Knowl Inf Syst (2017) 53:479–506
DOI 10.1007/s10115-017-1046-0

REGULAR PAPER

Data-dependent dissimilarity measure: an effective alternative to geometric distance measures

Sunil Aryal[1,2] · Kai Ming Ting[3] · Takashi Washio[4] · Gholamreza Haffari[2]

Received: 9 August 2016 / Revised: 3 March 2017 / Accepted: 17 March 2017 /
Published online: 4 April 2017

Abstract Nearest neighbor search is a core process in many data mining algorithms. Finding reliable closest matches of a test instance is still a challenging task as the effectiveness of many general-purpose distance measures such as ℓ_p-norm decreases as the number of dimensions increases. Their performances vary significantly in different data distributions. This is mainly because they compute the distance between two instances solely based on their geometric positions in the feature space, and data distribution has no influence on the distance measure. This paper presents a simple data-dependent general-purpose dissimilarity measure called 'm_p-dissimilarity'. Rather than relying on geometric distance, it measures the dissimilarity between two instances as a probability mass in a region that encloses the two instances in every dimension. It deems two instances in a sparse region to be more similar than two instances of equal inter-point geometric distance in a dense region. Our empirical results in k-NN classification and content-based multimedia information retrieval tasks show that the proposed m_p-dissimilarity measure produces better task-specific performance than existing widely used general-purpose distance measures such as ℓ_p-norm and cosine distance across

✉ Sunil Aryal
sunil.aryal@federation.edu.au

Kai Ming Ting
kaiming.ting@federation.edu.au

Takashi Washio
washio@ar.sanken.osaka-u.ac.jp

Gholamreza Haffari
gholamreza.haffari@monash.edu

1 School of Engineering and Information Technology, Faculty of Science and Technology, Federation University, Mt. Helen Campus, University Drive, Mount Helen, VIC 3350, Australia

2 Clayton School of Information Technology, Monash University, Clayton, VIC, Australia

3 School of Engineering and Information Technology, Federation University, Gippsland Campus, Churchill, VIC, Australia

4 The Institute of Scientific and Industrial Research, Osaka University, Ibaraki, Japan

a wide range of moderate- to high-dimensional data sets with continuous only, discrete only, and mixed attributes.

Keywords Distance measure · ℓ_p-norm · Cosine distance · m_p-dissimilarity

1 Introduction

In order to make a prediction for a test instance, many data mining algorithms search for its k closest matches or nearest neighbors (k-NNs) in the given training set and make a prediction based on the k-NNs. They use a (dis)similarity or distance measure to find k-NNs. However, finding reliable k-NNs becomes a challenging task as the number of dimensions increases. In high-dimensional space, data distribution becomes sparse which makes the concept of distance meaningless, i.e., all pairs of points are almost equidistant for a wide range of data distributions and distance measures [1,6,12].

Let $D = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(N)}\}$ be a collection of N data instances in an M-dimensional space $\mathcal{X}$. Each instance $\mathbf{x}$ is represented as an M-dimensional vector $\langle x_1, x_2, \cdots, x_M \rangle$. Let $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}$ (where $\mathcal{R}$ is a real domain) be a measure of dissimilarity between two vectors in $\mathcal{X}$. The most common approach of measuring dissimilarity of two data instances $\mathbf{x}$ and $\mathbf{y}$ is based on a geometric model where $\mathcal{X}$ is assumed to be a metric space (which has nice mathematical properties) and $d(\mathbf{x}, \mathbf{y})$ is estimated as their geometric distance in the space. We use distance measures to refer to dissimilarity measures which are metric.

Minkowski distance (aka ℓ_p-norm) [10] is a widely used distance measure. It estimates the dissimilarity of $\mathbf{x}$ and $\mathbf{y}$ by combining their distances in each dimension. Euclidean distance (ℓ_2-norm) is a popular choice of distance function as it intuitively corresponds to the distance defined in the real three-dimensional world. In bag-of-words vector representation of documents, cosine distance (aka angular distance) has been shown to produce more reliable k-NNs than ℓ_2-norm [26]. Cosine distance is proportional to the Euclidean distance of the length normalized vectors (i.e., they are translated in the space to be of unit lengths).

