Journal of Machine Learning Research 13 (2012) 2871-2902 Submitted 8/11; Revised 6/12; Published 10/12

Local and Global Scaling Reduce Hubs in Space

Dominik Schnitzer DOMINIK.SCHNITZER@OFAI.AT
Arthur Flexer ARTHUR.FLEXER@OFAI.AT
Austrian Research Institute for Artificial Intelligence (OFAI)
Freyung 6/6, 1010 Vienna, Austria

Markus Schedl MARKUS.SCHEDL@JKU.AT
Gerhard Widmer GERHARD.WIDMER@JKU.AT
Department of Computational Perception
Johannes Kepler University
Altenbergerstraße 69, 4040 Linz, Austria

Editor: Gert Lanckriet

Abstract

'Hubness' has recently been identified as a general problem of high dimensional data spaces, manifesting itself in the emergence of objects, so-called hubs, which tend to be among the k nearest neighbors of a large number of data items. As a consequence many nearest neighbor relations in the distance space are asymmetric, that is, object y is amongst the nearest neighbors of x but not vice versa. The work presented here discusses two classes of methods that try to symmetrize nearest neighbor relations and investigates to what extent they can mitigate the negative effects of hubs. We evaluate local distance scaling and propose a global variant which has the advantage of being easy to approximate for large data sets and of having a probabilistic interpretation. Both local and global approaches are shown to be effective especially for high-dimensional data sets, which are affected by high hubness. Both methods lead to a strong decrease of hubness in these data sets, while at the same time improving properties like classification accuracy. We evaluate the methods on a large number of public machine learning data sets and synthetic data. Finally we present a real-world application where we are able to achieve significantly higher retrieval quality.

Keywords: local and global scaling, shared near neighbors, hubness, classification, curse of dimensionality, nearest neighbor relation

1. Introduction

In a recent publication in this journal, Radovanović et al. (2010) describe the so-called 'hubness' phenomenon and explore it as a general problem of machine learning in high-dimensional data spaces. Hubs are data points which keep appearing unwontedly often as nearest neighbors of a large number of other data points. This effect is particularly problematic in algorithms for similarity search (for example, similarity-based recommenders), as the same similar objects are found over and over again and other objects are never recommended. The effect has been shown to be a natural consequence of high dimensionality and as such is yet another aspect of the curse of dimensionality (Bellman, 1961).

A direct consequence of the presence of hubs is that a large number of nearest neighbor relations in the distance space are asymmetric, that is, object y is amongst the nearest neighbors of x but not vice versa. A hub is by definition the nearest neighbor of a large number of objects, but

these objects cannot possibly all be the nearest neighbor of the hub. This observation connects the hub problem to methods that attempt to symmetrize nearest neighbor relations, such as 'shared near neighbors' (Jarvis and Patrick, 1973) and 'local scaling' (Zelnik-Manor and Perona, 2005). While these methods require knowledge of the local neighborhood of every data point, we propose a global variant that combines the idea of 'shared near neighbor' approaches with a transformation of distances to nearest neighbor 'probabilities' to define a concept we call *Mutual Proximity*. The approach is unsupervised and transforms an arbitrary distance function to a probabilistic similarity (distance) measure. Contrary to the local variants, this new approach lends itself to fast approximation for very large data bases and enables easy combination of multiple distance spaces due to its probabilistic nature.

In experiments with a large number of public machine learning databases we show that both local and global scaling methods lead to: (i) a significant decrease of hubness, (ii) an increase of k-nearest neighbor classification accuracy, and (iii) a strengthening of the pairwise class stability of the nearest neighbors. To demonstrate the practical relevance, we apply our global scaling algorithm to a real-world music recommendation system and show that doing so significantly improves the retrieval quality.

To permit other researchers to reproduce the results of this paper, all databases and the main evaluation scripts used in this work have been made publicly available.[1]

2. Related Work

The starting point for our investigations is a field where the existence of hubs has been well documented and established, namely, Music Information Retrieval (MIR). One of the central notions in MIR is that of music similarity. Proper modeling of music similarity is at the heart of many applications involving the automatic organization and processing of music data bases. In Aucouturier and Pachet (2004), hub songs were defined as songs which are, according to an audio similarity function, similar to very many other songs and therefore keep appearing unwontedly often in recommendation lists, preventing other songs from being recommended at all. Such songs that do not appear in any recommendation list have been termed 'orphans'. Similar observations about false positives in music recommendation that are not perceptually meaningful have been made elsewhere (Pampalk et al., 2003; Flexer et al., 2010; Karydis et al., 2010). The existence of the hub problem has also been reported for music recommendation based on collaborative filtering instead of audio content analysis (Celma, 2008). Similar effects have been observed in image (Doddington et al., 1998; Hicklin et al., 2005) and text retrieval (Radovanović et al., 2010), making this phenomenon a general problem in multimedia retrieval and recommendation.

In the MIR literature, Berenzweig (2007) first suspected a connection between the hub problem and the high dimensionality of the feature space. The hub problem was seen as a direct result of the curse of dimensionality (Bellman, 1961), a term that refers to a number of challenges related to the high dimensionality of data spaces. Radovanović et al. (2010) were able to provide more insight by linking the hub problem to the property of *concentration* (François et al., 2007) which occurs as a natural consequence of high dimensionality. Concentration is the surprising characteristic of all points in a high dimensional space to be at almost the same distance to all other points in that space. It is usually measured as a ratio between some measure of spread and magnitude. For example, the ratio between the standard deviation of all distances to an arbitrary reference point and the mean of

1. Databases and scripts can be found at `http://www.ofai.at/~dominik.schnitzer/mp`.

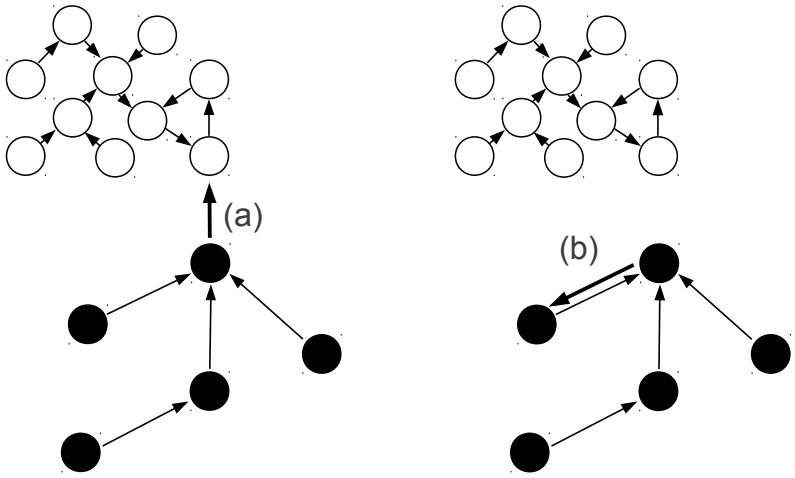
these distances. If this ratio converges to zero as the dimensionality goes to infinity, the distances are said to concentrate. For example, in the case of the Euclidean distance and growing dimensionality, the standard deviation of distances converges to a constant while the mean keeps growing. Thus the ratio converges to zero and the distances are said to concentrate.

The effect of distance concentration has been studied for Euclidean spaces and other ℓ^p norms (Aggarwal et al., 2001; François et al., 2007). Radovanović et al. (2010) presented the argument that in the finite case, due to this phenomenon some points are expected to be closer to the data-set mean than other points and are at the same time closer, on average, to all other points. Such points closer to the data-set mean have a high probability of being hubs, that is, of appearing in nearest neighbor lists of many other points. Points which are further away from that mean have a high probability of being 'orphans', that is, never appearing in any nearest neighbor list.

Nearest neighbor search is an essential method in many areas of computer science, such as pattern recognition, multimedia search, vector compression, computational statistics and data mining (Shakhnarovich et al., 2006) and, of course, information retrieval and recommendation. It is a well defined task: given an object x, find the most similar object in a collection of related objects. In the case of recommendation, the k most similar objects are retrieved with $k << n$ (n being the number of all objects in the data base). Since hubs appear in very many nearest neighbor lists, they tend to render many nearest neighbor relations asymmetric, that is, a hub y is the nearest neighbor of x, but the nearest neighbor of the hub y is another point a ($a \neq x$). This is because hubs are nearest neighbors to very many data points but only k data points can be nearest neighbors to a hub since the size of a nearest neighbor list is fixed. This behavior is especially problematic in classification or clustering if x and y belong to the same class but a does not, violating what Bennett et al. (1999) called the *pairwise stability* of clusters. Radovanović et al. (2010) coined the term *bad hubs* for points that show a disagreement of class information for the majority of data points they are nearest neighbors to. Figure 1 illustrates the effect: although a is, in terms of the distance measure, the correct answer to the nearest neighbor query for y, it may be beneficial to use a distance measure that enforces symmetric nearest neighbors. Thus a small distance between two objects should be returned only if their nearest neighbors concur.

This links the hub problem to 'shared near neighbor' (SNN) approaches, which try to symmetrize nearest neighbor relations. The first work to use common near neighbor information dates back to the 1970s. Jarvis and Patrick (1973) proposed a 'shared near neighbor' similarity measure to improve the clustering of 'non-globular' clusters. As the name suggests, the shared near neighbor (SNN) similarity is based on computing the overlap between the k nearest neighbors of two objects. Shared near neighbor similarity was also used by Ertöz et al. (2003) to find the most representative items in a set of objects. Pohle et al. (2006) define a related similarity measure based on the rank of nearest neighbors. They call their method 'proximity verification' and use it to enhance audio similarity search. Jin et al. (2006) use the reverse nearest neighbor (RNN) relation to define a general measure for outlier detection.

Related to SNN approaches are local scaling methods, which use local neighborhood information to rescale distances between data points. The intention is to find specific scaling parameters for each point, to be used to tune the pairwise distances in order to account for different local densities (scales) of the neighborhoods. Local scaling in this sense was first introduced as part of a spectral clustering method by Zelnik-Manor and Perona (2005). It transforms arbitrary distances using the distance between object x and its k'th nearest neighbor (see Section 3.1 below). In the context of image retrieval, Jegou et al. (2010) describe a related method called 'contextual dissimilarity measure'



(a) Original nearest neighbor relations

(b) Desired nearest neighbor relations

Figure 1: Schematic plot of two classes (black/white filled circles). Each circle has its nearest neighbor marked with an arrow: (a) violates the *pairwise stability* clustering assumption, (b) fulfills the assumption. In many classification and retrieval scenarios, (b) would be the desired nearest neighbor relation for the data set.

(CDM) and show that it reduces the error rates of the retrieval algorithm significantly, observing that "*the neighborhood symmetry rate increases*", while at the same time "*the percentage of never seen images decreases*", and in addition that "*the most frequent image is returned 54 times in the first 10 positions with the CDM, against 1062 times using the standard L1 distance*". While they do not explicitly make reference to the notion of hubs, their observations indicate the potential of local distance scaling to mitigate hub-related problems.

3. Scaling Methods

In the previous section we have seen that (i) the emergence of hubs leads to asymmetric nearest neighbor relations and that (ii) literature already contains hints that local scaling methods seem to improve the situation. However a detailed analysis of these facts and a systematic connection to the investigations of Radovanović et al. (2010) has not yet been done.

In what follows we review the local scaling methods and introduce a new global variant, which is also very simple to use. Due to its probabilistic modeling it possesses certain advantages over the local variant. Both methods are evaluated in regard to their effects on hubness in Section 4.

All the methods described here assume an underlying distance (divergence) measure with the following properties:

Definition 1 *Given a non-empty set M with n objects, each element $m_x \in M$ is assigned an index $x = 1 \ldots n$. We define a divergence measure $d : M \times M \to \mathbb{R}$ satisfying the condition of non-negativity in its distances:*

- *non-negativity:* $d(m_x, m_y) \geq 0, \quad m_x, m_y \in M,$

Individual objects $m_x \in M$ are referenced in the text by their index x. The distance between two objects x and y is denoted as $d_{x,y}$.

