Similarity Estimation using Bayes Ensembles

Tobias Emrich, Franz Graf, Hans-Peter Kriegel, Matthias Schubert, and Marisa Thoma

Ludwig-Maximilians-Universität München
Oettingenstr. 67, Munich, Germany
{emrich,graf,kriegel,schubert,thoma}@dbs.ifi.lmu.de

Abstract. Similarity search and data mining often rely on distance or similarity functions in order to provide meaningful results and semantically meaningful patterns. However, standard distance measures like L_p-norms are often not capable to accurately mirror the expected similarity between two objects. To bridge the so-called semantic gap between feature representation and object similarity, the distance function has to be adjusted to the current application context or user. In this paper, we propose a new probabilistic framework for estimating a similarity value based on a Bayesian setting. In our framework, distance comparisons are modeled based on distribution functions on the difference vectors. To combine these functions, a similarity score is computed by an Ensemble of weak Bayesian learners for each dimension in the feature space. To find independent dimensions of maximum meaning, we apply a space transformation based on eigenvalue decomposition. In our experiments, we demonstrate that our new method shows promising results compared to related Mahalanobis learners on several test data sets w.r.t. nearest-neighbor classification and precision-recall-graphs.

Keywords

similarity estimation, distance learning, supervised learning

1 Introduction

Learning similarity functions is an important task for multimedia retrieval and data mining. In data mining, distance measures can be used in various algorithms for classification and clustering. To improve classification, the learned distance measure can be plugged into any instance-based learner like kNN classification. Though clustering is basically an unsupervised problem, learning a similarity function on a small set of manually annotated objects is often enough to guide clustering algorithms to group semantically more similar objects. For similarity search, adaptive similarity measures provide a powerful method to bridge the semantic gap between feature representations and user expectations. In most settings, the similarity between two objects cannot be described by a standardized distance measure fitting all applications. Instead, object similarity

is often a matter of application context and personal preference. Thus, two objects might be similar in one context while they are not very similar in another context. For example, assume an image collection of various general images of persons, vehicles, animals, and buildings. In this context, a picture showing a red Ferrari will be considered as quite similar to a picture of a red Volkswagen. Now, take the same images and put them into a different context like a catalogue of rental cars. In this more specialized context, both pictures will most likely be considered as dissimilar. An important assumption in this paper is that there is no exact value specifying object similarity. Instead, we consider object similarity as the probability that a user would label the objects as similar.

Learning a distance or similarity function requires a general framework for comparing objects. In most established approaches to similarity learning, this framework is provided by using Mahalanobis distances or quadratic forms. In general, a Mahalanobis distance can be considered to be the Euclidean distance in a linear transformation of the original feature space. Thus, Mahalanobis distances are metric distance functions guaranteeing reflexivity, symmetry and the triangular inequality. Furthermore, the computed dissimilarity of two objects might be increased infinitely. In this paper, we argue that these mathematical characteristics are unnecessarily strict and sometimes even against intuition when trying to construct a similarity measure. For example, it is known from cognition science that humans do not distinguish dissimilar objects to an infinite degree. Thus, a human would not care whether object o_1 is more dissimilar to the query object q than object o_2 after having decided that both objects o_1, o_2 have nothing in common with the query object q. Furthermore, it is questionable if characteristics such as strictness are necessary for successful similarity search. In most feature transformations, it is possible that two different objects are mapped to the same feature representation. Thus, even if we can guarantee that two objects having a zero distance are represented by the same feature description, we have no guarantee that the corresponding objects should be considered to be maximally similar as well.

In this paper, we describe similarity in a different way by considering it as the probability that an object o is relevant for a similarity query object q. The core idea of our similarity estimation approach is to consider each feature as evidence for similarity or dissimilarity. Thus, we can express the implication of a certain feature i to the similarity of objects o and q as a probability $p(\text{SIMILAR}(o, q) \mid (o[i] - q[i]))$. To calculate this probability, we employ a simple one-dimensional Bayes estimate (BE). However, to build a statement comprising all available information about object similarity, we do not build the joint probability over all features. We argue that in most applications considering a single feature it is not sufficient to decide either similarity or dissimilarity. Thus, to derive a joined estimation considering all available features, we average the probabilities derived from each BE. Our new estimate is basically an Ensemble of weak Bayesian learners. Therefore, we call our new dissimilarity function Bayes Ensemble Distance (BED). A major benefit of BED is that dissimilarity is very insensitive to outlier values in a single dimension which is a drawback of

classical L_p-norm based measures. The major factors to successfully employing an Ensemble of learners are the quality and the independence of the underlying weak classifiers. Therefore, we will introduce a new optimization problem that derives a linear transformation of the feature space, allowing the construction of more descriptive BEs. To conclude, the contributions of this paper are:

- A discussion about L_p-norms and Mahalanobis distances for modelling object similarity.
- A new framework for similarity estimation that is built on an Ensemble of Bayes learners.
- An optimization method for generating a linear transformation of the feature space that is aimed at deriving independent features which are suitable for training high quality weak classifiers.

The rest of the paper is organized as follows. In Sect. 2, we discuss L_p norm and Mahalanobis distances for modeling object similarity. Our new framework for modeling object similarity is described in Sect. 3. In Sect. 4, we introduce an optimization problem to derive an affine transformation that allows the training of more accurate Bayes estimates. Section 5 briefly reviews related similarity learners. Afterwards, Sect. 6 illustrates the results of our experimental evaluation comparing our new method with related metric learners on several UCI classification datasets and two image retrieval data sets. Finally, Sect. 7 concludes the paper with a summary and some directions for future work.

