easier, linearly separable problems [41], as demonstrated for, e.g., image based face recognition in [43]. For LVQ schemes, the kernel distance is assumed to be differentiable, which implies that also the kernel $\kappa(\mathbf{x}, \mathbf{y})$ has to be differentiable [44].

4 Adaptive Distances and Relevance LVQ

In an ideal situation, insight into the problem suggests the use of a specific, fixed distance measure. Very often, however, prior knowledge is limited and only a suitable parametric form of the distance can be specified. In Relevance Learning, a particularly elegant extension of LVQ, the corresponding parameters are adapted in the same data driven training process that identifies the prototypes.

4.1 Matrix Relevance LVQ

In the following we discuss Matrix Relevance LVQ as an extension of the basic Euclidean scheme [20]. An obvious problem of the standard measure is that all dimensions are taken into account on the same footing. First of all, some of the features may be very *noisy* and potentially corrupt the classifier. Furthermore, features can be correlated or scale very differently. Euclidean or other pre-defined measures are sensitive to rescaling and more general linear transformations of the features. Consequently, their naive use can be problematic in practice. Matrix Relevance LVQ in its simplest form addresses these problems by using a generalized quadratic distance of the form

$$d(\mathbf{x}, \mathbf{w}) = (\mathbf{x} - \mathbf{w})^\top \Lambda (\mathbf{x} - \mathbf{w}) \quad \text{with} \quad \Lambda = \Omega^\top \Omega \quad \text{where} \quad \Lambda, \Omega \in \mathbb{R}^{N \times N}. \tag{21}$$

Here the specific parameterization of Λ as a square guarantees that the distance is positive semi-definite: $d(\mathbf{x}, \mathbf{w}) \geq 0$.

The elements of the matrix Ω are considered adaptive quantities in the training process. The distance (21) is differentiable with respect to $\mathbf{w}$ and Ω:

$$\frac{\partial d(\mathbf{w}, \mathbf{x})}{\partial \mathbf{w}} = \Omega^\top \Omega \, (\mathbf{w} - \mathbf{x}), \qquad \frac{\partial d(\mathbf{w}, \mathbf{x})}{\partial \Omega} = \Omega \, (\mathbf{w} - \mathbf{x})(\mathbf{w} - \mathbf{x})^\top \tag{22}$$

which facilitates gradient based updates of prototypes and distance measure. In the corresponding extension of LVQ1-like updates, the WTA prototype update (13) is combined with

$$\Omega \leftarrow \Omega - \eta_\Omega \; \Psi(c^*, y) \; \frac{\partial}{\partial \Omega} \, d(\mathbf{w}^*, \mathbf{x}). \tag{23}$$

Generalized Matrix Relevance LVQ (GMLVQ) updates Ω according to

$$\Omega \leftarrow \Omega - \eta_\Omega \, \Phi'(e) \left(\frac{2 d_K}{(d_J + d_K)^2} \frac{\partial \, d(\mathbf{w}^J, \mathbf{x})}{\partial \Omega} - \frac{2 d_J}{(d_J + d_K)^2} \frac{\partial \, d(\mathbf{w}^K, \mathbf{x})}{\partial \Omega} \right) \tag{24}$$

together with the prototype updates (14). Both, (23) and (24) can be followed by an explicit normalization to enforce $\sum_{ij} \Omega_{ij}^2 = 1$. The matrix learning rate η_Ω is frequently chosen smaller than that of the prototype updates. We refer the reader to [20,45] for details and the full form of the updates and a discussion of their variants.

Note that the above correspond to only the simplest versions of matrix relevance learning. A number of non-trivial variations have been suggested, including the use of prototype- or class-specific localized matrices which yield piece-wise quadratic decision boundaries in feature space [20]. Rectangular matrices Ω can be employed in order to avoid the adaptation of $\mathcal{O}(N^2)$ degrees of freedom in high-dimensional data sets [45]. They facilitate also the discriminative low-dim. representation or visualization of labeled data sets [45,46]. The restriction to diagonal matrices Ω and Λ reduces the scheme to a weighting of single features, which had been introduced earlier as RLVQ [47] and GRLVQ [48], respectively. Formally, Euclidean LVQ versions are recovered by setting Ω proportional to the N-dimensional identity matrix.

Similar parameterized distance measures have been used in the context of various classification frameworks. For instance, the cost function based optimization of a quadratic distance (21) can be integrated in an extended kNN approach as introduced in [49], see also references therein. As another example we would like to mention Radial Basis Function networks [1] which, in combination with relevance learning, have been applied in problems of vital importance recently [50].