The performance of general-purpose distance measures such as ℓ_p-norm and cosine distance depends on the data distribution: A distance measure that performs well in one distribution may perform poorly in others. This has been suspected to be due to the fact that these distance measures compute the dissimilarity between two instances solely based on their geometric positions in the vector space, and data distribution (positions of other vectors) is not taken into consideration.

Many psychologists have expressed their concerns on the geometric model of dissimilarity measure [17,31] arguing that the judged dissimilarity between two objects is influenced by the context of measurements and other objects in proximity. Krumhansl [17] has suggested a distance density model of dissimilarity measure arguing that two objects in a relatively dense region would be less similar than two objects of equal distance but located in a less dense region. For example, two Chinese individuals will be judged as more similar when compared in Europe (where there are less Chinese and more Caucasian people) than in China (where there are many Chinese people).

In order to understand the influence of data distribution in judged dissimilarity, let us consider an example of a data set with distributions in dimensions i and j as shown in Table 1. In this example, $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ have the same values in dimensions i and j. Their value in dimension i is significantly different from the rest of the instances, but their value is a common value in dimension j (9 out of 10 instances have the same value). In a geometric distance measure such as ℓ_p, because $x_i^{(1)} - x_i^{(2)} = x_j^{(1)} - x_j^{(2)} = 0$, the differences in dimensions i and j have the same contribution in $d(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$. The main concern raised by

Table 1 An example of data distribution in two dimensions

$\mathbf{x}$	...	x_i	x_j	...
$\mathbf{x}^{(1)}$	...	9	1	...
$\mathbf{x}^{(2)}$	...	9	1	...
$\mathbf{x}^{(3)}$	...	2	1	...
$\mathbf{x}^{(4)}$	...	1	1	...
$\mathbf{x}^{(5)}$	...	1	1	...
$\mathbf{x}^{(6)}$	...	1	1	...
$\mathbf{x}^{(7)}$	...	1	1	...
$\mathbf{x}^{(8)}$	...	1	1	...
$\mathbf{x}^{(9)}$	...	1	1	...
$\mathbf{x}^{(10)}$	...	0	5	...

psychologists is that having the same value in dimension j (where probability of the value is high) does not provide the same amount of information about the (dis)similarity between $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ as having the same value in dimension i (where the probability of the value is small). This scenario where many instances have the same value in many dimensions can be very common in high-dimensional spaces as data often lies in a low-dimensional subspace. For example, in bag-of-words vector representation, many entries in document vectors are zero as each document has only a small number of terms from the dictionary.

In this paper, we propose a simple data-dependent general-purpose dissimilarity measure called 'm_p-dissimilarity' in which dissimilarity between two instances is estimated based on data distribution in each dimension. Rather than using the spatial distance in each dimension, m_p-dissimilarity evaluates the dissimilarity between two instances in terms of probability data mass in a region covering the two instances in each dimension. The final dissimilarity between the two instances is estimated by combining dissimilarity in every dimension as in ℓ_p-norm. The intuition behind the proposed dissimilarity measure is that two instances are likely to be dissimilar if there are many instances in-between and around them in many dimensions. Under the proposed data-dependent dissimilarity measure, two instances in a dense region of the distribution are more dissimilar than two instances in a sparse region, even if the two pairs have the same geometric distance, which is prescribed by psychologists.

Our empirical evaluation in k-NN classification and content-based multimedia information retrieval tasks shows that the proposed m_p-dissimilarity measure produces better task-specific performance than existing widely used general-purpose distance measures such as ℓ_p-norm and cosine distance across a wide range of moderate- to high-dimensional data sets with continuous only, discrete only, and mixed attributes.