2 L_p-norms and Problem Definition

The task of similarity learning is to find a function mapping a pair of objects o_1, o_2 to a similarity value $\text{SIM}(o_1, o_2)$ describing how strongly the first object resembles the other one in the best possible way. To train this function, it is necessary to have training examples representing the notion of similarity which underlies the given application. Let us note that there might be various notions of similarity on the same data set depending on the application context or even the current user.

Basically, there are two categories of examples used for learning similarity functions. The first type is providing class labels to a training set indicating that objects with equal labels are similar and objects with different labels are considered as dissimilar. Most machine learning approaches in metric learning use class labels because most of the proposed methods in this area aim at improving the accuracy of instance-based learners. One important advantage of this type of labeling is that there is a large variety of classification data sets available. Additionally, having n labeled objects results in $\frac{n \cdot (n-1)}{2}$ labeled object pairs. Finally, in classification data sets the labeling is usually quite consistent because the classes are usually reproducible by several persons. As a drawback of this approach, it is required to find an universal set of classes before learning a similarity function. Thus, this type of user feedback is difficult to use when learning similarity measures for similarity search. The second type of user feedback is

direct relevance feedback providing a similarity value for a set of object pairs. Using relevance feedback allows to determine a degree of similarity for each pair and thus, the similarity information is not necessarily binary. Additionally, relevance feedback does not require to define explicitly known classes and is thus more attractive for similarity search systems. A drawback of relevance feedback is that labelling a sufficiently large set of object pairs with similarity scores is usually much more strenuous than labelling objects with classes. Furthermore, it is often much more difficult to generate a consistent labelling because there usually are no well-defined criteria for object similarity.

After describing the labels of our examples, we will now formalize our object descriptions, i.e. the feature vectors. A feature is a type of observation about an object and the corresponding feature value describes how an object behaves w.r.t. this type of observation. Mathematically, we will treat a feature F as a numerical value $x_F \in \mathbb{R}$. Considering a predefined number of features d leads to a feature vector $x \in \mathbb{R}^d$. Formally, a training example in our setting is a triple (x_1, x_2, y) where $x_1, x_2 \in \mathbb{R}^d$ are two d-dimensional feature vectors and $y \in [0, \ldots, l]$ is a dissimilarity score, i.e. a 0 represents maximum similarity whereas l describes maximum dissimilarity. In case of class labels, we assign 1 to dissimilar and 0 to similar objects. The most common approach for describing object similarity is to sum up the differences of feature values which is the basis of L_p-norm-based similarity. Given two feature vectors $x_1, x_2 \in \mathbb{R}^d$, the L_p-norms are defined as:

$$L_p(x_1, x_2) = \left(\sum_{i=1}^{d} |x_{1,i} - x_{2,i}|^p \right)^{\frac{1}{p}}$$

For $p = 2$, the L_p-norm is called Euclidean distance which is the most common distance metric in similarity search and distance-based data mining. Semantically, we can interpret the L_p-norm as an evidence framework. Each feature represents an observation about an object and the difference of feature values determines how similar two objects behave with respect to this observation. Since a single observation is usually not enough to decide similarity, all observations are combined. By summing up over the differences for each observation, the L_p-norm describes the degree of dissimilarity of two objects. The parameter p determines the influence of large difference values in some dimensions to the complete distance. For $p \to \infty$, the object distance is completely determined by the largest object difference in any dimension. Let us note that the exponent $\frac{1}{p}$ is used for normalization reasons only. Therefore, it is not required in algorithms that require a similarity ranking.

Given a specialized application context, the standard L_p-norms have several drawbacks:

1. Correlated features are based on the same characteristics of an object and thus, they implicitly increase the impact of this characteristics when calculating the dissimilarity.
2. Not each observation is equally important when deciding about object similarity. When, for instance, deciding between large and small people, the *height* parameter will be more significant than the *weight* parameter.

3. In order to have a large distance w.r.t. an L_p-norm, it is sufficient to have a considerably large difference in any single feature. Correspondingly, a small dissimilarity requires that both vectors display small difference values in each feature. On the other hand, to decide dissimilarity, any single feature is sufficient. This effect is a serious drawback because object similarity might not necessarily always depend on the same set of features. Having an extraordinarily large difference w.r.t. a single rather unimportant feature, could thus prevent two otherwise identical objects from being found in a similarity query. Thus, we argue that dissimilarity as well as similarity should be decided based on a combination of several features.

To solve the problems (1) and (2), the Euclidean distance has been extended to the Mahalanobis distance or quadratic form. The idea of this approach is to employ an affine transformation of the original feature space which is applied within the distance measure itself:

$$D_{\text{Mah}}(x_1, x_2) = \big((x_1 - x_2)^T \cdot A \cdot (x_1 - x_2)\big)^{\frac{1}{2}}$$

In order to make D_{Mah} a metric, the transformation matrix A has to be positive definite. In this case, A implies an affine transformation of the vector space B where the Euclidean distance is equivalent to D_{Mah} in the original space.

$$\begin{aligned}\big((x_1 - x_2)^T A (x_1 - x_2)\big)^{\frac{1}{2}} &= \big((x_1 - x_2)^T B^T B (x_1 - x_2)\big)^{\frac{1}{2}} \\ &= \big((Bx_1 - Bx_2)^T (Bx_1 - Bx_2)\big)^{\frac{1}{2}}\end{aligned}$$

When properly derived, this matrix A can achieve that the directions in the target space are uncorrelated. Additionally, the directions are weighted by their importance to the given application. There are multiple methods to learn a proper Mahalanobis distance like Fisherfaces [2], RCA [1], ITML [7] or LMNN [21] which are described in Section 5.