A Matlab toolbox *Relevance and Matrix adaptation in Learning Vector Quantization*, including GMLVQ and a number of variants, is available at the website [51].

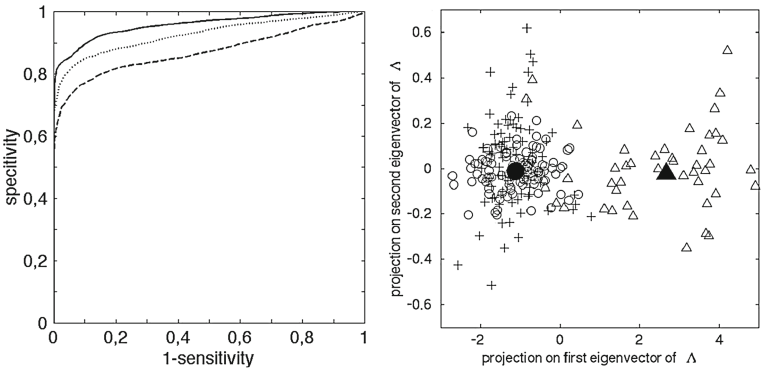

Fig. 4. Left: ROC curves as obtained by GLVQ (dashed), GRLVQ (dotted), GMLVQ (solid line) with respect to the detection of malignant ACC, see Sect. 4.3. **Right:** Visualization of the data set, displaying projections on the leading eigenvalues of Λ. In addition to malignant ACC (triangles) and benign ACA (circles), healthy control data (crosses) are displayed. Prototypes for ACA and ACC are marked by filled symbols.

4.2 Interpretation of the Relevance Matrix

It is instructive to note that the quadratic distance (21) can be rewritten as $d(\mathbf{w}, \mathbf{x}) = [\Omega(\mathbf{w} - \mathbf{x})]^2$, implying that plain Euclidean distance is applied after a linear transformation of feature vectors and prototypes. The transformation can account for the above mentioned problems of noisy or correlated features by assigning weights to single dimensions and pairs of features, respectively. Note that the diagonal element $\Lambda_{jj} = \sum_i \Omega_{ij}^2$ quantifies the total contribution of the original feature dimension j to the linear combinations $[\Omega(\mathbf{w} - \mathbf{x})]_i$.

The direct interpretation of Λ_{jj} as the *relevance* of feature j for the classification is only justified if different features are of the same magnitude, typically. This can be achieved by, for instance, a *z-score transformation* in a preprocessing step, such that $\sum_\mu x_j^\mu / P = 0$ and $\sum_\mu (x_j^\mu)^2 / P = 1$. Additional measures have to be taken in the presence of strongly correlated or linearly dependent features, see [12] for a detailed discussion of the interpretation of Λ and related regularization techniques.

It is instructive to note that, given Λ, a continuum of matrices Ω satisfies $\Omega^\top \Omega = \Lambda$. However, this does not pose a problem, since the ambiguity reflects invariances of the distance measure with respect to reflections or rotations of the data. For convenience, e.g. when comparing the results of different training processes, one can resort to a canonical representation of Ω in terms of the eigenvectors of Λ, see [12] for a more detailed discussion.

4.3 Example Application: Classification of Adrenal Tumors

We briefly illustrate the MRLVQ approach in terms of a recent medical application [52,53]. Tumors of the adrenal gland occur in an estimated 1–2 % of

the population and are mostly found incidentally. The non-invasive differentiation between malignant Adrenocortical Carcinoma (ACC) and benign Adenomas (ACA) constitutes a diagnostic challenge of great significance. To this end, a panel of 32 steroid biomarkers – produced by the adrenal gland - has been suggested in [52] where details are given. The 24h excretion of these steroids has been analysed for a number of example patients with confirmed diagnosis, retrospectively. Preprocessing and normalization steps are also detailed in [52,53]. The available data set was analysed by means of GMLVQ in its simplest setting, employing one 32-dim. prototype per class and an adaptive $\Omega \in \mathbb{R}^{32\times 32}$.

Standard validation procedures, for details see [52,53], were used to demonstrate that the resulting classifier achieves very good sensitivity (true positive rate) and specificity (1-false positive rate) with respect to the detection of malignancy. The obtained Receiver Operator Characteristics (ROC) [54] is shown in Fig. 4 (left panel). For comparison, the ROC is also displayed for simple GLVQ using the plain Euclidean distance in $\mathbb{R}^{32}$ and for a system restricted to an adaptive, diagonal Λ, which corresponds to GRLVQ [48]. Evidently, relevance learning and in particular the matrix scheme improves the performance significantly over the use of the naive Euclidean distance.