The rest of the paper is organized as follows. Previous work related to this paper is discussed in Sect. 2. The proposed m_p-dissimilarity is presented in Sect. 3, followed by empirical results in Sect. 4. The relationship of m_p-dissimilarity with ℓ_p-norm after rank transformation of data is discussed in Sect. 5 followed by the related discussion in Sect. 6. Finally, we conclude the paper with conclusions and future work in the last section. From now on, we refer to m_p-dissimilarity and ℓ_p-norm by m_p and ℓ_p, respectively.

2 Related work

In this section, we review some widely used techniques to measure dissimilarity between instances in domains with continuous only, discrete only, and mixed attributes.

2.1 Dissimilarity measures in continuous domain

In continuous domain where each dimension is numeric, i.e., $\forall_i\ x_i \in \mathcal{R}$, the dissimilarity between two M-dimensional vectors $\mathbf{x}$ and $\mathbf{y}$ is primarily based on their positions in the vector space. Minkowski distance of order $p > 0$ (also known as ℓ_p-norm distance) is defined as follows:

$$d_{mink,p}(\mathbf{x}, \mathbf{y}) = \ell_p(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_p = \left(\sum_{i=1}^{M} abs(x_i - y_i)^p\right)^{\frac{1}{p}} \tag{1}$$

where $abs(\cdot)$ is an absolute value.

Euclidean distance ($p = 2$) is a popular choice of distance function as it intuitively corresponds to the distance defined in the real three-dimensional world.

As distance in each dimension has equal influence, ℓ_p is very sensitive to the units and scales of measurement. Min–max normalization ($x'_i = \frac{x_i - \min_i}{\max_i - \min_i}$, where $\min_i$ and $\max_i$ are the minimum and maximum values in dimension i respectively) is commonly used to rescale feature values in the unit range ([0,1]). Even though min–max normalization takes care of scale differences between different dimensions, it does not take care of differences in variance across different dimensions. A unit distance in a dimension with low variance may not be the same as that in a dimension with high variance. In order to ensure the equal variance in each dimension, standard deviation normalization ($x''_i = \frac{x_i}{\sigma_i}$ where σ_i is the standard deviation of values of instances in dimension i) is used in the literature. We call the ℓ_p applied on standard deviation normalized vectors as standardized ℓ_p (s-ℓ_p) i.e., s-$\ell_p(\mathbf{x}, \mathbf{y}) = \ell_p(\mathbf{x}'', \mathbf{y}'')$. Standardized ℓ_p with $p = 2$ (s-ℓ_2) is the simplest variant of Mahalanobis distance [10] where the covariance matrix is a diagonal matrix of variance of values in each dimension.

The Mahalanobis distance [10,22] is defined as follows:

$$d_{mah}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y})} \tag{2}$$

where $\Sigma \in \mathcal{R}^{M \times M}$ is the covariance matrix of D.

Rather than using the inverse of the sample covariance matrix, metric learning literature focus on learning a generalized Mahalanobis distance [5,18,32,33] from D defined as follows:

$$d_{genMah}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \Omega (\mathbf{x} - \mathbf{y})} \tag{3}$$

where $\Omega \in \mathcal{R}^{M \times M}$ is a positive semi-definite matrix.

Since Ω is positive semi-definite, it can be factorized as $\Omega = \Lambda^T \Lambda$ where $\Lambda \in \mathcal{R}^{\omega \times M}$ and ω is a positive integer and $d_{genMah}(\mathbf{x}, \mathbf{y})$ can be written as: $d_{genMah}(\mathbf{x}, \mathbf{y}) = \|\Lambda\mathbf{x} - \Lambda\mathbf{y}\|_2$ [5,18,32]. The generalized Mahalanobis distance is the Euclidean distance of vectors transformed by matrix Λ. The goal of metric learning is to learn a transformation matrix Λ to improve the task-specific performance of the Euclidean distance, subject to some optimality constraints, e.g., similar instances become closer to each other (similarity constraints) and dissimilar instances are separated further apart from each other (dissimilarity constraints). Learning the best Λ requires learning intensive optimization which is expensive in high-dimensional and/or large data sets. Furthermore, Λ is optimized specifically for the given task; and it may not be good for other tasks using the same data set. It is not a general-purpose measure like ℓ_p.