However, the Mahalanobis distance does not adequately solve the third problem named above because the feature values are only linearly scaled. Thus, all observed difference values are decreased by the same factor. Therefore, when decreasing a very large difference value to limit its too strong impact in a comparison, the impact of the feature is limited in all other comparisons as well. Thus, by preventing a too large impact in some distance calculations, we would generate too small distance values in others. To conclude, Mahalanobis distances are still equivalent to an Euclidean distance in a transformed data space and thus, these methods are no solution to the third problem mentioned above.

3 Ensembles of Bayes Estimates

In the following, we formally describe our method. We start with the definition of Bayes Estimates (BE) and Bayes Ensemble Distance (BED) on the original

feature dimensions. Afterwards, we introduce our solution to the problem of correlated features and provide a new way to derive an affine transformation of the feature space that allows the training of a meaningful BED.

3.1 Bayes Estimates and Bayes Ensemble Distance

As mentioned above, we want to learn a function having a pair of feature vectors as input and returning a similarity score as output. Similar to the L_p-norm, we describe the comparison between two feature vectors $x_1, x_2 \in \mathbb{R}^d$ by their difference vectors $(x_1 - x_2)$, or $(x_2 - x_1)$. Thus, our method assigns a similarity score to each difference vector. Since both difference vectors should provide the same dissimilarity score, we have to make sure that our similarity function is symmetric with respect to the direction of the input difference vector. As mentioned before, our approach treats each dimension of the input space separately. Thus, we define the Bayes Estimates (BE) for feature dimension i as simple Bayes classifier receiving a difference value $x_{1,i} - x_{2,i}$ as input. This classifier distinguishes object comparisons of similar objects (SIM) from comparisons of dissimilar objects (DIS). Thus, we learn two distribution functions over the difference values for similar objects and dissimilar objects. Additionally, we employ a prior distribution describing whether similarity is less likely than dissimilarity. As a result, we can calculate the conditional probability $P(\text{DIS} \mid x_{1,i} - x_{2,i})$ describing the dissimilarity likelihood for two objects under the condition of the observed difference value in dimension i. Correspondingly, $P(\text{SIM} \mid x_{1,i} - x_{2,i})$ expresses the likelihood that two objects are similar and can be used as similarity function. Formally, the Bayes Estimate (BE) for comparing two vectors $x_1, x_2 \in \mathbb{R}^d$ w.r.t. dimension i is defined as:

Definition 1 (Bayes Estimate). *Let $x_1, x_2 \in \mathbb{R}^d$ be two feature vectors. Let p_s and p_d represent a prior distribution describing the general likelihood that objects are considered to be similar. Then, the* Bayes Estimate (BE) *for x_1 and x_2 w.r.t. dimension i is defined as follows:*

$$\text{BE}_i(x_1, x_2) = \frac{p_d \cdot P((x_{1,i} - x_{2,i}) \mid \text{DIS})}{P_{total}(x_{1,i}, x_{2,i})} \ ,$$

where $p_{total}(x_{1,i}, x_{2,i})$ is the sum of the similarity and the dissimilarity probabilities ($p_s \cdot P((x_{1,i} - x_{2,i}) \mid \text{SIM})$ and $p_d \cdot P((x_{1,i} - x_{2,i}) \mid \text{DIS})$) in the i^{th} dimension.

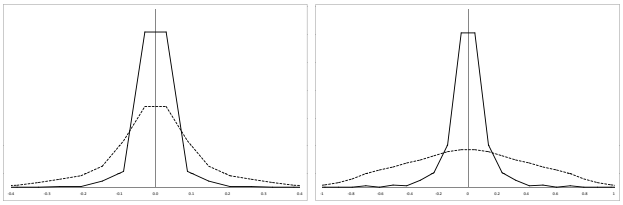
To combine these probabilities, we take the average estimates over all dimensions. Thus, we employ an Ensemble approach combining the descriptiveness of all available features. Let us note that this approach is different from building the joint probability for class DIS like in an ordinary naïve Bayes classifier. This approach would imply that in order to be similar, two objects have to be sufficiently similar in each dimension. Correspondingly, dissimilarity would require a sufficiently large difference value in all dimensions. Thus, the joint probability could again be determined by a single dimension. By building the average, our method underlies the more flexible understanding of similarity. Thus, neither a

very large difference nor a very small difference in a single dimension can imply similarity or dissimilarity on its own. Formally, we define the Bayes Ensemble Distance (BED) in the following way:

Definition 2 (Bayes Ensemble Distance). *Let $x_1, x_2 \in \mathbb{R}^d$ be two feature vectors. Let p_s and p_d represent a prior distribution describing the general likelihood that objects are considered to be similar. Then, the* Bayes Ensemble Distance (BED) *for x_1 and x_2 is defined as follows:*

$$\mathrm{BED}(x_1, x_2) = \frac{1}{d} \cdot \sum_{i=1}^{d} \mathrm{BE}_i(x_1, x_2)$$

From a data mining point of view, the BED is an Ensemble of d weak Bayesian learners, each deriving a probabilistic statement from the corresponding feature. Each learner distinguishes two classes, i.e. similarity and dissimilarity. Let us note that our method does not directly distinguish degrees of similarity. Instead, a quantitative view on object similarity is provided by the average probability that both objects are similar.



(a) Conf-hsv (10D) (b) Conf-hsv (29D)

Fig. 1. Difference distributions for similar (solid lines) and dissimilar (dashed lines) objects in a retrieval data set in dimension 10 and 29.