The resulting relevance matrix, see [53], shows that a few of the steroid markers play a dominant role in the classification as marked by large diagonal elements Λ_{jj}. Based on these results, a reduced panel of 9 markers was proposed in [52]. This reduction facilitates an efficient technical realization of the method, while the performance is essentially retained. The method constitutes a promising tool for the sensitive and specific differential diagnosis of ACC in clinical practice [52].

An additional feature of matrix relevance learning becomes apparent in this application example. Typically, relevance matrices become low rank in the course of training. Theoretical considerations which support this general, empirical finding are presented in [55]. As a consequence, the dominating eigenvectors of the relevance matrix can be used for the discriminative visualization of the labelled examples. Figure 4 (right panel) displays the projections of all ACA and ACC data and the obtained prototypes on the first two eigenvectors of Λ. In addition, healthy control data is displayed which was not explicitly used in the training process. The example demonstrates how the combination of prototype based and relevance learning can provide novel insight and facilitates fruitful discussions with the domain experts. For a similar application of GMLVQ in a different medical context, see [56].

5 Summary

This contribution provides only a first introduction to distance based classification schemes. To a large extent, the discussion is presented in terms of two classical approaches: the k-Nearest-Neighbor classifier and Kohonen's Learning Vector Quantization. The latter requires a training phase which tunes the classifier according to available training data. Examples for heuristic and cost function

based training prescriptions are given. Mainly in the context of LVQ the use of generalized dissimilarity measures is discussed, which go beyond the standard choice of Euclidean distance. Relevance Learning is presented as an extension of LVQ, which makes use of adaptive distances. Their data driven optimization can be integrated naturally in the LVQ training process. As an example, matrix relevance learning is briefly presented and illustrated in terms of an application example in the medical domain.

This article and the suggested references can merely serve as a starting point for the interested reader. It is far from giving a complete overview of this fascinating area of ongoing fundamental and application oriented research.

References

1. Bishop, C.: Pattern Recognition and Machine Learning. Cambridge University Press, Cambridge (2007)
2. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer, New York (2009)
3. Duda, R., Hart, P., Storck, D.: Pattern Classification, 2nd edn. Wiley, New York (2001)
4. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
5. Biehl, M., Hammer, B., Verleysen, M., Villmann, T. (eds.): Similarity-Based Clustering. LNCS, vol. 5400. Springer, Heidelberg (2009)
6. Hammer, B., Schleif, F.-M., Zhu, X.: Relational extensions of learning vector quantization. In: Lu, B.-L., Zhang, L., Kwok, J. (eds.) ICONIP 2011, Part II. LNCS, vol. 7063, pp. 481–489. Springer, Heidelberg (2011)
7. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theor. **13**, 21–27 (1967)
8. Kohonen, T.: Self-Organizing Maps, 2nd edn. Springer, Heidelberg (1997)
9. Kohonen, T.: Improved versions of learning vector quantization. In: International Joint Conference on Neural Networks, vol. 1, pp. 545–550 (1990)
10. Hart, P.: The condensed nearest neighbor rule. IEEE Trans. Inf. Theor. **14**, 515–516 (1968)
11. Wu, Y., Ianakiev, K., Govindaraju, V.: Improved k-nearest neighbor classification. Pattern Recogn. **35**, 2311–2318 (2002)
12. Strickert, M., Hammer, B., Villmann, T., Biehl, M.: Regularization and improved interpretation of linear data mappings and adaptive distance measures. In: Proceedings of the IEEE Symposium on Computational Intelligence (IEEE SSCI), IEEE, vol. 2013, p. 8 (2013)
13. Helsinki University of Technology: Bibliography on the Self-Organizing Map (SOM) and Learning Vector Quantization (LVQ). Neural Networks Research Centre, HUT (2002)
14. Sato, A., Yamada, K.: Generalized Learning vector quantization. In: Touretzky, D.S., Hasselmo, M.E. (eds.) Proceedings of the 1995 Conference, Cambridge, MA, USA, MIT Press. vol. 8, Advances in Neural Information Processing Systems, pp. 423–429 (1996)
15. Sato, A., Yamada, K.: An analysis of convergence in generalized LVQ. In: Niklasson, L., Bodn, M., Ziemke, T. (eds.) Proceedings of the International Conference on Artificial Neural Networks, Springer, pp. 170–176 (1998)