3.1 Local Scaling

Local scaling (Zelnik-Manor and Perona, 2005) transforms arbitrary distances to so-called *affinities* (that is, similarities) according to:

$$LS(d_{x,y}) = \exp\left(-\frac{{d_{x,y}}^2}{\sigma_x \sigma_y}\right), \tag{1}$$

where σ_x denotes the distance between object x and its k'th nearest neighbor. $LS(d_{x,y})$ tends to make neighborhood relations more symmetric by including local distance statistics of both data points x and y in the scaling. The exponent in Equation 1 can be rewritten as ${d_{x,y}}^2/\sigma_x\sigma_y = (d_{x,y}/\sigma_x)(d_{x,y}/\sigma_y)$: only when both parts in this product are small will the locally scaled similarity $LS(d_{x,y})$ be high. That is, x and y will be considered close neighbors only if the distance $d_{x,y}$ is small relative to both local scales σ_x and σ_y. Jegou et al. (2007) introduce a closely related variant called non-iterative contextual dissimilarity measure (NICDM). Instead of using the distance to the k'th nearest neighbor to rescale the distances, the average distance of the k nearest neighbors is used. This should return more stable scaling numbers and will therefore be used in all our evaluations. The non-iterative contextual dissimilarity measure (NICDM) transforms distances according to:

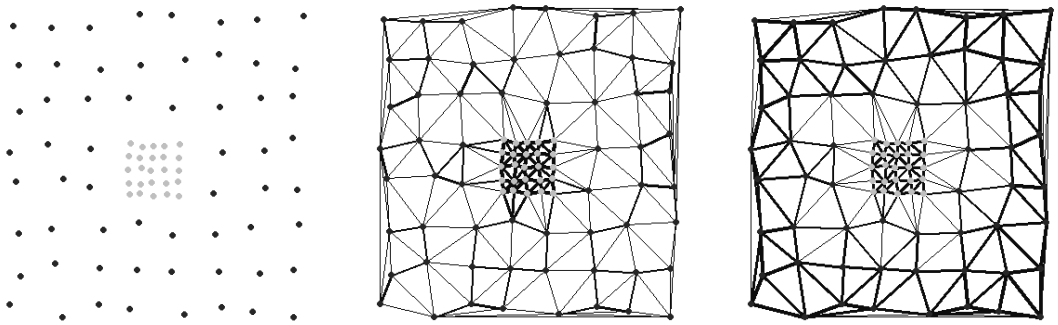
$$NICDM(d_{x,y}) = \frac{d_{x,y}}{\sqrt{\mu_x \mu_y}},$$

where μ_x denotes the average distance to the k nearest neighbors of object x. The iterative version of this algorithm performs the same transformation multiple times until a stopping criterion is met. Since these iterations yield only very minor improvements at the cost of increased computation time, we used the non-iterative version in our evaluations.

3.2 Global Scaling - Mutual Proximity

In this section we introduce a global scaling method that is based on: (i) transforming a distance between points x and y into something that can be interpreted as the probability that y is the closest neighbor to x given the distribution of the distances of all points to x in the data base; and (ii) combining these probabilistic distances from x to y and y to x via their joint probability. The result is a general unsupervised method to transform arbitrary distance matrices to matrices of probabilistic *mutual proximity* (MP). In contrast to local scaling methods, which use the local neighborhood information, MP uses information about all objects—thus the term global scaling.

The general idea of MP is to reinterpret the original distance space so that two objects sharing similar nearest neighbors are more closely tied to each other, while two objects with dissimilar neighborhoods are repelled from each other. This is done by reinterpreting the distance of two objects as a mutual proximity in terms of their distribution of distances.



(a) Original space: A tight cluster (gray dots in the center) placed in a loose background cluster (black dots).

(b) The affinity of two points in the original space is indicated by the thickness of the line connecting two points.

(c) Affinities after applying Mutual Proximity to the distances.

Figure 2: The effect of scaling techniques. Objects with similar nearest neighbors are tied together closely, while objects with dissimilar neighbors are repelled.

Figure 2 illustrates the effect of this reinterpretation in an example. The effect which can be seen here is similar to the intuitive repair of nearest neighbor relation as it was discussed in the beginning in Section 2 (Figure 1).

Figure 2a plots points from two classes on a two dimensional plane. A tight cluster (the gray dots in the center) is placed in a loose background cluster (black dots). Figure 2b connects close neighbors with lines according to a Delaunay triangulation.[2] The thickness of the lines shows the affinity of two neighboring points according to their Euclidean distance. The third plot (Figure 2c) plots the affinities after applying MP. It can be clearly seen that points from the loose cluster as well as points from the tight cluster now both have a high intra-class affinity. However, at the cluster borders there is weak affinity and strong separation as points from the tight cluster have different nearest neighbors than points from the background cluster.

This visible increase in class separation can also be measured in terms of classification rates. Simple two class k-nearest neighbor classification (tight vs. loose cluster) with this artificially generated data yields the following results: In the original space 96.4% of the nearest neighbors (at $k = 1$) are classified correctly; after applying MP, all (100%) of the nearest neighbors can be classified correctly. For $k = 5$ the classification rate increases from 95.2% to 98.8%.

3.2.1 COMPUTING MUTUAL PROXIMITY (MP)

To compute MP, we assume that the distances $d_{x,i=1..n}$ from an object x to all other objects in our collection follow a certain probability distribution. For example, Casey et al. (2008) and Cai et al. (2007) show that the ℓ^p distances they compute follow a Gamma distribution. Ferencz et al. (2005) used the Gamma distribution to model ℓ^2 distances from image regions. Pękalska and Duin (2000) show in general that based on the central limit theorem and if the feature vectors are independent and

2. A Delaunay triangulation ensures that the circumcircle associated with each triangle contains no other point in its interior, that is, no lines cross. This restriction is helpful for visualization purposes.

identically distributed (i.i.d.), their ℓ^2 distances approximately follow a normal distribution. As real data is not i.i.d., this can not be assumed. We, however, note that the accuracy of this approximation increases with increasing intrinsic dimensionality (François et al., 2007).

Under the assumption that all distances in a data set follow a certain distribution, any distance $d_{x,y}$ can now be reinterpreted as the probability of y being the nearest neighbor of x, given their distance $d_{x,y}$ and the probability distribution $P(X)$. $P(X)$ is defined by the distances of x to all other objects in the collection. In fact the probability that a randomly drawn element z will have a distance $d_{x,z} > d_{x,y}$ can then be computed:

$$P(X > d_{x,y}) = 1 - P(X \leq d_{x,y}) = 1 - \mathcal{F}_x(d_{x,y}).$$

$\mathcal{F}_x$ denotes the cumulative distribution function (cdf) which is assumed for the distribution of distances $d_{x,i=1..n}$. This way the probability of an element being a nearest neighbor of x increases with decreasing distance.

For illustration purposes we assume that in our collection the distances are normally distributed. Figure 3a shows a schematic plot of the probability density function (pdf) that was estimated for the distances of some object x. The mean distance or expected distance from x ($\hat{\mu}_x$) is in the center of the density function. Objects with a small distance to x (that is, objects with high similarity in the original space) find their distance towards the left of the density function. Note that the leftmost possible distance in this sketch is $d_{x,x} = 0$.[3] Figure 3b plots the probability of y being the nearest neighbor of x given $d_{x,y}$ (the gray filled area). The probability increases the smaller the distance to x is, or the farther left its distance is on the x-axis of the pdf.

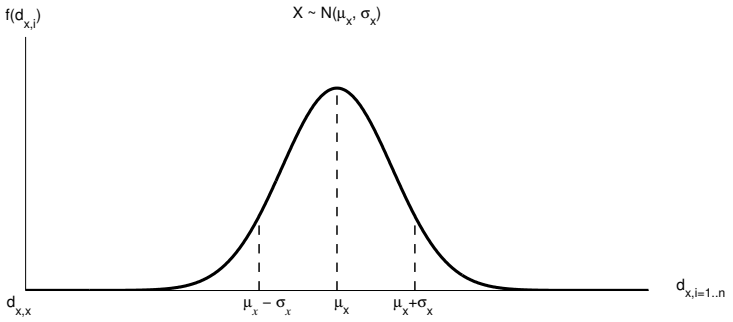
Note that this reinterpretation naturally leads to asymmetric probabilities for a given distance, as the distance distribution estimated for x may be different from the one estimated for y; x might be the nearest neighbor of y but not vice versa. Contrary to the original distances the probabilities now encode this asymmetric information. This allows for a convenient way to combine the asymmetric probabilities into a single expression, expressing the probability of x being a nearest neighbor of y and vice versa.

Definition 2 *We compute the probability that y is the nearest neighbor of x given $P(X)$ (the pdf defined by the distances $d_{x,i=1...n}$) and x is the nearest neighbor of y given $P(Y)$ (the pdf defined by the distances $d_{y,i=1...n}$), with their joint distribution $P(X,Y)$. The resulting probability is called* ***Mutual Proximity (MP)****:*

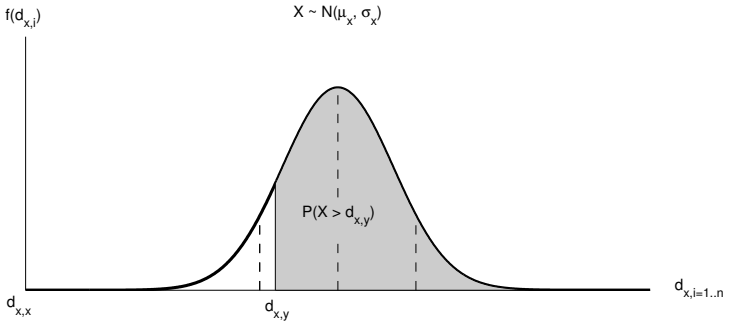
$$\begin{aligned} MP(d_{x,y}) &= P(X > d_{x,y} \cap Y > d_{y,x}) \\ &= 1 - P(X \leq d_{x,y} \cup Y \leq d_{x,y}) \\ &= 1 - \left[P(X \leq d_{x,y}) + P(Y \leq d_{y,x}) - P(X \leq d_{x,y} \cap Y \leq d_{y,x})\right]. \end{aligned}$$

Figure 4 illustrates MP for the distance $d_{x,y}$ and the joint distance distribution of X and Y, $P(X,Y)$. Each point of the plot refers to an object in the collection and its distance to points x and y. The shaded area (II) then defines the probability which is computed by MP. Sectors I+III correspond

3. Strictly speaking, then, the interpretation of this as a normal distribution is incorrect, since distances < 0 are not possible. However, we find the interpretation useful as a metaphor that helps understand why it makes sense to combine different views. We will do so in this section.



(a) The closer other elements are to x, the more to the left is their distance located on the x-axis of the density function plot. The leftmost possible observation in the data is the distance $d_{x,x} = 0$.



(b) The shaded area shows the probability that y is the nearest neighbor of x based on the distance $d_{x,y}$ and X. The closer y is to x (the smaller $d_{x,y}$) the higher the probability.

Figure 3: Schematic plot of the probability density function of a normal distribution which was estimated from the distances $d_{x,i=1...n}$: $X \backsim N(\hat{\mu}_x, \hat{\sigma}_x)$.

to the probability of x being the nearest neighbor of y, IV+III to the probability of y being a nearest neighbor of x and III to their joint probability:

$$II = MP(d_{x,y}) = 1 - [(I + III) + (IV + III) - III].$$

It is straightforward to compute MP using the empirical distribution, as illustrated in Figure 4. If the number of observations is large enough, we will tend to model the true underlying distribution closely. Computing MP for a given distance $d_{x,y}$ in a collection of n objects and using the empirical

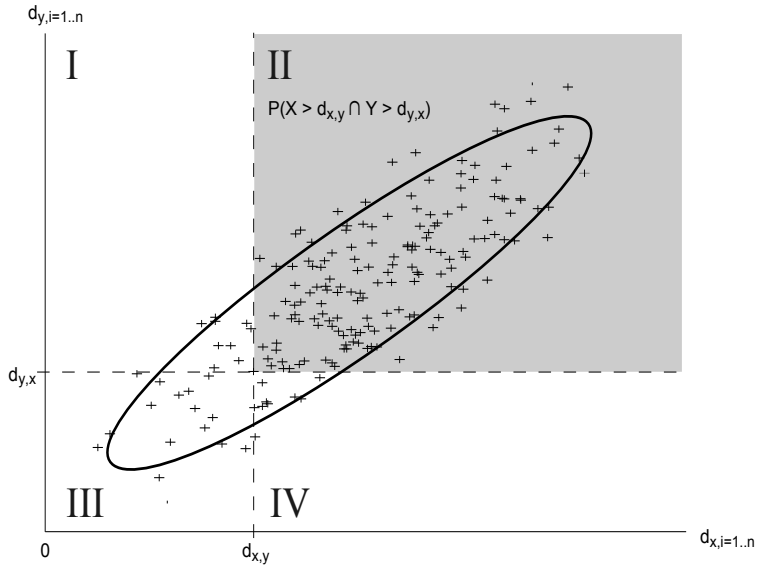


Figure 4: Visualizing the Mutual Proximity for the two points x, y, and their distances $d_{x,y}$, $d_{y,x}$.

distribution boils down to simply counting the number of objects j having a distance to x and y which is greater than $d_{x,y}$:

$$MP(d_{x,y}) = \frac{\left|\left\{j : d_{x,j} > d_{x,y}\right\} \cap \left\{j : d_{y,j} > d_{y,x}\right\}\right|}{n}.$$

For distances where the underlying distribution is known, estimating its parameters can be straightforward. For example, the parameters of normal distributions $N(\hat{\mu}, \hat{\sigma}^2)$, or Gamma distributions $\Gamma(\hat{k}, \hat{\theta})$ can be estimated quickly with the sample mean $\hat{\mu}_x$ and variance $\hat{\sigma}_x^2$ of the distances $d_{x,i=1..n}$:

$$\begin{aligned} N_x \backsim \quad & \hat{\mu}_x = \frac{1}{n}\sum_{i=1}^{n} d_{x,i}, \qquad \hat{\sigma}_x^2 = \frac{1}{n}\sum_{i=1}^{n}(d_{x,i} - \hat{\mu}_x)^2, \\ \Gamma_x \backsim \quad & \hat{k}_x = \frac{\hat{\mu}_x^2}{\hat{\sigma}_x^2}, \qquad \hat{\theta}_x = \frac{\hat{\sigma}_x^2}{\hat{\mu}_x}. \end{aligned} \tag{2}$$

In our experiments we will generally estimate MP directly from the empirical distribution. In addition we will also evaluate MP with different underlying parametric distance distributions, such as the Gauss or Gamma distribution (Section 4).