An open issue to the use of BED is the type of probability distribution being used to model the Bayes estimate. To select a well-suited probability density function, we examined several data sets with respect to their difference vector distribution for similar and dissimilar objects. Therefore, we built histograms on the observed difference values in each dimension. Remember that the all distributions have to be symmetric to the origin because of the pairwise appearance of positive and negative distance values. An example for the histograms derived from two image retrieval data sets is displayed in Fig. 1. In this and all other examined data sets, we observed a normal distribution for similar objects. Very similar or identical objects will usually display almost identical feature values. For the distributions describing dissimilarity, we sometimes observed distributions that also resemble a normal distribution but displayed a larger variance. In cases having well separated classes, the dissimilarity distribution often is split into two components, one for positive and one for negative difference values.

Thus, the dissimilarity resembled a mixture model having two symmetric components of equal weight where the first has a positive mean value and the second component has a negative mean value. In our experiments, we employed Gaussians as basis distribution. However, the general method is applicable for any other type of distribution function, e.g. exponential power distributions.

3.2 Training BEs

Training BEDs consists of determining the distribution parameters for each dimension, e.g. mean and variance for a Gaussian. Furthermore, it is often useful to determine prior probabilities for similarity and dissimilarity.

In the case that the examples are provided with class labels, it is easy to decide whether an object comparison is counted for the similar class (SIM) or for the dissimilarity class (DIS). If both objects belong to the same class, the observed difference value contributes to the SIM distribution. If both objects belong to different classes, the observed difference vector contributes to the distribution describing DIS. For small data sets, it is possible to consider all possible difference vectors occurring in the training set. However, this approach is not feasible for large data sets because the number of difference vectors is increasing with the squared number of training vectors. Thus, it is often advisable to select a subset of the difference vectors instead of employing all available samples. To find this subset, random sampling is applicable. In our experiments, we adapt the idea of target neighbors from [21] and select the difference vectors corresponding to the k-nearest neighbors of the same class and the k-nearest neighbors belonging to any other class for each training object. We employed the Euclidean distance to determine the target neighbors.

In case of labeled pairs, selecting examples is usually not an option because each object comparison has to be manually labeled and thus, it is rather unlikely that there will be too many examples for efficient training. However, labeling object pairs allows to distinguish several degrees of similarity $y \in [0..1]$, e.g. the label could indicate a similarity of 0.8 or 0.1. To employ these more detailed labels, we propose to proceed in a similar way as in EM clustering and let the training example contribute to both distributions. However, to consider the class labels, we weight the contribution to the similar distribution by y and the contribution to the dissimilar distribution by $1-y$. This way, undecidable comparisons having a label of 0.5 would equally contribute to both distribution functions, whereas a comparison having a label of 1.0 would exclusively contribute to the similar distribution.

In many applications, using a prior distribution can improve the accuracy of similarity search and object classification. Especially when using BED for nearest neighbor classification, we can assume that we know how many objects belong to the same class and how many objects belong to any other class. In these cases, we can determine the frequency $|c_i|$ of examples for each class $c_i \in C$ in the training set and easily derive the prior probability for similarity:

$$p_s = \frac{\sum_{c_i \in C} |c_i|^2}{(\sum_{c_i \in C} |c_i|)^2} \qquad p_d = 1 - p_s$$

In other words, we know that there are $|c_i|^2$ comparisons of similar objects within each class c_i. Dividing the amount of these comparisons by all possible comparisons computes the relative frequency of p_s. Since we only distinguish two cases, we can calculate p_d as $1 - p_s$.

In case of relevance feedback, directly determining the relative portion of similarity in the training objects is also easily possible. However, depending on the selection of the object pairs to be labeled it is often very unlikely that the label distribution is representative for the distribution on the complete database. Thus, it is often more useful to manually assign a value for the occurrence of each class.

4 Optimizing the Feature Space for BEs

Employing BED on the original dimensions ensures that neither similarity nor dissimilarity can be decided based on the difference value in a single dimension. Additionally, the importance of each dimension is indicated by the distinction of both distribution functions. However, correlated features still pose a problem for the performance of BED. First of all, the advantage of using an ensemble of learners strongly depends on their statistical independence. Additionally, it might occur that the single BEs in the original dimension might be very informative. However, there often exist dimensions in the data space allowing a good separation of the distribution function. An example is illustrated in Fig. 2. In the displayed case, the distributions of similar and dissimilar objects are modeled as multivariate Gaussians. If we consider the projection of both distributions onto the x-axis, we cannot decide between the two distributions at all. Projecting the Gaussians onto the main diagonal enables a clear separation. In this example, it can be assumed that the BE on the main diagonal has a much stronger predictive quality. To conclude, analogously to the Euclidean distance, BED can be improved by a linear transformation of the input space which decreases feature dependency and provides features allowing meaningful similarity estimation.

Formally, we want to find a set of base vectors $W = [w_1, \ldots, w_d]$ for transforming each original vector $x \in \mathbb{R}^{d'}$ into another d-dimensional feature space where each new dimension allows to build a better BE. Since we want to have independent learners, we additionally require that $w_i \perp w_j$ for $i \neq j$.