16. Seo, S., Obermayer, K.: Soft learning vector quantization. Neural Comput. **15**(7), 1589–1604 (2003)
17. Seo, S., Bode, M., Obermayer, K.: Soft nearest prototype classification. Trans. Neural Netw. **14**, 390–398 (2003)
18. Seo, S., Obermayer, K.: Dynamic hyperparameter scaling method for LVQ algorithms. In: IJCNN'06, International Joint Conference on Neural Networks, IEEE, pp. 3196–3203 (2006)
19. Schneider, P., Biehl, M., Hammer, B.: Hyperparameter learning in probabilistic prototype-based models. Neurocomputing **73**(7–9), 1117–1124 (2010)
20. Schneider, P., Biehl, M., Hammer, B.: Adaptive relevance matrices in learning vector quantization. Neural Comput. **21**(12), 3532–3561 (2009)
21. Robbins, H., Monro, S.: A stochastic approximation method. Ann. Math. Stat. **22**, 405 (1951)
22. Bottou, L.: Online algorithms and stochastic approximations. In: Saad, D. (ed.) Online Learning and Neural Networks. Cambridge University Press, Cambridge (1998)
23. Lee, J., Verleysen, M.: Nonlinear Dimension Reduction. Springer, New York (2007)
24. Hammer, B., Villmann, T.: Classification using non-standard metrics. In: Verleysen, M. (ed.) European Symposium on Artificial Neural Networks, ESANN 2005, pp. 303–316. d-side publishing (2005)
25. Lee, J., Verleysen, M.: Generalization of the Lp-norm for time series and its application to self-organizing maps. In: Cottrell, M. (ed.) Proceedings of the Workshop on Self-Organizing Maps (WSOM), Paris, Sorbonne, pp. 733–740 (2005)
26. Villmann, T., Hammer, B.: Functional principal component learning using Oja's method and Sobolev norms. In: Príncipe, J.C., Miikkulainen, R. (eds.) WSOM 2009. LNCS, vol. 5629, pp. 325–333. Springer, Heidelberg (2009)
27. Lange, M., Villmann, T.: Derivatives of Lp-norms and their approximations. Machine Learning Reports MLR-04-2013, pp. 43–59 (2013)
28. Giles, J.: Classes of semi-inner-product spaces. Trans. Am. Math. Soc. **129**, 436–446 (1967)
29. Lumer, G.: Semi-inner-product spaces. Trans. Am. Math. Soc. **100**, 29–43 (1961)
30. Golubitsky, O., Watt, S.: Distance-based classification of handwritten symbols. Int. J. Doc. Anal. Recogn. (IJDAR) **13**(2), 133–146 (2010)
31. Biehl, M., Breitling, R., Li, Y.: Analysis of tiling microarray data by learning vector quantization and relevance learning. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 880–889. Springer, Heidelberg (2007)
32. Joliffe, I.: Principal Component Analysis. Springer, New York (2002)
33. Biehl, M., Kästner, M., Lange, M., Villmann, T.: Non-euclidean principal component analysis and Oja's learning rule – theoretical aspects. In: Estevez, P.A., Principe, J.C., Zegers, P. (eds.) Advances in Self-Organizing Maps. AISC, vol. 198, pp. 23–34. Springer, Heidelberg (2013)
34. Villmann, T., Kästner, M., Backhaus, A., Seiffert, U.: Processing hyperspectral data in machine learning. In: Verleysen, M. (ed.) European Symposium on Artificial Neural Networks, ESANN 2013, p. 6. d-side publishing (2013)
35. Kullback, S., Leibler, R.: On information and sufficiency. Ann. Math. Stat. **22**, 79–86 (1951)
36. Villmann, T., Haase, S.: Divergence based vector quantization. Neural Comput. **23**(5), 1343–1392 (2011)