3.2.2 APPROXIMATIONS

The definition of MP (Definition 2) requires estimating a joint distribution $P(X,Y)$ for all distance pairs $d_{x,y}$, which is usually expensive to compute. On the other hand, if independence could be assumed between distributions $P(X)$ and $P(Y)$, the computation of MP would simplify in accordance with the product rule:

$$MP_I(d_{x,y}) = P(X > d_{x,y}) \cdot P(Y > d_{y,x}). \tag{3}$$

We will show in our experiments that assuming independence in the computation of MP does not affect the results in an adverse way (Section 4).

In the base case where MP is computed from the empirical distribution as well as all other variants presented so far, the computational cost of computing MP grows quadratically with the size of the data set as all methods require the full distance matrix (that is, all possible distances) to be computed. To circumvent this, we propose to estimate the distribution parameters by randomly selecting a small fraction of objects to compute the mean and standard deviation of distances for each object using only the subset of objects. We denote MP where the parameters have been estimated by sampling from the collection with MP_S. The parameter S specifies how many objects have been randomly sampled. The appropriate sample size is naturally dependent on the underlying distribution. However if a normal distribution may be assumed, a sample size as small as $S = 30$ will already yield stable results for MP.

The difference to the original estimation of the parameters in Equation 2 is that only a small fraction of distances ($S \times n$) needs to be computed, which, for constant S, reduces the complexity from quadratic to linear in n. This is also more efficient than local scaling, where the actual nearest neighbors of points x and y need to be identified. While local scaling methods can of course be used with fast nearest neighbor search algorithms indexing the high dimensional spaces, the complexity is far higher than randomly sampling only a constant number of distances.

Experimental verification that these approximations of the original idea are still valid will be presented in Section 4.

3.2.3 LINEAR COMBINATION OF DISTANCE MEASURES

Another nice property of MP which can be useful in some contexts is that MP yields $[0,1]$-normalized similarities. Thus, the MP transformation can easily be used to linearly combine multiple different distance measures d_1 and d_2 for some combination weights $\omega_{1,2}$:

$$d = \omega_1 MP(d_1) + \omega_2 MP(d_2).$$

Similar to a global zero-mean unit-variance normalization, each object's distances are also standardized by their respective mean and standard deviation. Thus, no distance measure can dominate the other in this combination. This property is useful in scenarios where multiple different distance measures (describing different aspects of a phenomenon) need to be linearly combined. A real-world example where this is necessary is presented in Section 5.

4. Evaluation

To investigate the effects of using local neighborhood scaling methods and MP, we first evaluate the methods on 30 public machine learning data sets. Each data set is characterized by the following parameters: name/origin, number of classes, size/number of items n and data dimensionality d. For each data set we evaluate the original distance space and compare it to the distances that are generated by the local scaling method and by MP.

After showing the impact of the scaling methods in regard to the hub problem on real data sets in the first set of experiments, a second series of experiments investigates the effects of the methods more deeply. Synthetic as well as real data is used.

4.1 Benchmarks

To quantify the impact of the two methods, a number of properties and quality measures are computed for the original and the new distances. The characteristics which we compute for each data set are:

4.1.1 LEAVE-ONE-OUT k-NEAREST NEIGHBOR CLASSIFICATION ACCURACY (C^k)

We report the k-nearest neighbor (kNN) classification accuracy using leave-one-out cross-validation, where classification is performed via a majority vote among the k nearest neighbors, with the class of the nearest neighbor used for breaking ties. We denote the k-NN accuracy as C^k. In the context of a retrieval problem, higher values would indicate better retrieval quality.[4]

To test for statistical significance differences in classification accuracy between two algorithms, we use McNemar's test (see Salzberg, 1997 and Dietterich, 1998 for a discussion of using this test in conjunction with leave-one-out classification). When comparing two algorithms A and B, only classification instances where A and B disagree are being analyzed. More specifically, it is tested whether the number of times that A classifies correctly and B does not is significantly different from the number of times B classifies correctly and A does not.

4.1.2 K-OCCURRENCE ($N^k(x)$)

Defines the k-occurrences of object x, that is, the number of times x occurs in the k nearest neighbor lists of all other objects in the collection.

4.1.3 HUBNESS (S^k)

We also compute the *hubness* of the distance space of each collection according to Radovanović et al. (2010). Hubness is defined as the skewness of the distribution of k-occurrences N_k:

$$S^k = \frac{\mathrm{E}\left[(N_k - \mu_{N_k})^3\right]}{\sigma_{N_k}^3}.$$

Positive skewness indicates high hubness, negative values low hubness.

4. To clarify the cross-validation (CV) process: We first compute the distance matrix for the entire data set of n instances, transform this into an MP matrix, and then perform leave-one-out cross-validation on the data set of n instances, in each iteration i using one of the n instances (x_i) as a test case to be classified by its nearest neighbors among the remaining $n-1$ instances. It might seem that the 'test' example x_i plays an undue role in this process, having contributed to the the normalization of the distance matrix before being used as a 'new' test case. However, in a 'real' classification scenario, where we would have a fixed training set X^{n-1} (consisting of $n-1$ instances) and are presented with a new object x_i to classify (not contained in X^{n-1}), we would also first have to compute the full distance matrix over $X^{n-1} \cup \{x_i\}$ in order to then be able to compute MP over this matrix. (That is because MP needs information about all distances to and from x_i.)

   The result of this would be exactly the MP matrix we compute beforehand in our cross-validation process—and it is the exact same matrix for all other 'test' instances from X. Thus, it is legitimate to compute this once and for all before the CV. On the other hand, the above means that using MP in a 'real' classification scenario is expensive, because before being able to classify a new instance, first a complete distance and MP matrix have to be computed. What makes this process feasible in practice is the MP approximation MP_S described in Section 3.2.2.

4.1.4 GOODMAN-KRUSKAL INDEX (I_{GK})

The Goodman-Kruskal Index (Günter and Bunke, 2003) is a clustering quality measure that relates the number of *concordant* (Q_c) and *discordant* (Q_d) tuples ($d_{i,j}$, $d_{k,l}$) of a distance matrix.

- A tuple is concordant if its items i, j are from the same class, items k, l are from different classes and $d_{i,j} < d_{k,l}$.
- A tuple is discordant if its items i, j are from the same class, items k, l are from different classes and $d_{i,j} > d_{k,l}$.
- A tuple is not counted if it is neither concordant nor discordant, that is, if $d_{i,j} = d_{k,l}$.

The Goodman-Kruskal Index (I_{GK}) is defined as:

$$I_{GK} = \frac{Q_c - Q_d}{Q_c + Q_d}.$$

I_{GK} is bounded to the interval $[-1,1]$, and the higher I_{GK}, the more concordant and fewer discordant quadruples are present in the data set. Thus a large index value indicates a good clustering (in terms of *pairwise stability*—see Section 2).

Other indices to compare clustering algorithms like the classic Dunn's Index or Davies-Bouldin Index (Bezdek and Pal, 1998) cannot be used here as their values do not allow a comparison across different distance measures.

4.1.5 INTRINSIC DIMENSIONALITY (d_{mle})

To further characterize each data set we compute an estimate of the intrinsic dimensionality of the feature spaces. Whereas the embedding dimension is the actual number of dimensions of a feature space, the intrinsic dimension is the—often much smaller—number of dimensions necessary to represent a data set without loss of information. It has also been demonstrated that hubness depends on the intrinsic rather than embedding dimensionality (Radovanović et al., 2010). We use the maximum likelihood estimator proposed by Levina and Bickel (2005), which was also used by Radovanović et al. (2010) to characterize the data sets.

4.1.6 PERCENTAGE OF SYMMETRIC NEIGHBORHOOD RELATIONS

We call a nearest neighbor relation between two points x and y 'symmetric' if the following holds: object x is a nearest neighbor of y if and only if y is also the nearest neighbor of x. As both examined methods aim at symmetrizing neighborhood relations, we report the percentage of symmetric relations at different neighborhood sizes k.

4.2 Public Machine Learning Data Sets

We evaluate the proposed method by applying it to 30 different public machine learning data sets. The data sets include problems from the general machine learning field, and the bio-medical, image, text and music retrieval domains. We use the following data sets:

- The UCI Machine Learning Repository (*UCI*, see Frank and Asuncion, 2010) data sets: *arcene*, *gisette*, *mfeat-pixels/karhunen/factors*, *dexter*, *mini-newsgroups*, *dorothea*, *reuters-transcribed*.[5]
- The Kent Ridge bio-medical data sets (*KR*): *amlall*, *lungcancer* and *ovarian-61902*.[6]
- The LibSVM data sets (*LibSVM*, see Chang and Lin, 2001): *australian*, *diabetes*, *german numbers*, *liver-disorders*, *breast-cancer*, *duke (train)*, *heart*, *sonar*, *colon-cancer*, *fourclass*, *ionosphere*, *splice*.[7]
- The Music Information Retrieval Evaluation eXchange (MIREX) data sets (*Mirex*, see Downie, 2008): *ballroom* and *ismir2004*.[8]
- Two music artist web pages and tweets data sets (*CP*, see Schedl, 2010): *c1ka-twitter* and *c224a-web*.[9]

For the general machine learning data sets from the statistical or biological domains no feature extraction is necessary. The feature vectors can be downloaded directly from the respective repositories. These general machine learning data sets use the standard Euclidean distance (denoted as ℓ^2) as similarity measure.

The text retrieval data sets (*reuters-transcribed*, *c224a-web*, *movie-reviews*, *dexter*, *mini-newsgroups*, *c1ka-twitter*) need to be preprocessed before evaluating them.[10] To this end we employ stop-word removal and stemming. They are transformed into the bag-of-words representation, and standard $tf \cdot idf$ (term frequency $\cdot$ inverse document frequency) weights are computed (see for example Baeza-Yates and Ribeiro-Neto, 1999). The word vectors are normalized to the average document length. Individual document vectors are compared with the cosine distance (denoted as *cos*).

For the image retrieval data set (*corel*) normalized color histograms are computed as features. They show reasonable classification performance despite their simplicity, as Chapelle et al. (1999) show. The three 64-dimensional color histograms are concatenated into a single vector and compared using the Euclidean distance (ℓ^2).

To extract the features for the two music information retrieval data sets (*ismir2004*, *ballroom*) we use a standard algorithm from Mandel and Ellis (2005) which computes Mel Frequency Cepstrum Coefficients (Logan, 2000) and models each object as a multivariate normal distribution over these. Objects (models) are compared using the symmetrized Kullback-Leibler divergence (denoted as *skl*).

4.3 Results

In the following experiments we compute all previously introduced benchmark numbers for the original data and the distance spaces after applying the scaling methods (NICDM, MP). We use MP as defined in Section 3.2 and model the distance distributions with the empirical distribution.