In many high-dimensional problems, data have the same value (0 or any other constant) in many dimensions. This leads to sparseness in data distribution. For example, only a small

proportion of terms in a dictionary appear in each document of a corpus. Many entries of a term vector representing a document are zero. Euclidean distance is not a good choice of distance measure in such problems. The direction of vectors is more important than their lengths. The angular distance measure (aka cosine distance) [10] is a more sensible choice to measure dissimilarity between two documents. The cosine distance between two vectors **x** and **y** is defined as follows [10]:

$$d_{cos}(\mathbf{x}, \mathbf{y}) = 1 - \frac{\sum_{i=1}^{M} x_i \times y_i}{\sqrt{\sum_{i=1}^{M} x_i^2} \times \sqrt{\sum_{i=1}^{M} y_i^2}} \tag{4}$$

Cosine distance is proportional to Euclidean distance when the vectors are length normalized to be of unit lengths which is referred as cosine normalization in the literature. Different term weighting schemes are used to adjust the positions of the document vectors in the space based on the importance of their terms in order to improve the task-specific performance of the cosine distance [19,25]. Cosine distance with term frequency–inverse document frequency (TF-IDF)-based term weighting [25] has been shown to perform well in many text-mining problems such as text categorization, text clustering, and text retrieval tasks.

In both metric learning and term weighting, the focus is to transform data so that task-specific performance of the Euclidean or cosine distance is maximized in the given data set. Some aspects of data distribution is taken into consideration in the transformation in metric learning and in term weighting, but still restricted to be a metric in the transformed space, i.e., dissimilarity is still computed solely based on geometrical positions in the transformed space.

2.2 Dissimilarity measures in discrete domain

In discrete domain, each attribute is a categorical attribute, i.e., $\forall_i\ x_i \in \{v_{i,1}, \cdots, v_{i,u_i}\}$ where $v_{i,j}$ is a label out of u_i possible labels for x_i. A discrete attribute can be ordinal (where there is an ordering of discrete labels $v_{i,1} < v_{i,2} < \cdots < v_{i,u_i}$) or nominal (where there is no ordering of discrete labels).

In order to measure similarity between two labels x_i and y_i for a discrete attribute i, $s(x_i, y_i)$, the simplest overlap approach assigns maximum similarity of 1 if $x_i = y_i$ and minimum similarity of 0 if $x_i \neq y_i$ [7,29]. Other approaches such as occurrence frequency (OF) and inverse occurrence frequency (IOF) [7] estimate $s(x_i, y_i)$ based on the frequencies of x_i and y_i in D if $x_i \neq y_i$ and assign maximum similarity of 1 if $x_i = y_i$ regardless of the frequency. The definition of $s(x_i, y_i)$ based on overlap, OF, and IOF [7] is provided in Table 2.

Lin [20] defined similarity using information theory and suggested a probabilistic measure of similarity in ordinal discrete domain. The similarity between two ordinal labels x_i and y_i is defined as follows:

Table 2 $s(x_i, y_i)$ of two labels x_i and y_i of a nominal attribute i. $f(x_i)$ is the occurrence frequency of label x_i in D; $N = |D|$

$s(x_i, y_i)$	$x_i = y_i$	$x_i \neq y_i$
Overlap	1	0
OF	1	$[1 + \log \frac{N}{f(x_i)} \times \log \frac{N}{f(y_i)}]^{-1}$
IOF	1	$[1 + \log f(x_i) \times \log f(y_i)]^{-1}$

$$s_{lin,ord}(x_i, y_i) = \frac{2 \times \log \sum_{z_i=\min(x_i,y_i)}^{\max(x_i,y_i)} P(z_i)}{\log P(x_i) + \log P(y_i)} \tag{5}$$

where $P(x_i)$ is the probability of x_i and it is estimated from D as $\hat{P}(x_i) = \frac{f(x_i)+1}{N+u_i}$ where $f(x_i)$ is the occurrence frequency of label x_i in D.