37. Mwebaze, E., Schneider, P., Schleif, F.M., Aduwo, J., Quinn, J., Haase, S., Villmann, T., Biehl, M.: Divergence based classification and learning vector quantization. Neurocomputing **74**, 1429–1435 (2011)
38. Schölkopf, B.: The kernel trick for distances. In: Tresp, V. (ed.) Advances in Neural Information Processing Systems, pp. 301–307. MIT Press, Cambridg (2001)
39. Inokuchi, R., Miyamoto, S.: LVQ clustering and SOM using a kernel function. In: Proceedings of the 2004 IEEE International Conference on Fuzzy Systems, vol. 3, pp. 1497–1500 (2004)
40. Schleif, F.-M., Villmann, T., Hammer, B., Schneider, P., Biehl, M.: Generalized derivative based kernelized learning vector quantization. In: Fyfe, C., Tino, P., Charles, D., Garcia-Osorio, C., Yin, H. (eds.) IDEAL 2010. LNCS, vol. 6283, pp. 21–28. Springer, Heidelberg (2010)
41. Schölkopf, B., Smola, A.: Learning with Kernels. MIT Press, Cambridge (2002)
42. Steinwart, I.: On the influence of the kernel on the consistency of support vector machines. J. Mach. Learn. Res. **2**, 67–93 (2001)
43. Villmann, T., Kästner, M., Nebel, D., Riedel, M.: ICMLA face recognition challenge - results of the team 'Computational Intelligence Mittweida'. In: Proceedings of the International Conference on Machine Learning Applications (ICMLA'12), pp. 7–10. IEEE Computer Society Press (2012)
44. Villmann, T., Haase, S., Kästner, M.: Gradient based learning in vector quantization using differentiable kernels. In: Estevez, P.A., Principe, J.C., Zegers, P. (eds.) Advances in Self-Organizing Maps. AISC, vol. 198, pp. 193–204. Springer, Heidelberg (2013)
45. Bunte, K., Schneider, P., Hammer, B., Schleif, F.M., Villmann, T., Biehl, M.: Limited rank matrix learning, discriminative dimension reduction, and visualization. Neural Netw. **26**, 159–173 (2012)
46. Biehl, M., Bunte, K., Schleif, F.M., Schneider, P., Villmann, T.: Large margin linear discriminative visualization by matrix relevance learning. In: Proceedings of the WCCI 2012 - IEEE World Congress on Computational Intelligence, IEEE Press (2012)
47. Bojer, T., Hammer, B., Schunk, D., von Toschanowitz, K.T.: Relevance determination in learning vector quantization. In: Verleysen, M. (ed.) European Symposium on Artificial Neural Networks, pp. 271–276 (2001)
48. Hammer, B., Villmann, T.: Generalized relevance learning vector quantization. Neural Netw. **15**(8–9), 1059–1068 (2002)
49. Weinberger, K., Blitzer, J., Saul, L.: Distance metric learning for large margin nearest neighbor classification. In: Weiss, Y., Schölkopf, B., Platt, J. (eds.) Advances in Neural Information Processing Systems, vol. 18, pp. 1473–1480. MIT Press, Cambridge (2006)
50. Backhaus, A., Ashok, P., Praveen, B., Dholakia, K., Seiffert, U.: Classifying Scotch Whisky from near-infrared Raman spectra with a radial basis function network with relevance learning. In: Verleysen, M. (ed.) European symposium on Artificial Neural Networks, vol. 2012, pp. 411–416 (2012)
51. Biehl, M., Bunte, K., Schneider, P.: Relevance and matrix adaptation in learning vector quantization (2013). http://matlabserver.cs.rug.nl/gmlvqweb/web
52. Arlt, W., Biehl, M., Taylor, A., Hahner, S., Libe, R., Hughes, B., Schneider, P., Smith, D., Stiekema, H., Krone, N., Porfiri, E., Opocher, G., Bertherat, J., Mantero, F., Allolio, B., Terzolo, M., Nightingale, P., Shackleton, C., Bertagna, X., Fassnacht, M., Stewart, P.: Urine steroid metabolomics as a biomarker tool for detecting malignancy in adrenal tumors. J. Clin. Endocrinol. Metab. **96**, 3775–3784 (2011)

53. Biehl, M., Schneider, P., Smith, D., Stiekema, H., Taylor, A., Hughes, B., Shackleton, C., Stewart, P., Arlt, W.: Matrix relevance LVQ in steroid metabolomics based classification of adrenal tumors. In: Verleysen, M. (ed.) 20th European Symposium on Artificial Neural Networks (ESANN 2012), pp. 423–428, d-side publishing (2012)
54. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. **27**, 861–874 (2006)
55. Biehl, M., Hammer, B., Schleif, F.M., Schneider, P., Villmann, T.: Stationarity of matrix relevance learning vector quantization. Technical report MLR-01-2009, Machine Learning Reports, University of Leipzig (2009)
56. Biehl, M., Bunte, K., Schneider, P.: Analysis of flow cytometry data by matrix relevance learning vector quantization. PLoS ONE **8**(3), e59401 (2013)