5. The UCI Repository can be found at `http://archive.ics.uci.edu/ml/`.
6. The Kent Ridge data sets can be found at `http://datam.i2r.a-star.edu.sg/datasets/krbd/`.
7. LibSVM can be found at `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.
8. The MIREX data sets can be found at `http://www.music-ir.org/mirex`.
9. These data sets can be found at `http://www.cp.jku.at/people/schedl/datasets.html`.
10. Set *c1ka-twitter* equals *c3ka-twitter* from *CP*, omitting artists classified as 'rock' to balance the data.

Tables 1 and 2 show the results of the evaluations conducted on the 30 previously introduced public data sets. The collections have very diverse properties. There are collections like *fourclass* or *liver-disorders* with very low dimensionality ($d = 2$ and $d = 6$), as well as data sets with very high embedding dimensionality, such as *dorothea* ($d = 100\,000$) or *c1ka-twitter* ($d = 49\,820$). Related to that, column d_{mle} lists the intrinsic dimensionality according to the maximum likelihood estimator. Using the intrinsic dimensionality estimate we can see that there are data sets where the data is originally represented using high-dimensional feature vectors, although the data's intrinsic dimensionality is quite low. For example the *ovarian_61902* data set has an embedding dimension of $d = 15\,154$ but its estimated intrinsic dimension is only $d_{mle} = 9$.

The evaluation results in Tables 1 and 2 are sorted by the hubness $S^{k=5}$ of the original distance space (printed in bold). In subsequent plots individual collections are referenced by their numbers as given in Tables 1 and 2. The columns $C^{k=1}/C^{k=5}$ show the k-nearest neighbor classification rates of the collections. The classification rates with the original distances, the local scaling (NICDM) and the global scaling (MP) are documented. For convenience the column +/- shows the difference in classification accuracy, in terms of absolute percentage points, between the original distances and NICDM/MP. All improvements compared to the original distances are printed in bold. Statistically significant differences are marked with an asterisk (McNemar's test, $df = 1$, $\alpha = .05$ error probability).

Looking at the tables, a first observation is that very high-dimensional data sets (in terms of their intrinsic as well as their embedding dimensionality) also tend to have high hubness. This is in agreement with the results of Radovanović et al. (2010) and the theory that hubness is a consequence of high dimensionality.

By looking at the classification rates (columns $C^{k=1}$ and $C^{k=5}$) it can also clearly be observed that the higher the hubness and intrinsic dimensionality, the more beneficial, in terms of classification accuracy, NICDM and MP. For data sets with high hubness (in the collections we used, a value above 1.4 seems to be a limit) the increase in classification rates is notable. For $C^{k=1}$, the accuracy gain ranges from rather moderate 1 to up to $7-8$ percentage points, and in the case of *c1ka-twitter* it is 15.9 percentage points for NICDM and 17.1 percentage points for MP. For $C^{k=5}$ the trend is even clearer. Whereas only three changes in accuracy (relative to the original distances) are significant for data sets with low hubness ($S^{k=5} \leq 1.4$, data sets 1–17), 34 changes in accuracy are significant for data sets with high hubness ($S^{k=5} > 1.4$, data sets 18–30). There is no statistically significant negative change in terms of classification accuracies.

Figures 5 and 6 (left hand sides) present these results in bar plots where the y-axis shows the index of the data sets (ordered according to hubness as in Tables 1 and 2) and the bars show the increase or decrease of classification rates. The bar plots also directly show how MP compares to NICDM in terms of classification accuracy for $k = 1, 5$. Generally speaking, results for MP and NICDM are very comparable. As for $k = 1$, MP and NICDM perform equally well and there is no statistically significant difference between MP and NICDM (McNemar's test, $df = 1$, $\alpha = .05$ error probability). Based on the same statistical testing procedure, results for NICDM and $k = 5$ are significantly better than for MP for data sets 18, 20, 22 (marked with asterisks in Figure 6). The general tendency of both MP and NICDM is comparable in the sense that if there is an improvement compared to the original distances, it can be seen for both MP and NICDM.

Another observation from the results listed in Tables 1 and 2 is that both NICDM and MP reduce the hubness of the distance space for all data sets to relatively low values. The hubness $S^{k=5}$ decreases from an average value of 2.5 (original) to 0.29 (MP) and 0.94 (NICDM), indicating a

Name/Src.	Cls.	n	d	d_{mle}	Dist.	$C^{k=1}$	+/- %-pts	$C^{k=5}$	+/- %-pts	$S^{k=5}$	I_{GK}
fourclass (sc)	2	862	2	2	ℓ^2	100%		100%		**0.15**	0.22
1. *LibSVM*					NICDM	100%	0	100%	0	0.06	0.21
					MP	100%	0	100%	0	0.04	0.23
arcene	2	100	10 000	399	ℓ^2	82.0%		75.0%		**0.25**	0.07
2. *UCI*					NICDM	81.0%	-1.0	77.0%	**2.0**	-0.27	0.06
					MP	80.0%	-2.0	81.0%	**6.0**	0.15	0.10
liver-disorders (sc)	2	345	6	6	ℓ^2	62.6%		60.6%		**0.39**	0.00
3. *UCI*					NICDM	63.2%	**0.6**	65.8%	***5.2**	-0.04	0.03
					MP	62.9%	**0.3**	65.5%	***4.9**	-0.03	0.01
australian	2	690	14	3	ℓ^2	65.5%		68.8%		**0.44**	0.13
4. *LibSVM*					NICDM	65.7%	**0.2**	69.4%	**0.6**	-0.09	0.14
					MP	65.4%	-0.1	68.4%	-0.4	0.08	0.14
diabetes (sc)	2	768	8	6	ℓ^2	70.6%		74.1%		**0.49**	0.20
5. *UCI*					NICDM	69.8%	-0.8	74.1%	0	0.04	0.15
					MP	70.3%	-0.3	73.2%	-0.9	-0.02	0.19
heart (sc)	2	270	13	7	ℓ^2	75.6%		80.0%		**0.50**	0.35
6. *LibSVM*					NICDM	75.9%	**0.3**	79.3%	-0.7	-0.00	0.27
					MP	75.6%	0	80.4%	**0.4**	0.08	0.39
ovarian-61902	2	253	15 154	10	ℓ^2	95.3%		93.7%		**0.66**	0.20
7. *KR*					NICDM	95.7%	**0.4**	93.3%	-0.4	-0.10	0.19
					MP	94.1%	-1.2	94.1%	**0.4**	-0.28	0.19
breast-cancer (sc)	2	683	10	5	ℓ^2	95.6%		97.4%		**0.71**	0.89
8. *LibSVM*					NICDM	95.8%	**0.2**	97.1%	-0.3	0.19	0.42
					MP	96.0%	**0.4**	97.1%	-0.3	0.22	0.91
mfeat-factors	10	2 000	216	7	ℓ^2	95.0%		94.7%		**0.79**	0.71
9. *UCI*					NICDM	94.8%	-0.2	94.7%	0	0.15	0.76
					MP	94.5%	-0.5	94.9%	**0.2**	0.01	0.77
colon-cancer	2	62	2 000	11	ℓ^2	72.6%		77.4%		**0.81**	0.19
10. *LibSVM*					NICDM	69.4%	-3.2	82.3%	**4.9**	0.08	0.18
					MP	67.7%	-4.9	82.3%	**4.9**	-0.11	0.19
ger.num (sc)	2	1 000	24	8	ℓ^2	67.5%		71.7%		**0.81**	0.07
11. *LibSVM*					NICDM	66.8%	-0.7	72.0%	**0.3**	0.32	0.03
					MP	67.6%	**0.1**	71.4%	-0.3	0.11	0.07
amlall	2	72	7 129	32	ℓ^2	91.7%		93.1%		**0.83**	0.31
12. *KR*					NICDM	93.1%	**1.4**	97.2%	**4.1**	0.56	0.33
					MP	88.9%	-2.8	91.7%	-1.4	-0.01	0.34
mfeat-karhunen	10	2 000	64	15	ℓ^2	97.4%		97.4%		**0.84**	0.76
13. *UCI*					NICDM	97.2%	-0.2	97.6%	**0.2**	0.27	0.74
					MP	97.0%	-0.4	97.5%	**0.1**	0.08	0.79
lungcancer	2	181	12 533	60	ℓ^2	98.9%		100%		**1.07**	0.56
14. *KR*					NICDM	99.4%	**0.5**	98.9%	-1.1	0.31	0.50
					MP	98.3%	-0.6	97.8%	-2.2	0.01	0.56
c224a-web	14	224	1 244	41	cos	86.2%		89.3%		**1.09**	0.79
15. *CP*					NICDM	87.9%	**1.7**	92.4%	***3.1**	0.42	0.89
					MP	88.4%	**2.2**	92.4%	**3.1**	0.22	0.89

Table 1: Evaluation results ordered by ascending hubness ($S^{k=5}$) of the original distance space. This table reports data sets with small hubness. Each evaluated data set (*Name/Src*) is described by its number of classes (*Cls.*), its size (n), its extrinsic (d) and intrinsic (d_{mle}) data dimension and the distance measure used (*Dist*). Columns C^k report the classification accuracies at a given k, the respective adjacent column +/- the difference in classification accuracy between the original distances and NICDM/MP (in percentage points), column I_{GK} the Goodman-Kruskal Index. See Section 4.1 for an explanation of the individual benchmarks.

Name/Src.	Cls.	n	d	d_{mle}	Dist.	$C^{k=1}$	+/- %-pts	$C^{k=5}$	+/- %-pts	$S^{k=5}$	I_{GK}
mfeat-pixels	10	2 000	240	12	ℓ^2	97.6%		97.7%		**1.25**	0.75
16. *UCI*					NICDM	97.2%	-0.4	97.8%	**0.1**	0.28	0.75
					MP	97.2%	-0.4	97.5%	-0.2	0.13	0.79
duke (train)	2	38	7 129	16	ℓ^2	73.7%		68.4%		**1.37**	0.02
17. *UCI*					NICDM	81.6%	**7.9**	68.4%	0	0.43	0.06
					MP	76.3%	**2.6**	68.4%	0	0.21	0.03
corel1000	10	1 000	192	9	ℓ^2	70.7%		65.2%		**1.45**	0.33
18. *Corel*					NICDM	72.9%	***2.2**	72.0%	***6.8**	0.39	0.47
					MP	71.6%	**0.9**	70.3%	***5.1**	0.31	0.50
sonar (sc)	2	208	60	11	ℓ^2	87.5%		82.2%		**1.54**	0.07
19. *UCI*					NICDM	87.0%	-0.5	87.0%	**4.8**	0.47	0.08
					MP	87.5%	0	84.1%	**1.9**	0.32	0.08
ionosphere (sc)	2	351	34	13	ℓ^2	86.9%		85.5%		**1.55**	0.31
20. *UCI*					NICDM	92.3%	***5.4**	94.3%	***8.8**	0.28	0.07
					MP	91.7%	***4.8**	89.7%	***4.2**	0.50	0.27
reuters-transcribed	10	201	2 730	70	cos	44.3%		49.3%		**1.61**	0.38
21. *UCI*					NICDM	45.3%	**1.0**	52.7%	**3.4**	0.63	0.32
					MP	42.3%	-2.0	55.2%	***5.9**	0.18	0.43
ballroom	8	698	820	12	skl	54.3%		48.1%		**2.98**	0.15
22. *Mirex*					NICDM	57.2%	**2.9**	51.6%	***3.5**	1.09	0.20
					MP	56.6%	**2.3**	54.3%	***6.2**	0.30	0.18
ismir2004	6	729	820	25	skl	80.4%		74.1%		**3.20**	0.37
23. *Mirex*					NICDM	83.8%	***3.4**	79.0%	***4.9**	0.77	0.21
					MP	83.4%	***3.0**	77.0%	***2.9**	0.46	0.45
movie-reviews	2	2 000	10 382	28	cos	71.1%		75.7%		**4.07**	0.05
24. *PaBo*					NICDM	72.0%	**0.9**	76.0%	**0.3**	1.22	0.07
					MP	71.8%	**0.7**	76.7%	**1.0**	0.36	0.07
dexter	2	300	20 000	161	cos	80.3%		80.3%		**4.22**	0.10
25. *UCI*					NICDM	84.3%	**4.0**	86.0%	***5.7**	2.02	0.13
					MP	83.0%	**2.7**	90.0%	***9.7**	0.58	0.13
gisette	2	6 000	5 000	149	ℓ^2	96.0%		96.3%		**4.48**	0.16
26. *UCI*					NICDM	97.2%	***1.2**	98.1%	***1.8**	0.78	0.20
					MP	97.4%	***1.4**	97.9%	***1.6**	0.34	0.20
splice (sc)	2	1 000	60	27	ℓ^2	69.6%		69.4%		**4.55**	0.07
27. *LibSVM*					NICDM	73.3%	***3.7**	79.3%	***9.9**	1.51	0.11
					MP	72.4%	**2.8**	77.2%	***7.8**	0.48	0.10
mini-newsgroups	20	2 000	8 811	188	cos	64.4%		65.6%		**5.14**	0.47
28. *UCI*					NICDM	67.2%	***2.8**	68.5%	***2.9**	1.32	0.52
					MP	67.7%	***3.3**	68.4%	***2.8**	0.60	0.57
dorothea	2	800	100 000	201	ℓ^2	90.6%		90.2%		**12.91**	0.21
29. *UCI*					NICDM	92.2%	**1.6**	93.0%	***2.8**	11.72	0.21
					MP	91.5%	**0.9**	93.1%	***2.9**	1.66	0.20
c1ka-twitter	17	969	49 820	46	cos	31.9%		26.6%		**14.63**	0.08
30. *CP*					NICDM	47.8%	***15.9**	53.0%	***26.4**	2.94	0.33
					MP	49.0%	***17.1**	50.8%	***24.2**	1.79	0.16

Table 2: Evaluation results ordered by ascending hubness ($S^{k=5}$) of the original distance space. This table reports data sets with large hubness. Each evaluated data set (*Name/Src*) is described by its number of classes (*Cls.*), its size (n), its extrinsic (d) and intrinsic (d_{mle}) data dimension and the distance measure used (*Dist*). Columns C^k report the classification accuracies at a given k, the respective adjacent column +/- the difference in classification accuracy between the original distances and NICDM/MP (in percentage points), column I_{GK} the Goodman-Kruskal Index. See Section 4.1 for an explanation of the individual benchmarks.