To determine the suitability of a dimension to train a useful BE, we need to find a criterion that is independent of the used type of distribution function. A certain dimension in the feature space is useful in the case that the distance values between similar objects are in average smaller than the distance values of dissimilar objects. Let us note that the mean value for both distributions has to be zero regardless of the underlying density function. Since distance values always occur in pairs of negative and positive values, the mean is always zero

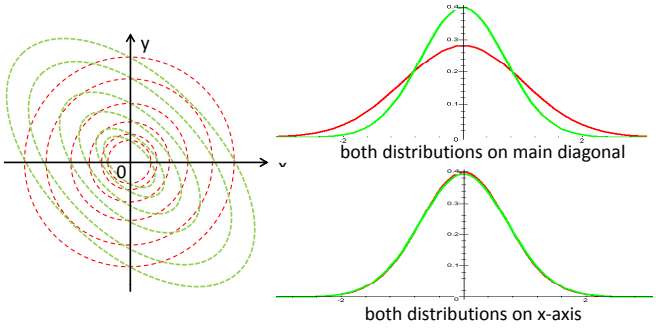


Fig. 2. Distributions of similar (green) and dissimilar (red) objects. Top view of multivariate Gaussians (left) and projections onto different dimensions (right).

in each dimension. Now, a direction is well-suited if the distance values being observed for similar objects are grouped closer to the origin than the values being observed for dissimilar objects. To quantify this intuition, we calculate the variance of the samples for both distributions SIM and DIS in dimension i and build the difference between both values:

$$q_i = \frac{1}{n} \cdot \left(\sum_{x_d \in \text{DIS}}^{n} \left(x_{d,i}^2 - 0\right) - \sum_{x_s \in \text{SIM}}^{n} \left(x_{s,i}^2 - 0\right) \right) = \frac{1}{n} \cdot \left(\sum_{x_d \in \text{DIS}}^{n} x_{d,i}^2 - \sum_{x_s \in \text{SIM}}^{n} x_{s,i}^2 \right)$$

If q_i is large, the difference values between similar objects are generally grouped more closely around zero than the difference values between dissimilar objects in dimension i. If q_i converges to zero, dimension i will usually not allow the training of a useful BE.

To describe the variance along all possible dimensions in the space of distance values, we can build the covariance matrix for similar and dissimilar difference vectors.

$$(\Sigma_{\text{SIM}})_{i,j} = \sum_{x_s \in \text{SIM}} (x_{s,i} - 0) \cdot (x_{s,j} - 0) = \sum_{x_s \in \text{SIM}} x_{s,i} \cdot x_{s,j}$$

Σ_{DIS} is built correspondingly on the difference vectors of dissimilar objects.

Our task is to find a set of orthogonal dimensions for which the difference between the variance of the dissimilar distribution and the variance of the similar distribution is as large as possible. Formally, we can define the following optimization problem:

$$\begin{aligned} &\text{Maximize } L(w_i) = w_i^T \Sigma_{\text{DIS}} w_i - w_i^T \Sigma_{\text{SIM}} w_i = w_i^T \cdot (\Sigma_{\text{DIS}} - \Sigma_{\text{SIM}})\, w_i \\ &\quad \text{s.t. } w_i \perp w_j \end{aligned}$$

The following eigenvalue equation solves this problem:

$$\lambda w = (\Sigma_{\text{DIS}} - \Sigma_{\text{SIM}}) \cdot w \ .$$

To integrate the learned affine transformation into the training of BED, we can either transform all feature vectors before training and testing by W or integrate the transformation directly into the BE distance by rotating each difference vector before it is processed. To conclude, the training of a BED is performed as follows:

1. Determine Σ_{SIM} and Σ_{DIS} from the labeled data.
2. Calculate W by solving the corresponding eigenvalue problem.
3. Rotate Σ_{SIM} and Σ_{DIS} by W.
4. Derive the variance values of the similarity and dissimilarity distributions for each $w_i \in W$.

Let us note that it is necessary to rotate the compared objects or their distance vector using W before calculating BED.

A final aspect of this space transformation is that it allows to reduce the number of considered dimensions. This can be done by selecting a fixed number of features and keep only the top k dimensions w.r.t. the quality q_i. Another alternative is to determine a threshold τ and keep only those dimensions offering a quality q_i which is better than τ.

5 Related Work

5.1 Metric Distance Learning

Most distance learning methods use the Mahalanobis distance, represented by a semi-definite matrix. The shared principle among all of those approaches is to ensure that the relations among a dataset's objects are transformed such that they best represent an underlying characteristic of the data.

In the following, we give a short summary of existing metric learning approaches. For detailed surveys, see [24, 23]. The main idea of unsupervised approaches is to reduce the feature space to a lower-dimensional space in order to eliminate noise and enable a more efficient object comparison. The criteria for selecting such a subspace are manifold. Principal Component Analysis (PCA) [10] builds an orthogonal basis aimed at best preserving the data's variance, Multidimensional Scaling (MDS) [6] seeks the transformation which best preserves the geodesic distances and Independent Component Analysis (ICA) [5] targets a subspace that guarantees maximal statistical independence. ISOMAP [19] by Tenenbaum et al. is a non-linear enhancement of the MDS principle, in identifying the geodesic manifold of the data and preserving its intrinsic geometry. Other unsupervised approaches (e.g. [16, 3]) try to fulfill the above criteria on a local scale.

Among supervised approaches, the first to be named is Fisher's Linear Discriminant (FLD) [8]. It maximizes the ratio of the between-class variance and the

within-class variance using a generalized eigenvalue decomposition. This method has been extended by Belhumeur et al. [2] to the Fisherfaces approach. It precedes FLD with a reduction of the input space to its principal components and can thus filter unreliable input dimensions. BED and especially the target function L share several important ideas with Fisherfaces. However, FLD assumes that the data is partitioned into classes which are modeled using the Gaussian distribution function, whereas BED does not require explicit object classes. Furthermore, the BED is not determined to the use of Gaussian functions. Instead BEDs employ the difference vectors and always try to distinguish the two basic statements of object similarity and object dissimilarity which can be modeled by an arbitrary symmetric density function. Both methods generate covariance matrices of difference vectors representing similarity (in FLD: the within-class scatter matrix) and dissimilarity (in FLD: the between-class scatter matrix). However, in FLD the matrices are built based on the difference vectors w.r.t. a mean value whereas BED directly employs object-to-object comparisons. Where FLD tries to find dimensions where the ratio between the variances of dissimilarity and similarity are as large as possible, BED maximizes the difference between the variances of the dissimilarity and the similarity distributions.