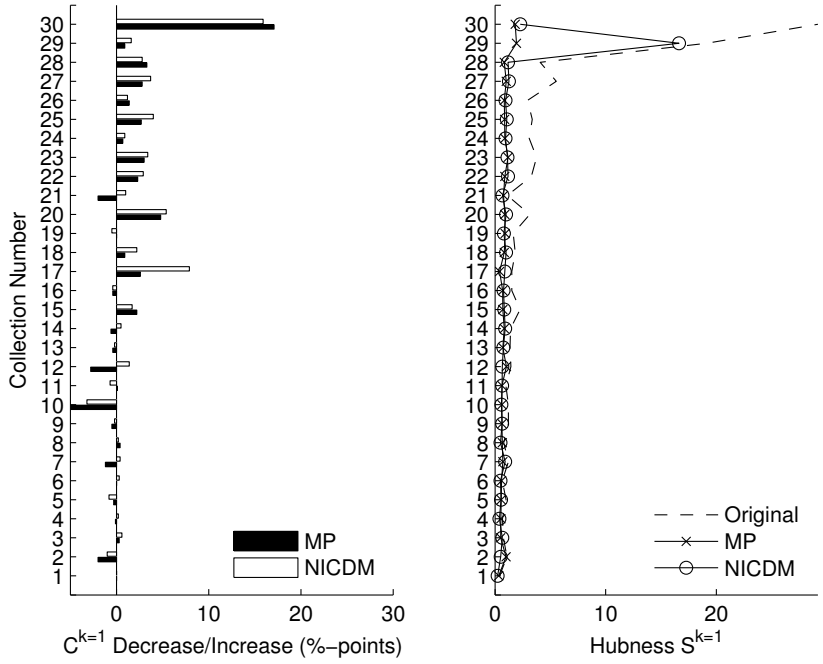


Figure 5: Improvements in accuracy (absolute percentage points) and hubness evaluated with $k = 1$.

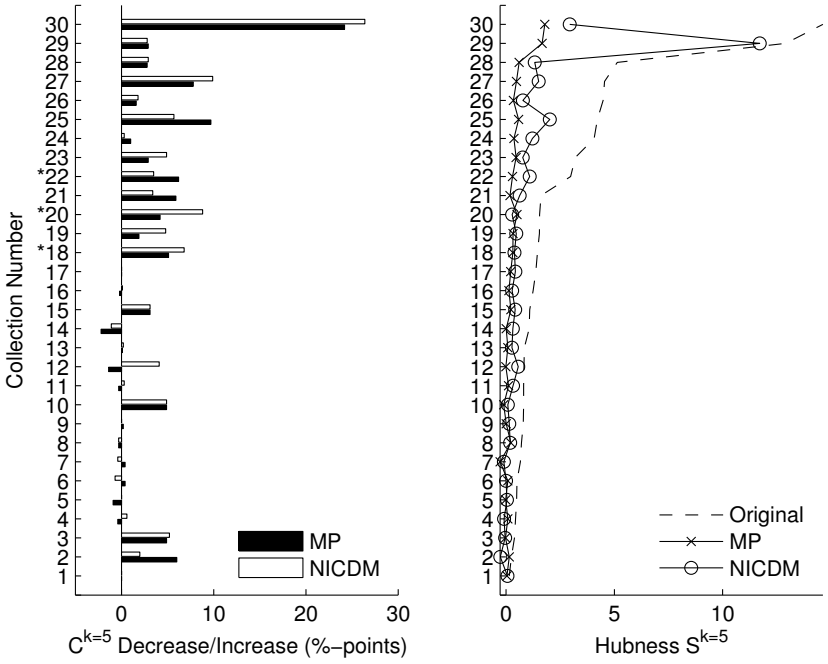


Figure 6: Improvements in accuracy (absolute percentage points, significant differences marked with an asterisk) and hubness evaluated with $k = 5$.

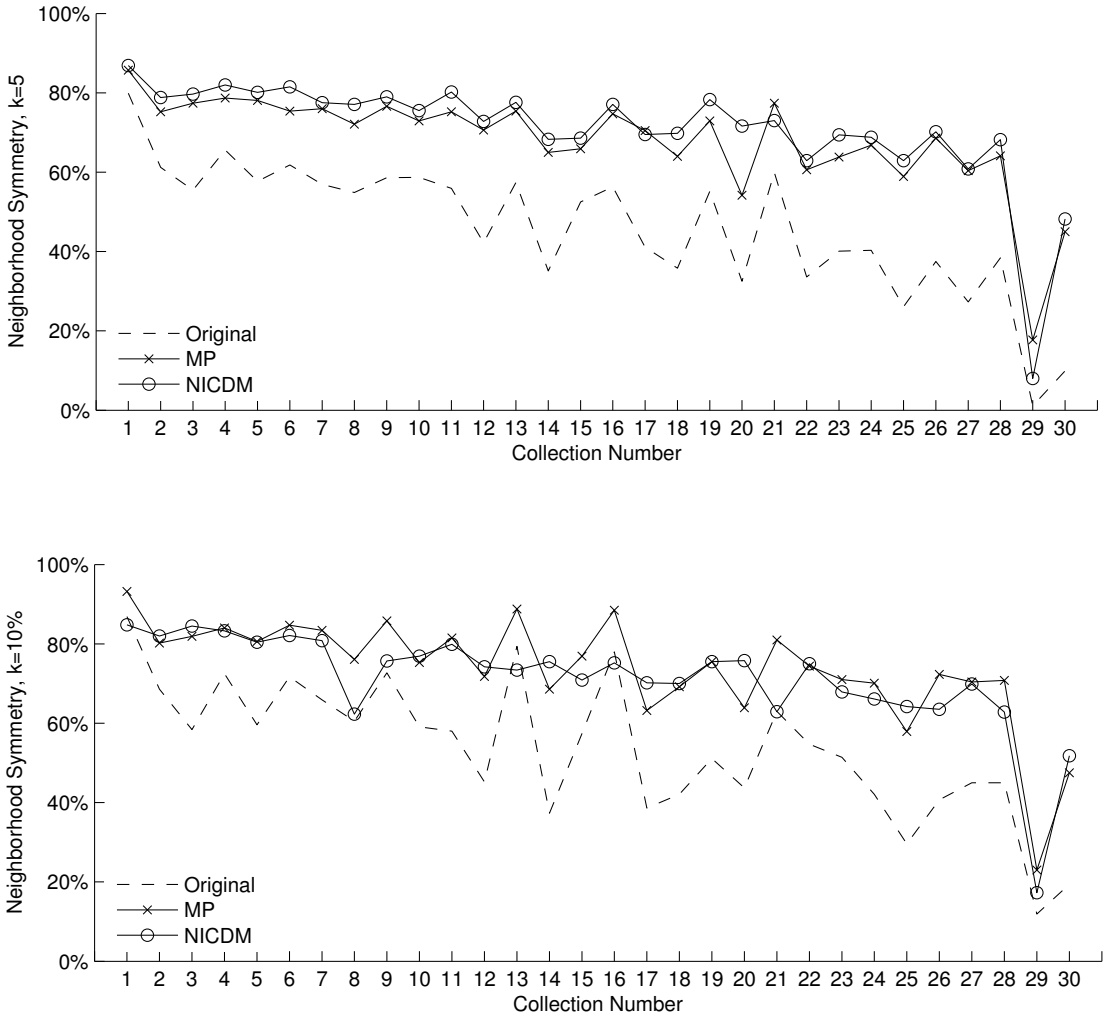


Figure 7: Percentage of symmetric neighborhood relations at $k = 5$ (above) and $k = 10\%$ (below) of the respective collection size.

well balanced distribution of nearest neighbors. The impact of MP and NICDM on the hubness per data set is plotted in Figures 5 and 6 (right hand sides). It can be seen that both MP and NICDM lead to lower hubness (measured for $S^{k=1,5}$) compared to the original distances. The effect is more pronounced for data sets having large hubness values according to the original distances.[11]

11. A notable exception is data set 29 ('dorothea') where the reduction in hubness is not so pronounced. This may be due to the extremely unbalanced distribution of its two classes (9:1).

More positive effects in the distances can also be seen in the increase of concordant (see Section 4 for the definition) distance quadruples indicated by higher Goodman-Kruskal index values (I_{GK}). This index improves or remains unchanged for 27 out of 30 data sets in the case of using MP. The effect is not so clear for NICDM, which improves the index or leaves it unchanged for only 17 out of 30 data sets. The effect of NICDM on I_{GK} is especially unclear for data with low hubness (data sets 1–17).

Finally we also checked whether both MP and NICDM are able to raise the percentage of symmetric neighborhood relations. Results for $k = 5$ and k set to 10% of the collection size (denoted by $k = 10\%$) are shown in Figure 7. As can be seen, the symmetry in the nearest neighbors for all data sets increases with both MP and NICDM. For NICDM there are two cases (data set 13 and 16) where the neighborhood symmetry does not increase. The average percentage of symmetric neighborhoods across all data sets for $k = 5$ is 46% for the original distances, 69% for MP, and 70.8% for NICDM. The numbers for $k = 10\%$ are 53% (original), 73.7% (MP), and 71.1% (NICDM).

4.4 Approximations

The general definition of MP (Definition 2, Section 3.2.1) allows for more specific uses if the underlying distribution of distances is known. All experiments conducted up until now use MP with the all-purpose empirical distribution. This section evaluates the use of different distributions in MP. Specifically, we will compare a Gaussian and a Gamma modeling to using the empirical distribution. For the two selected distributions, parameter estimation is straightforward (see Section 3.2.1). In case of the Gaussian, we will compute MP as it was defined. In our experiments this configuration will be denoted and referenced with 'MP (Gauss)'. As this variant involves computing a joint distribution in every step and this is expensive to calculate, there is no advantage to the original MP. Where things get interesting from a computational point of view, is using MP and assuming independence (MP_I, see Equation 3). In this case computing the joint distribution can be omitted. In our experiments we use the Gamma (denoted with, 'MP (i.Gamma)') and Gauss (denoted with 'MP (i.Gauss)') distribution with MP assuming independence.

Figure 8 plots the result of this experiment in the same way as we have done in the previous section. We compare the decrease/increase of classification accuracies and hubness at $k = 5$. Looking at the results, we can see that all methods seem to perform equally in terms of reducing hubness and increasing classification accuracies. More importantly, we notice that the simple variant ('MP (i.Gauss)'), which assumes a Gaussian distribution of distances and independence, performs similarly to all other variants.

This leads to the next experiment where we compare MP to a very simple approximation MP_S (see Section 3.2.2). As discussed in Section 3.2.2, assuming a Gaussian or Gamma distance distribution requires only a small sample size ($S = 30$) for a good estimate of the distribution parameters. Paired with the already evaluated simplification of MP assuming independence when computing the joint probability, MP is ready to be used instantly with any data collection. Figure 9 shows the results of a comparison of MP_S to MP. The classification accuracies are averages over ten approximations, that is, based on using ten times thirty randomly drawn data points for every data set. As can be seen, accuracy results are very comparable. We recorded three statistically significantly different results for MP_S using the approximative Gamma and Gauss variant (data sets 2, 10, 21, McNemar's test, $df = 1$, $\alpha = .05$ error probability). We also notice that with a sample size of $S = 30$ the decrease in hubness is not as pronounced for MP_S as for MP.

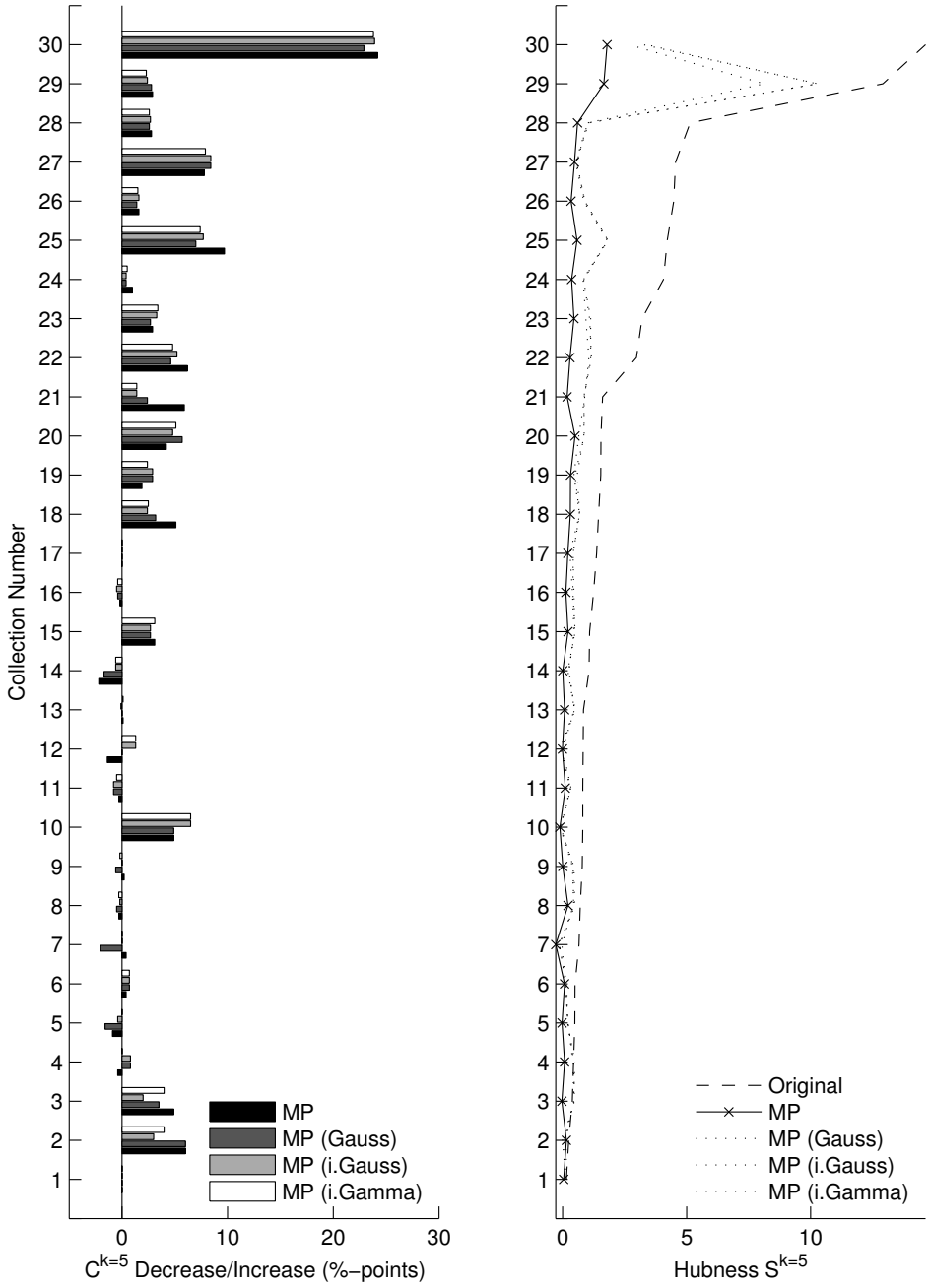


Figure 8: Comparison of different distance distributions in MP in terms of classification rates and hubness.

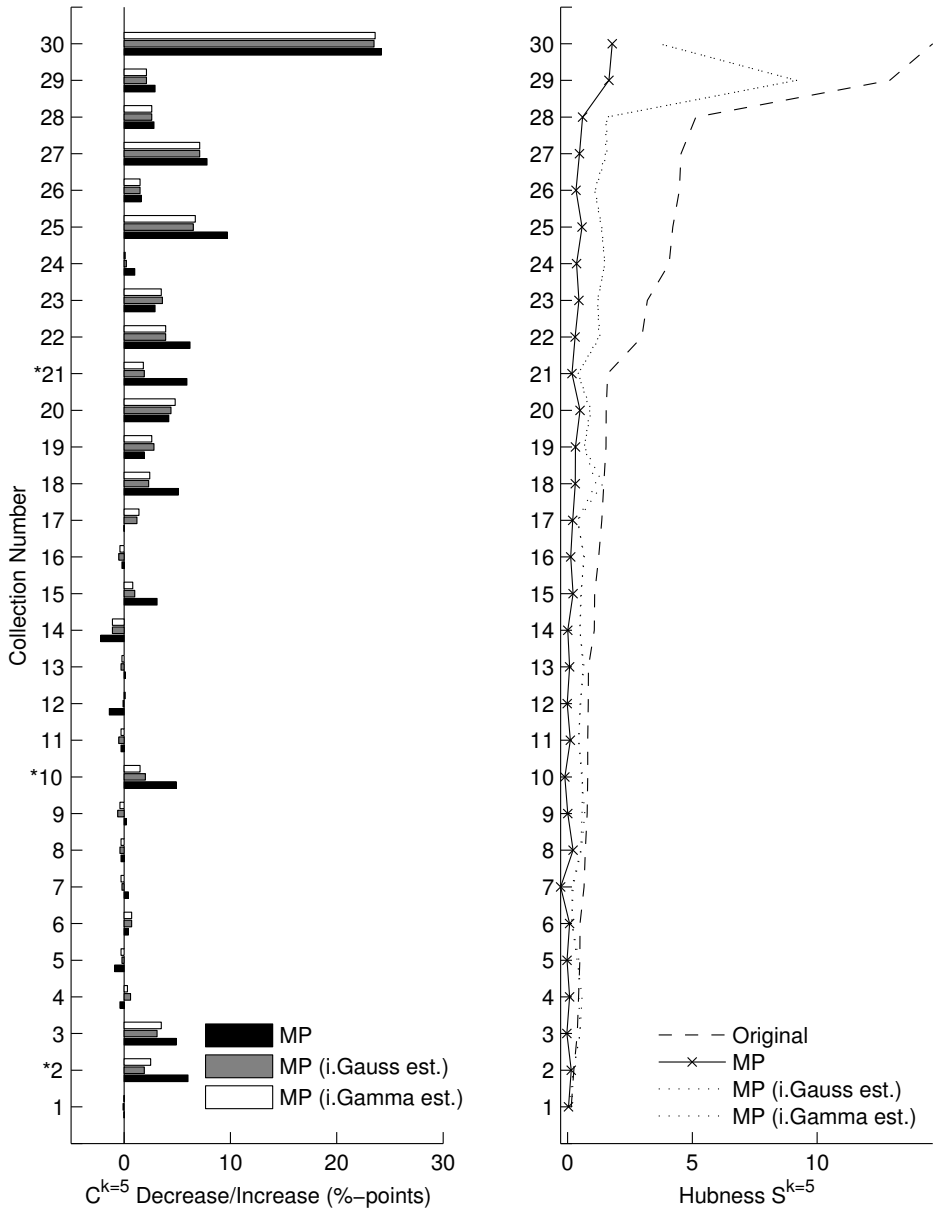


Figure 9: Improvements in accuracy (absolute percentage points, significant differences marked with an asterisk) and hubness evaluated with $k = 5$ for MP (black) and its approximative variant MP_{S} (gray).

4.5 Further Evaluations and Discussion

The previous experimental results suggest that the considered distance scaling methods work well as they tend to reduce hubness and improve the classification/retrieval accuracy. In the following three experiments we examine the scaling methods on artificial data as well as real data in order to investigate the following three questions:

1. Does NICDM/MP work by effectively reducing the intrinsic dimensionality of the data?
2. What is the impact of NICDM/MP on hubs and orphans?
3. Is the changing role of hubs responsible for improved classification accuracy?

The artificial data used in the experiments is generated by randomly sampling i.i.d. high-dimensional data vectors ($n = 1\,000$) in the hypercube $[0,1]^d$ from the standard uniform distribution. We use the Euclidean distance function and MP with the empirical distribution in all experiments.

4.5.1 DIMENSIONALITY

As we have already shown that hubs tend to occur in high dimensional spaces, the first experiment examines the consequential question if the scaling methods actually reduce the intrinsic dimensionality of the data. In order to test this hypothesis, the following simple experiment was performed: We increase the dimensions of artificial data (generated as described above) to create high hubness, and measure the intrinsic dimensionality of the data spaces before and after scaling the distances with NICDM/MP.

We start with a low data dimensionality ($d = 5$) and increase the dimensionality to a maximum of $d = 50$. In each iteration we measure the hubness of the data and its intrinsic dimensionality. The maximum likelihood estimator proposed by Levina and Bickel (2005) is used to estimate the intrinsic dimensionality of the generated vector spaces.

In Figure 10a we can see that a vector space dimension as low as 30 already leads to a distance space with very high hubness ($S^{k=5} > 2$). We can further see that NICDM/MP are able to reduce the hubness of the data spaces as expected. Figure 10b shows the measured intrinsic dimensionality of the original data. As anticipated it increases with its embedding dimensionality. However, to measure the intrinsic dimensionality of the data spaces created by MP and NICDM, we first have to map their distance space to a vector space. We perform this vector mapping using multidimensional scaling (MDS), doubling the target dimensionality to ensure a good mapping.

Figure 11 shows the results. For verification purposes, we (i) also map the original distance space with MDS and (ii) re-compute the hubness for the new data spaces (Figure 11a). Figure 11b finally compares the measured intrinsic dimensionality. We can clearly see that neither MP or NICDM decreases the intrinsic dimensionality notably. In none of the experiments does the estimated intrinsic dimensionality of the new distance space fall below the one measured in the original space.

4.5.2 IMPACT ON HUBS/ORPHANS

In the second experiment, we evaluate the question of what exactly happens to the hub and anti-hub (orphan) objects. Do hubs, after scaling the distances, still remain hubs (but 'less severely' so), or do they stop being hubs altogether? To look into this, we repeatedly generate a random, artificial, and

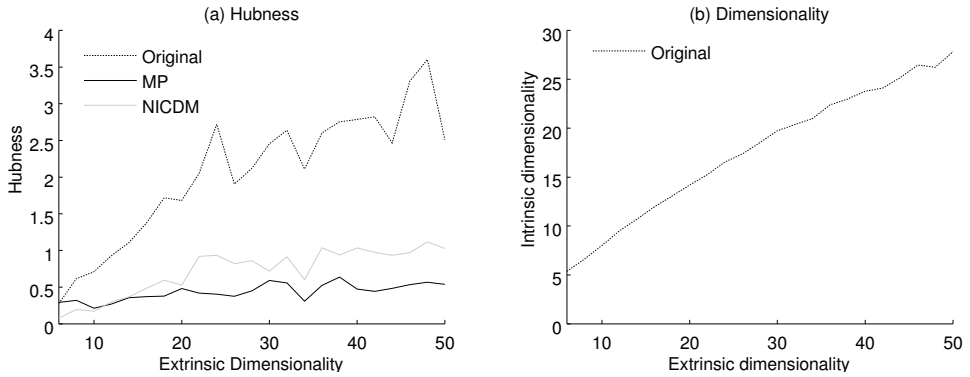


Figure 10: Increasing the dimensionality of artificially generated random data. Measuring (a) hubness of the original and scaled data, (b) the intrinsic dimensionality of the original data.