With RCA [1], Bar-Hillel et al. focus on the problem of minimizing within-*chunklet* variance. They argue that between-class differences are less informative than within-class differences and that class assignments frequently occur in such a way that only pairs of equally-labelled objects can be extracted. These pairs are extended into chunklets (sets) of equivalent objects. The inverse chunklet covariance matrix is used for calculating the Mahalanobis distance. This step should usually be preceded by dimensionality reduction. The main difference between BED and RCA is that RCA does not build a distribution function for object comparison corresponding to dissimilarity. Correspondingly, RCA only requires examples for comparison between the objects of the same class. As a result, the optimization which is provided by RCA is not aimed at distinguishing both classes of difference vectors. Instead, RCA is mostly based on a whitening transformation of the matrix which is similar to the within-class-scatter-matrix of FLD.

NCA [9] proposed by Goldberger et al. optimizes an objective function based on a soft neighborhood assignment evaluated via the leave-one-out error. This setting makes it more resistant against multi-modal distributions. The result of this optimization is a Mahalanobis distance directly aimed at improving nearest-neighbor classification. The objective function is, however, not guaranteed to be convex.

With Information-Theoretic Metric Learning (ITML) [7], Davis et al. propose a low-rank kernel learning problem which generates a Mahalanobis matrix subject to an upper bound for inner-class distances and a lower bound to between-class distances. They regularize by choosing the matrix closest to the identity matrix and introduce a way to reduce the rank of the learning problem.

LMNN (Large Margin Nearest Neighbor) [21] by Weinberger et al. is based on a semi-definite program for directly learning a Mahalanobis matrix. They

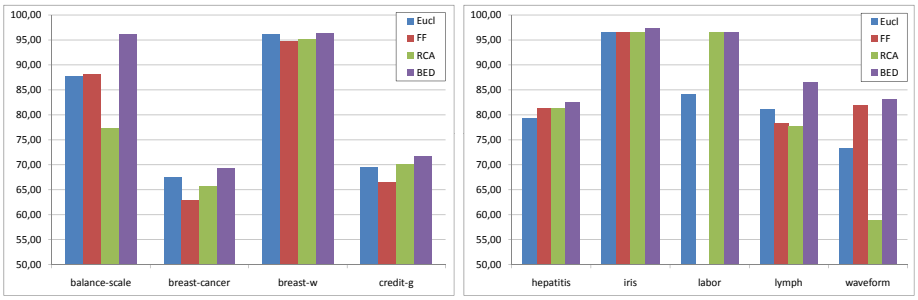
require *k-target neighbors* for each input object x, specifying a list of objects, usually of the same class as x, which should always be mapped closer to x than any object of another class. These k-target neighbors are the within-class k-nearest neighbors. Hence, the loss function consists of two terms for all data points x: the first penalizes the distance of x to its k-target neighbors and the second penalizes close objects being closer to x than any of its target neighbors. In [22], they propose several extensions, involving a more flexible handling of the k-target neighbors, a multiple-metric variant, a kernelized version for datasets of larger dimension than size and they deal with efficiency issues arising from the repeated computation of close objects. Nonetheless, LMNN requires a specialized solver in order to be run on larger data sets.

5.2 Non-Metric Distance Learning

In order to be metric, a distance has to fulfill the metric axioms (i.e. self-similarity, symmetry, triangle inequality). In fact, several recent studies have shown that these axioms (triangle inequality above all) are often not conform with the perceptual distance of human beings [17, 20] and thus not suitable for robust pattern recognition [12]. Most of the approaches learning a non-metric function as distance function only use fragments of the objects for the similarity calculation between them (e.g. [18, 12]). This can be useful for image retrieval and classification, where only small parts (not a subset of features) of two images can yield to perception of similarity, but is not applicable for object representations in general. Another class of non-metrical distance learners are Bayesian Learners as used in [13], which are also designed for the special case of object recognition in images. In this work, we do not want to restrict similarity to images, but rather present a more general view applicable for a broad range of applications.

6 Experimental Evaluation

In this section, we present the results of our experimental evaluation. As comparison partner we selected the methods that are closest to our approach: Relevant Component Analysis (RCA) and Fisher Faces (FF). Let us note that RCA requires only chunks of data objects having the same class and no explicit class set. However, since we used datasets having class labels, we provided RCA the complete set of training objects for each class as a chunk. Furthermore, we compared Bayes Estimate Distance (BED) to the standard Euclidean distance to have baseline method. We evaluated all methods on several real-world datasets to test their performance for classification and retrieval tasks. All methods were implemented in Java 1.6 and tests were run on a dual core (3.0 Ghz) workstation with 2 GB main memory.



(a) Results on 4 UCI datasets (b) Results on 5 UCI datasets

Fig. 3. NN-Classification results on several UCI datasets.