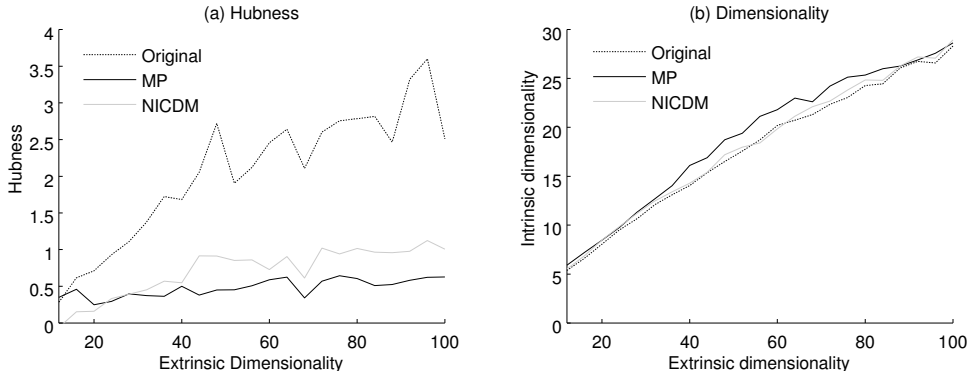


Figure 11: A vector space mapping of the distance spaces generated in Figure 10 allows to compare the intrinsic dimensionality of the original, MP, and NICDM data-spaces. No decrease of the intrinsic data dimensionality by using NICDM/MP can be observed.

high-dimensional ($d = 50$) data sample to (i) track hub and anti-hub objects and (ii) compute their k-occurrence (N^k) in the original space and in the distance spaces created after applying MP and NICDM. We define 'hub' objects as objects with a k-occurrence in the nearest neighbors greater than $5k$ and 'orphan' objects as having a k-occurrence of zero ($k = 5$). The experiment is repeated 100 times and for each iteration the observed mean k-occurrence of hubs/orphans is plotted in Figure 12.

Looking at the figure we can confirm that for the two studied cases (hubs/orphans) a weakening of the effects can be observed: after scaling the distances, hubs do not occur as often as nearest neighbors any more, while orphans re-appear in some nearest-neighbor lists. The k-occurrence of all other objects stays constant. Another observation is that in no instance of the experiment do hubs become orphans or orphans become hubs, as the measured $N^{k=5}$ never cross for the two classes.

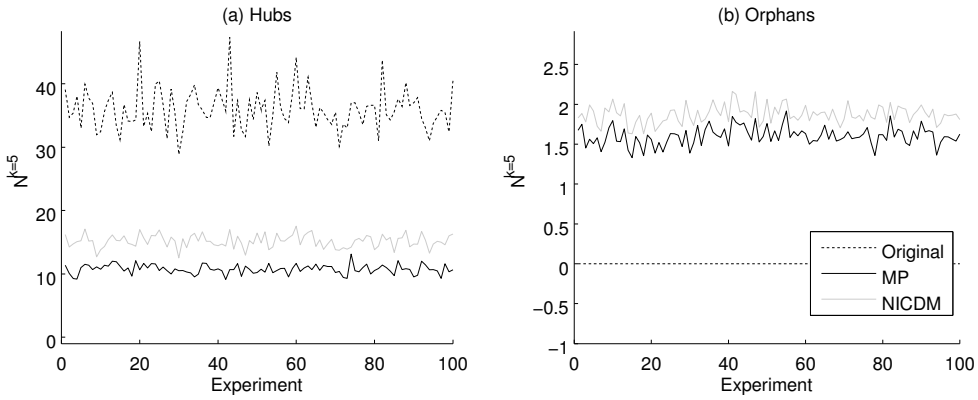


Figure 12: The k-occurrence ($N^{k=5}$) of hub and orphan data points before and after applying any of the scaling methods (NICDM, MP). Orphans re-appear in the nearest neighbor lists and the strength of hubs is reduced.

4.5.3 IMPACT OF HUBS/ORPHANS

In the final experiment we examine the increase in classification accuracies we observed previously when using NICDM or MP on the high dimensional machine learning data sets. To learn where the increase in classification accuracy came from, we distinguish between hubs, orphans, and all other objects. For each of the three classes we compute the so-called 'badness' ($BN^{k=5}$) as defined by Radovanović et al. (2010). Badness of an object x is the number of its occurrences as nearest neighbor at a given k where it appears with a different (that is, 'bad') class label. As this experiment makes only sense in collections with more than one class showing high hubness, we select machine learning data sets with high hubness of $S^{k=5} > 2$ from the 30 previously used databases. Table 3 documents the results of this experiment on the nine selected data sets.

For each collection the table shows the absolute number of hubs, orphans, and all other objects in the original data space. We then compute their badness before (columns *Orig.*) and after applying MP and NICDM. It can be clearly seen that indeed in each of the tested collections the badness of hubs decreases noticeably. In fact, on average $BN^{k=5}$ decreases more than 10 percentage points from 46.3% in the original space to 35.6% (NICDM) and 35.3% (MP). Another visible effect is that orphans re-appear in the nearest neighbor lists (see previous experiment, Figure 12) with an average badness of 36.5% (NICDM) and 35.1% (MP). The measured badness of orphan objects is comparable to the values reported for hubs, but is still notably higher than the numbers computed for the rest of the objects ('Other'). The badness of all other objects tends to stay the same: In three cases the badness increases slightly, in all other cases a slight decrease in badness can be observed. On average, badness decreases from 29.3% to 28.4% for both methods (MP and NICDM).

4.6 Summary of Results

Our main result is that both global (MP) and local (NICDM) scaling show very beneficial effects concerning hubness on data sets that exhibit high hubness in the original distance space. Both methods are able to decrease the hubness, raise classification accuracy, and improve other indicators

	Hubs, $BN^{k=5}$ (%)				*Orphans, $BN^{k=5}$ (%)*				*Other, $BN^{k=5}$ (%)*			
Data Set	#	Orig.	NICDM	MP	#	Orig.	NICDM	MP	#	Orig.	NICDM	MP
c1ka-twitter	13	83.5	54.0	55.7	540	/	59.3	59.2	416	46.2	47.9	50.1
dorothea	19	10.2	9.7	6.8	730	/	10.4	10.6	51	8.6	7.1	4.9
mini-newsgroups	38	67.2	62.2	60.7	304	/	45.6	43.5	1 658	42.2	41.5	41.6
splice (sc)	28	36.5	29.3	28.6	289	/	31.8	30.9	683	35.0	31.5	31.7
gisette	49	18.9	10.9	9.8	635	/	7.9	8.1	5316	4.7	4.0	3.9
dexter	11	44.3	27.9	28.4	80	/	33.5	30.5	209	18.2	18.1	17.7
movie-reviews	50	37.5	35.4	36.2	293	/	36.0	36.3	1 657	31.5	32.0	32.3
ismir2004	10	50.3	27.8	27.3	120	/	44.2	38.0	599	25.7	24.4	25.0
ballroom	12	67.9	62.8	63.8	148	/	59.5	58.6	538	51.6	49.0	48.3
Average (%-points):		46.3	35.6	35.3		/	36.5	35.1		29.3	28.4	28.4

Table 3: Relative badness ($BN^{k=5}$) of hub objects ($N^{k=5} > 5k$), orphan objects ($N^{k=5} = 0$), and all other objects. Data sets with $S^{k=5} > 2$.

like percentage of concordant distance quadruples or symmetric neighborhood relations. In case of MP, its approximation MP_S is able to perform at equal level with substantially less computational cost ($O(n)$, as opposed to $O(n^2)$ for both MP and local scaling). For data sets exhibiting low hubness in the original distance space, improvements are much smaller or non-existent, but there is no degradation of performance.

We have also shown that while MP and NICDM reduce hubness, which tends to occur as a consequence of high dimensional data, both methods do not decrease the intrinsic dimensionality of the distance spaces (at least for the type of data and measure of intrinsic dimensionality used in our experiments). By enforcing symmetry in the neighborhood of objects, both methods are able to naturally reduce the occurrence of hubs in nearest neighbor lists. Interestingly, at the same time as the occurrence of hubs in nearest neighbor lists decreases, hubs also lose their badness in terms of classification accuracy.

5. Mutual Proximity and Content-Based Music Similarity

This section presents an application where we can use Mutual Proximity, its approximation MP_S (Section 3.2.2) and a linear combination of multiple similarity measures (Section 3.2.3) to improve the retrieval quality of the similarity algorithm significantly. We chose to include this example as it demonstrates how MP_S with all its aspects introduced above can improve the quality of a real world application: the FM4 Soundpark.

The FM4 Soundpark is a web platform run by the Austrian public radio station FM4, a subsidiary of the Austrian Broadcasting Corporation (ORF).[12] The FM4 Soundpark was launched in 2001 and has gained significant public attention since then. Registered artists can upload and present their music free of charge. After a short editorial review period, new tracks are published on the front-page of the website. Older tracks remain accessible in the order of their publication date and in a large alphabetical list. Visitors of the website can listen to and download all the music at no cost. The FM4 Soundpark attracts a large and lively community interested in up and coming music,

12. FM4 Soundpark can be found at `http://fm4.orf.at/soundpark`.

Figure 13: The FM4 Soundpark music player web interface.

and the radio station FM4 also picks out selected artists and plays them on terrestrial radio. At the time of writing, there are more than 11 000 tracks by about 5 000 artists listed in the on-line catalog.

Whereas chronological publishing is suitable to promote new releases, older releases tend to disappear from the users' attention. In the case of the FM4 Soundpark this had the effect of users mostly listening to music that is advertised on the front-page, and therefore missing the full musical bandwidth. To allow access to the full database regardless of publication date of a song, we implemented a recommendation system using a content-based music similarity measure (see Gasser and Flexer, 2009 for a more detailed discussion of the system).

The user interface to the music recommender has been implemented as an Adobe Flash-based MP3 player with integrated visualization of the five songs most similar to the one currently playing. This web player can be launched from within an artist's web page on the Soundpark website by clicking on one of the artist's songs. Additionally to offering the usual player interface (start, stop, skipping forward/backward) it shows songs similar to the currently playing one in a text list and in a graph-based visualization (see Figure 13). The similar songs are retrieved by using an audio similarity function.

The graph visualization displays an incrementally constructed nearest neighbor graph (number of nearest neighbors = 5).

5.1 Similarity

The distance function used in the Soundpark to quantify music similarity was described by Pampalk (2006). To compute a similarity value between two music tracks (x, y), the method linearly combines rhythmic (d_r) and musical timbre (d_t) similarities into a single general music similarity (d) value. To combine the different similarities, they are normalized to zero-mean and unit-variance using static

normalization values (μ_r/σ_r, μ_t/σ_t) precomputed from a fixed training collection:

$$d(x,y) = 0.3\frac{d_r(x,y)-\mu_r}{\sigma_r} + 0.7\frac{d_t(x,y)-\mu_t}{\sigma_t}. \quad (4)$$

5.2 Limitations

The above algorithm for computing music similarity creates nearest neighbor lists which exhibit very high hubness. In the case of the FM4 Soundpark application, which always displays the top-5 nearest neighbors of a song, a similarity space with high hubness has an immediate negative impact on the quality of the results of the application. High hubness leads to circular recommendations and to the effect that some songs never occur in the nearest neighbor lists at all—hubs push out other objects from the $k = 5$ nearest neighbors. As a result of high hubness only 72.63% of the songs are reachable in the recommendation interface using the standard algorithm, that is, over a quarter of songs can never be reached in the application (more details are discussed in the next section).

In the following, we show that MP can improve this considerably. We use MP with two of the above mentioned aspects: (i) the linear combination of multiple similarity measures to combine timbre and rhythm similarities, and (ii) the approximation of the MP parameters, as computing all pairwise similarities would be highly impractical in a collection of this size.

5.3 Evaluation and Results

To evaluate the impact of MP_S on the application, we use MP_S in the linear combination of the rhythmic d_r and timbre d_t similarities:

$$d_{MP_S}(x,y) = 0.3\,MP_{S=30}(d_r(x,y) + 0.7\,MP_{S=30}(d_t(x,y)),$$

and compare the result to the standard variant (Equation 4) of the algorithm. Table 5 shows the results of the comparison (including a random baseline algorithm). As with the machine learning data sets evaluated previously, we observe that the hubness $S^{k=5}$ (which is particularly relevant for the application) decreases from 5.65 to 2.32. This is also visible in the k-occurrence (N^k) of the biggest hub object, N^k_{max}, which for $k = 5$ decreases from 242, with the standard algorithm, to 70.

We also compute the *retrieval accuracy* R^k (the average ratio of song genre labels matching the query object's genre) for $k = 1, 5, 10$. For a query song x and a list of recommendations $i = 1 \ldots k$, the retrieval accuracy over their multiple genres is computed as:

$$R^k(x) = \frac{1}{k}\sum_{i=1}^{k}\frac{|Genres(x) \cap Genres(i)|}{|Genres(x) \cup Genres(i)|}.$$

Similarly to the increase in classification accuracies for the machine learning data sets, R^k increases in all configurations. The music genre labels used in this evaluations originate from the artists who uploaded their songs to the Soundpark (see Table 4 for the music genres and their distribution in the collection).

The decrease of hubness produced by MP_S leads to a concurrent increase in the reachability of songs in the nearest neighbor graphs. Instead of only 72.6%, 86.2% of all songs are reachable via $k = 5$ nearest neighbor recommendation lists. If the application were to randomly sample 5 recommendations from the $k = 10$ nearest neighbors, the reachability with MP_S would even increase

Pop	*Rock*	*Electronica*	*Hip-Hop*	*Funk*	*Reggae*
37.6%	46.3%	44.0%	14.3%	19.7%	5.3%

Table 4: Music genre/class distribution of the songs in the FM4 Soundpark collection used for our experiments. Each artist can assign a newly uploaded song to one or more of these predefined genres. There are a total of 11 229 songs in our collection snapshot. As every song is allowed to belong to more than one genre, the percentages in the table add up to more than 100%.

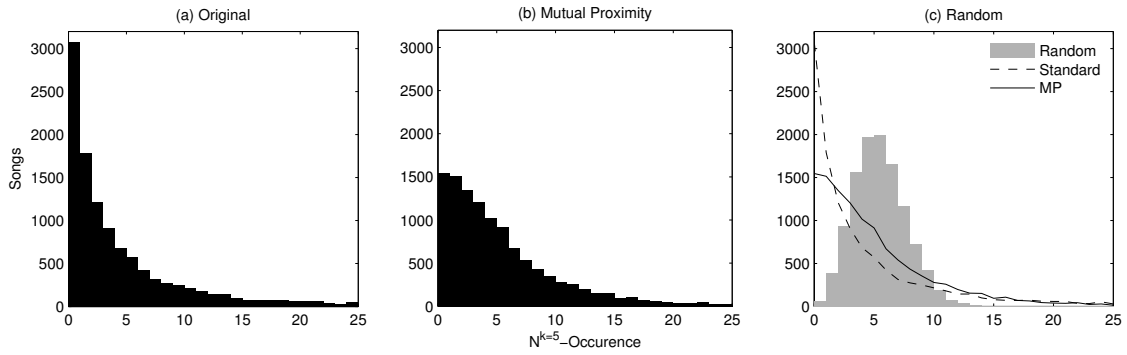


Figure 14: $N^{k=5}$-occurrences of songs in the nearest neighbors for (a) the standard algorithm, (b) the linear combination using $\mathrm{MP_S}$, and (c) random distances.

to 93.7% (from 81.9%) while the retrieval accuracy R^k for $k = 10$ would only slightly drop compared to $k = 5$.

Figure 14 shows a histogram plot of the $N^{k=5}$ occurrence of songs for the standard algorithm and $\mathrm{MP_S}$. The decrease of skewness is clearly visible as the number of songs that are never recommended drops from about 3000 to 1500—thus a more even distribution of objects in the nearest neighbors is achieved. The positive effects of using $\mathrm{MP_S}$ in this application are thus clearly visible: we obtain an improvement of retrieval accuracy and a decrease of hubness, paired with an increase of reachability in the nearest neighbors.

6. Conclusion

We have presented a possible remedy for the 'hubness' problems, which tend to occur when learning in high-dimensional data spaces. Considerations on the asymmetry of neighbor relations involving hub objects led us to evaluate a recent local scaling method, and to propose a new global variant named 'mutual proximity' (MP). In a comprehensive empirical study we showed that both scaling methods are able to reduce hubness and improve classification accuracy as well as other performance indices. Local and global methods perform at about the same level. Both methods are fully unsupervised and very easy to implement. Our own global scaling variant MP presented in this paper offers the additional advantage of being easy to approximate for large data sets which we show in an application to a real-world music recommendation service.

Characteristic	*Standard*	MP_S	(*Random*)
Retrieval Accuracy $R^{k=1}$	51.9%	54.5%	29.0%
Retrieval Accuracy $R^{k=5}$	48.2%	50.1%	28.5%
Retrieval Accuracy $R^{k=10}$	47.1%	48.6%	28.4%
Hubness $S^{k=5}$	5.65	2.31	0.46
Maximum Hub size $N_{max}^{k=5}$	242	70	17
Reachability $k = 5$	72.6%	86.2%	99.4%
Hubness $S^{k=10}$	5.01	2.14	0.36
Maximum Hub size $N_{max}^{k=10}$	416	130	25
Reachability $k = 10$	81.9%	93.7%	99.9%

Table 5: Evaluation results of the FM4 Soundpark data set comparing the standard method to MP_S. A random algorithm is added as baseline.

Our results indicate that both global and local scaling show very beneficial effects concerning hubness on a wide range of diverse data sets. They are especially effective for data sets of high dimensionality which are most affected by hubness. There is only little impact but no degradation of performance with data sets of low dimensionality. It is our hope that this empirical study will be the starting point of more theoretical work and consideration concerning the connection between hubness, asymmetric neighbor relations, and the benefits of similarity space transformations.

The main evaluation scripts used in this work are publicly available to permit reproduction of our results.[13]

Acknowledgments

This research is supported by the Austrian Research Fund (FWF) (grants P22856-N23, P24095, L511-N15, Z159) and the Vienna Science and Technology Fund WWTF (Audiominer, project number MA09-024). The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry for Transport, Innovation and Technology. We would also like to thank our anonymous reviewers for their comments, which helped to improve this publication substantially.

References

Charu Aggarwal, Alexander Hinneburg, and Daniel Keim. On the surprising behavior of distance metrics in high dimensional space. In *Database Theory - ICDT 2001*, Lecture Notes in Computer Science, pages 420–434. Springer Berlin/Heidelberg, 2001. doi: 10.1007/3-540-44503-X_27.

Jean-Julien Aucouturier and François Pachet. Improving timbre similarity: How high is the sky? *Journal of Negative Results in Speech and Audio Sciences*, 1(1), 2004.

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

13. The scripts can be found at `http://www.ofai.at/~dominik.schnitzer/mp`.

Richard Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

Kristin P. Bennett, Usama Fayyad, and Dan Geiger. Density-based indexing for approximate nearest-neighbor queries. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD)*, KDD '99, pages 233–243, New York, NY, USA, 1999. ACM. ISBN 1-58113-143-7. doi: 10.1145/312129.312236.

Adam Berenzweig. *Anchors and Hubs in Audio-based Music Similarity*. PhD thesis, Columbia University, 2007.

James C. Bezdek and Nihil R. Pal. Some new indexes of cluster validity. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 28(3):301 –315, 1998. ISSN 1083-4419. doi: 10.1109/3477.678624.

Rui Cai, Chao Zhang, Lei Zhang, and Wei-Ying Ma. Scalable music recommendation by search. In *Proceedings of the 15th International Conference on Multimedia*, pages 1065–1074, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-702-5. doi: 10.1145/1291233.1291466.

Michael Casey, Christophe Rhodes, and Malcolm Slaney. Analysis of minimum distances in high-dimensional musical spaces. *IEEE Transactions on Audio, Speech, and Language Processing*, 16 (5):1015–1028, 2008. doi: 10.1109/TASL.2008.925883.

Òscar Celma. *Music Recommendation and Discovery in the Long Tail*. PhD thesis, Universitat Pompeu Fabra, Barcelona, 2008.

Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: A Library for Support Vector Machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Olivier Chapelle, Patrick Haffner, and Vladimir N. Vapnik. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5):1055 –1064, 1999. ISSN 1045-9227. doi: 10.1109/72.788646.

Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923, 1998. doi: 10.1162/089976698300017197.

George Doddington, Walter Liggett, Alvin Martin, Mark Przybocki, and Douglas Reynolds. Sheep, goats, lambs and wolves a statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP-98)*, 1998.

J. Stephen Downie. The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoustical Science and Technology*, 29(4):247–255, 2008.

Levent Ertöz, Michael Steinbach, and Vipin Kumar. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of the Third SIAM International Conference on Data Mining*, San Francisco, CA, 2003.

Andras Ferencz, Erik G. Learned-Miller, and Jitendra Malik. Building a classification cascade for visual identification from one example. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, volume 1, pages 286–293. IEEE, 2005. doi: 10.1109/ICCV.2005.52.

Arthur Flexer, Dominik Schnitzer, Martin Gasser, and Tim Pohle. Combining features reduces hubness in audio similarity. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, 2010.

Damien François, Vincent Wertz, and Michel Verleysen. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, 19:873–886, 2007. ISSN 1041-4347. doi: 10.1109/TKDE.2007.1037.

Andrew Frank and Arthur Asuncion. UCI machine learning repository, 2010. Repository located at: `http://archive.ics.uci.edu/ml`.

Martin Gasser and Arthur Flexer. FM4 SoundPark: Audio-based music recommendation in everyday use. In *Proceedings of the 6th Sound and Music Computing Conference*, pages 23–25, 2009.

Simon Günter and Horst Bunke. Validation indices for graph clustering. *Pattern Recognition Letters*, 24(8):1107 – 1113, 2003. ISSN 0167-8655. doi: DOI:10.1016/S0167-8655(02)00257-X. Graph-based Representations in Pattern Recognition.

Austin Hicklin, Craig Watson, and Brad Ulery. *The Myth of Goats: How Many People have Fingerprints that are Hard to Match?* US Dept. of Commerce, National Institute of Standards and Technology (NIST), 2005.

R.A. Jarvis and Edward A. Patrick. Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers*, 22:1025–1034, 1973. ISSN 0018-9340. doi: 10.1109/T-C.1973.223640.

Herve Jegou, Hedi Harzallah, and Cordelia Schmid. A contextual dissimilarity measure for accurate and efficient image search. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007. doi: 10.1109/CVPR.2007.382970.

Herve Jegou, Cordelia Schmid, Hedi Harzallah, and Jakob Verbeek. Accurate image search using the contextual dissimilarity measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1), 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2008.285.

Wen Jin, Anthony Tung, Jiawei Han, and Wei Wang. Ranking outliers using symmetric neighborhood relationship. In *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, pages 577–593. Springer Berlin/Heidelberg, 2006. doi: 10.1007/11731139_68.

Ioannis Karydis, Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. Looking through the "glass ceiling": A conceptual framework for the problems of spectral similarity. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, 2010.

Elizaveta Levina and Peter J. Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in Neural Information Processing Systems 17*, pages 777–784. MIT Press, Cambridge, MA, 2005.

Beth Logan. Mel frequency cepstral coefficients for music modeling. In *Proceedings of the first International Symposium on Music Information Retrieval (ISMIR)*, Plymouth, MA, 2000.

Michael Mandel and Dan Ellis. Song-level features and support vector machines for music classification. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, London, UK, 2005.

Elias Pampalk. *Computational Models of Music Similarity and Their Application in Music Information Retrieval*. PhD thesis, Vienna University of Technology, 2006.

Elias Pampalk, Simon Dixon, and Gerhard Widmer. On the Evaluation of Perceptual Similarity Measures for Music. In *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx-03)*, pages 7–12, 2003.

Elżbieta Pękalska and Robert P.W. Duin. Classifiers for dissimilarity-based pattern recognition. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 2, pages 12–16. IEEE, 2000. doi: 10.1109/ICPR.2000.906008.

Tim Pohle, Peter Knees, Markus Schedl, and Gerhard Widmer. Automatically adapting the structure of audio similarity spaces. In *Proceedings of the First Learning the Semantics of Audio Signals (LSAS) International Workshop*, Athens, Greece, 2006.

Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11:2487–2531, December 2010. ISSN 1532-4435.

Steven Salzberg. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1:317–328, 1997. ISSN 1384-5810. doi: 10.1023/A:1009752403260.

Markus Schedl. On the Use of Microblogging Posts for Similarity Estimation and Artist Labeling. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, the Netherlands, August 2010.

Gregory Shakhnarovich, Trevor Darrell, and Piotr Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice (Neural Information Processing)*. MIT press, 2006.

Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, volume 17, pages 1601–1608. MIT Press, Cambridge, MA, 2005.