6.1 Nearest Neighbor Classification

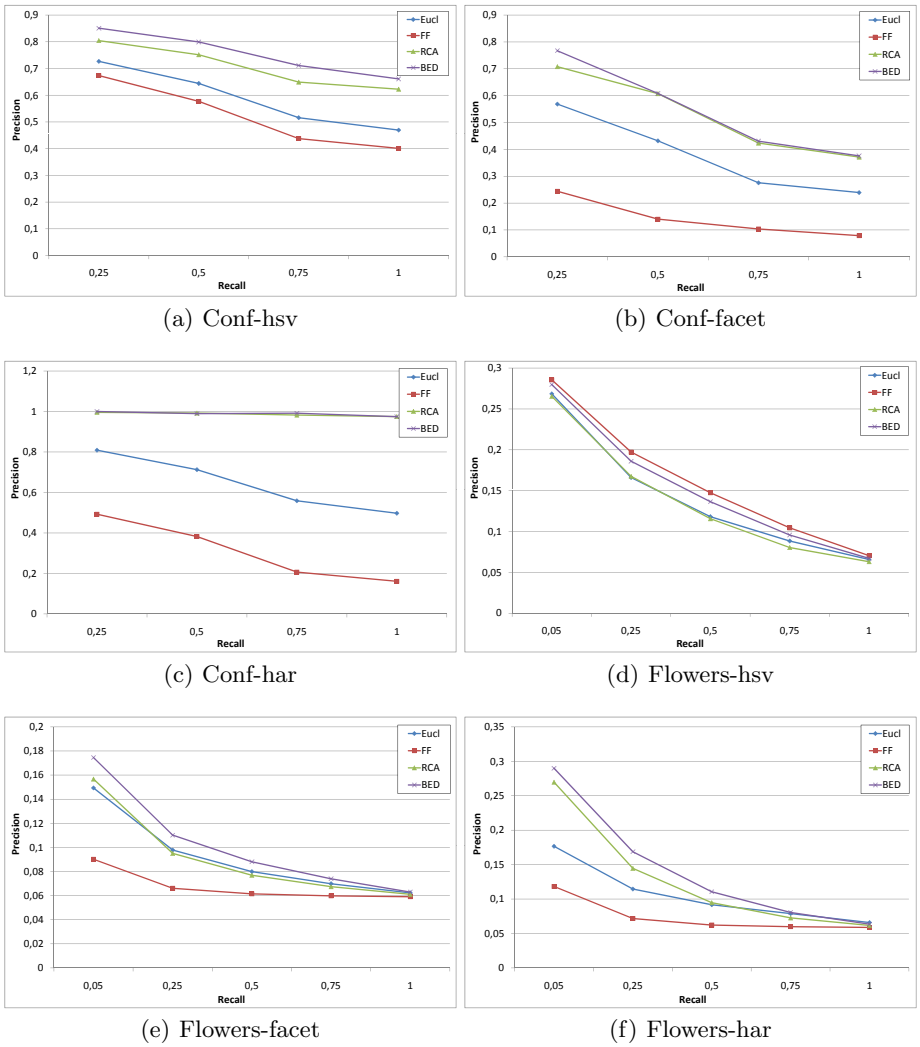
As mentioned before, our similarity learner can be applied for different applications. A first, well-established method is improving the quality of nearest neighbor classification. For the classification task, we used several datasets from the UCI Machine Learning Repository [14]. Evaluation on the datasets was performed using 10-fold cross-validation and all 4 distance measures were used for basic nearest neighbor classification. To train BED, we employed sampling based on the target neighbors. In other words, we took the difference vectors of all training objects to the k-nearest neighbors within the same class and the k-nearest neighbors in all other classes. To find out a suitable value for k, we screened over a small set of suitable values between 5 and 20.

Table 1. Image Retrieval Data Sets

Dataset	Instances	Attributes	Classes
Conf-hsv	183	32	35
Conf-facet	183	24	35
Conf-har	183	65	35
Flowers-hsv	1360	32	17
Flowers-facet	1360	24	17
Flowers-har	1360	65	17

The results for NN classification are shown in Fig. 3. BED displays the largest accuracy in all 9 datasets. Though RCA achieves the same result on the labor dataset, it impairs the Euclidean distance on data sets like waveform or balance scale. The same observation can be made for FF. Though the accuracy is comparably good in all data sets, there also exist datasets where FF does not yield an advantage even against Euclidean distance (breast-w). On the labor dataset, it was not possible to learn a distance using FF due to a matrix singularity. To conclude, BED leads to an up to 8% better classification of objects on the tested

datasets compared to the best of Eucl, FF and RCA. Thus, we can state that BED can be employed to improve the results of instance-based learners.



(a) Conf-hsv (b) Conf-facet

(c) Conf-har (d) Flowers-hsv

(e) Flowers-facet (f) Flowers-har

Fig. 4. Precision-Recall Graphs on the Conf and Flowers dataset

6.2 Precision and Recall Graphs

We employed two image datasets for testing the performance of our new distance measures for retrieval applications. The *Conf* dataset was created by ourselves

and contains 183 images of 35 different motives. The *Flowers* dataset was introduced in [15] and consists of 1360 images of 17 different types of flowers. From these two datasets, we extracted color histograms (based on the HSV color space), facet features [4] and haralick features [11]. The characteristics of the resulting feature datasets can be seen in Table 1. On these datasets, we measured the retrieval performance using precision-recall-graphs. We posed a ranking query for each image and measured the precision of the answer resulting from the remaining database for several levels of recall. In the retrieval task, we employed very large numbers of difference vectors for training, to adjust BED to achieving reasonable precision values for large levels of recall.

On the *Conf* dataset, BED shows an impressive boost of the retrieval quality using hsv-color-histograms (Fig. 4(a)), while it still leads to slightly better results using facet or Haralick features (see Figures 4(b) and 4(c)) in contrast to RCA. FF does not appear to be well-suited for these datasets, as it performs even worse than the Euclidean Distance. On the Flowers dataset, retrieval quality can again be improved by BED when using Facet and Haralick features respectively (see Figures 4(e) and 4(f)). On the feature dataset consisting of the hsv-color-histograms of *Flowers*, Fisherfaces lead to a better Precision-Recall-Graph (Fig. 4(d)) than the other approaches, but note that this is the only retrieval experiment where FF performed better than the Euclidean distance. Thus, we can state the BED is suitable for retrieval tasks as well as for data mining tasks.

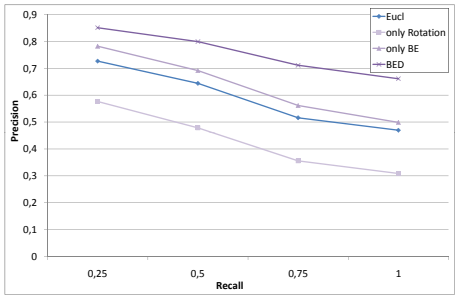


Fig. 5. Different versions of BE on Conf-hsv

6.3 Comparison to its Components

In our last experiment, we examine the performance of BEDs compared to their separated components. We trained BEs on the original dimensions (only BE) of the feature space. Furthermore, we wanted to find out whether the learned eigenvalue decomposition can be used for learning a Mahalanobis distance improving classification results. To create such a transformation, we additionally multiplied each eigenvector w by its inverse eigenvalue. The comparison was

performed for several retrieval datasets which all displayed similar results. An example precision-recall graph of the *Conf-hsv* data set is presented in Fig. 5. Using the BED without the rotation still increases the retrieval performance compared to the plain Euclidean distance on the same feature space. Thus, even without an affine transformation, the BED is capable of improving the retrieval quality. A second very interesting result is that the rotation component of BEDs does not yield any performance advantage when used as Mahalanobis learner. Though the learned directions do optimize the BEs being observed in the new dimensions, they seem to be unsuitable for improving the results obtained by the Euclidean distance.

7 Conclusion

In this paper, we have introduced Bayes Ensemble Distance (BED) as new adaptable dissimilarity measure. BED is applied to the difference vector of two feature vectors. For each dimension, BED independently determines the likelihood that both objects are dissimilar employing a simple Bayesian learner called Bayes Estimate (BE). The results of the BEs are combined by computing the average prediction. Thus, the derived similarity score is less dependent on outlier values in some of the dimensions. Since BED is dependent on the spatial rotation of the data space, it is possible to optimize the vector space in order to derive a feature space allowing the training of more descriptive and independent BEs. In our experimental evaluation, we have demonstrated that BEDs can largely increase the classification accuracy of instance-based learning. Additionally, we have demonstrated the suitability of BED for retrieval tasks. For future work, we plan to investigate efficiency issues when using BED for information retrieval. Furthermore, we plan to apply the idea of BEs to structured objects like graphs.

Acknowledgements

This research has been supported in part by the THESEUS program in the MEDICO and CTC projects. They are funded by the German Federal Ministry of Economics and Technology under the grant number 01MQ07020. The responsibility for this publication lies with the authors.

References

1. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning distance functions using equivalence relations. In: Proceedings of the 20th International Conference on Machine Learning (ICML), Washington, DC, USA. (2003) 11–18
2. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(7) (1997) 711–720

3. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation **15**(6) (2003) 1373–1396
4. Chinga, G., Gregersen, O., Dougherty, B.: Paper surface characterisation by laser profilometry and image analysis. Journal of Microscopy and Analysis **84** (2003) 5–7
5. Comon, P.: Independent component analysis, a new concept? Signal Processing **36**(3) (1994) 287–314
6. Cox, T.F., Cox, M.A.A.: Multidimensional Scaling. 2nd edn. Chapman & Hall/CRC (2001)
7. Davis, J., Kulis, B., Sra, S., Dhillon, I.: Information-theoretic metric learning. In: in NIPS 2006 Workshop on Learning to Compare Examples. (2007)
8. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics **7** (1936) 179–188
9. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighborhood component analysis. In: Advances in Neural Information Processing Systems, MIT Press (2004) 513–520
10. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2001)
11. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Transactions on Speech and Audio Processing **3**(6) (1973) 6103–623
12. Jacobs, D.W., Weinshall, D., Gdalyahu, Y.: Classification with non-metric distances: Image retrieval and class representation. IEEE Trans. Pattern Analysis and Machine Intelligence **22**(6) (2000) 583–600
13. Moghaddam, B., Pentland, A.: Probabilistic visual learning for object representation. IEEE Trans. Pattern Analysis and Machine Intelligence **19**(7) (1997) 696–710
14. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI repository of machine learning databases (1998)
15. Nilsback, M.E., Zisserman, A.: A visual vocabulary for flower classification. CVPR **2** (2006) 1447–1454
16. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500) (2000) 2323–2326
17. Santini, S., Jain, R.: Similarity measures. IEEE Trans. Pattern Analysis and Machine Intelligence **21** (1999) 871–883
18. Tan, X., Chen, S., Zhou, Z.H., Liu, J.: Learning non-metric partial similarity based on maximal margin criterion. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (2006) 138–145
19. Tenenbaum, J.B., Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500) (2000) 2319–2323
20. Tversky, A.: Features of similarity. Psychological Review **84**(4) (1977) 327–352
21. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. In: Advances in Neural Information Processing Systems, Cambridge, MA, USA, MIT Press (2006)
22. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research **10** (2009) 207–244
23. Yang, L.: An overview of distance metric learning. Technical report, Department of Computer Science and Engineering, Michigan State University (2007)
24. Yang, L., Jin, R.: Distance metric learning: A comprehensive survey. Technical report, Department of Computer Science and Engineering, Michigan State University (2006)