Boriah et al. [7] used Lin's information theoretic definition of similarity in nominal discrete domain as follows:

$$s_{lin,nom}(x_i, y_i) = \frac{2 \times \log P(x_i \vee y_i)}{\log P(x_i) + \log P(y_i)} \tag{6}$$

In multivariate discrete domain, dissimilarity[1] between two instances $\mathbf{x}$ and $\mathbf{y}$ using Lin's measure can be estimated as follows [7]:

$$d_{lin}(\mathbf{x}, \mathbf{y}) = 1 - \frac{1}{M} \sum_{i=1}^{M} s_{lin}(x_i, y_i) \tag{7}$$

Boriah et al. [7] have shown that d_{lin} performs better than d_{of} and d_{iof} in discrete domains. Even though measures such as s_{of}, s_{iof} and s_{lin} assign similarity between x_i and y_i in each dimension based on the distribution of labels if $x_i \neq y_i$, they assign the maximum similarity of 1 in the case of $x_i = y_i$ regardless of the distribution of the label.

2.3 Dissimilarity measures in mixed domain

Many real-world applications have both continuous and discrete attributes resulting in mixed domain. In order to measure (dis)similarity between two instances in such a domain, the most commonly used ℓ_p-norm uses the overlap approach to measure dissimilarity between two labels x_i and y_i of a discrete attribute i as $x_i - y_i = 0$ if $x_i = y_i$; and 1 otherwise.

Other approaches include converting attributes into continuous only or discrete only and using (dis)similarity measures designed for continuous or discrete domain. A continuous attribute can be converted into a discrete attribute through discretization [13]. A discrete attribute with u discrete labels can be converted into u continuous attributes as follows: Each discrete label is converted into a binary attribute where 0 represents the absence of the label and 1 represents the presence, and all converted u binary attributes are treated as continuous attributes [13].

3 Data-dependent dissimilarity measure

In order to measure dissimilarity between $\mathbf{x}$ and $\mathbf{y}$, instead of using $abs(x_i - y_i)$ in Eq. 1, we propose to consider the relative positions of $\mathbf{x}$ and $\mathbf{y}$ with respect to the rest of the data distribution in each dimension. The dissimilarity between $\mathbf{x}$ and $\mathbf{y}$ in dimension i can be estimated as the probability data mass in region $R_i(\mathbf{x}, \mathbf{y})$ that encloses $\mathbf{x}$ and $\mathbf{y}$. If there are many instances in $R_i(\mathbf{x}, \mathbf{y})$, $\mathbf{x}$ and $\mathbf{y}$ are likely to be dissimilar in dimension i. Using the same power mean formulation as in ℓ_p-norm, the data-dependent dissimilarity measure based on probability mass is defined as:

$$m_p(\mathbf{x}, \mathbf{y}) = \left(\frac{1}{M} \sum_{i=1}^{M} \left(\frac{|R_i(\mathbf{x}, \mathbf{y})|}{N} \right)^p \right)^{\frac{1}{p}} \tag{8}$$

[1] We used dissimilarity so that it is consistent with other distance or dissimilarity measures.

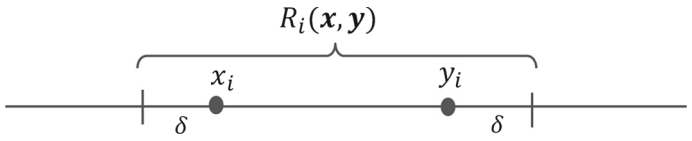

Fig. 1 $R_i(\mathbf{x}, \mathbf{y})$

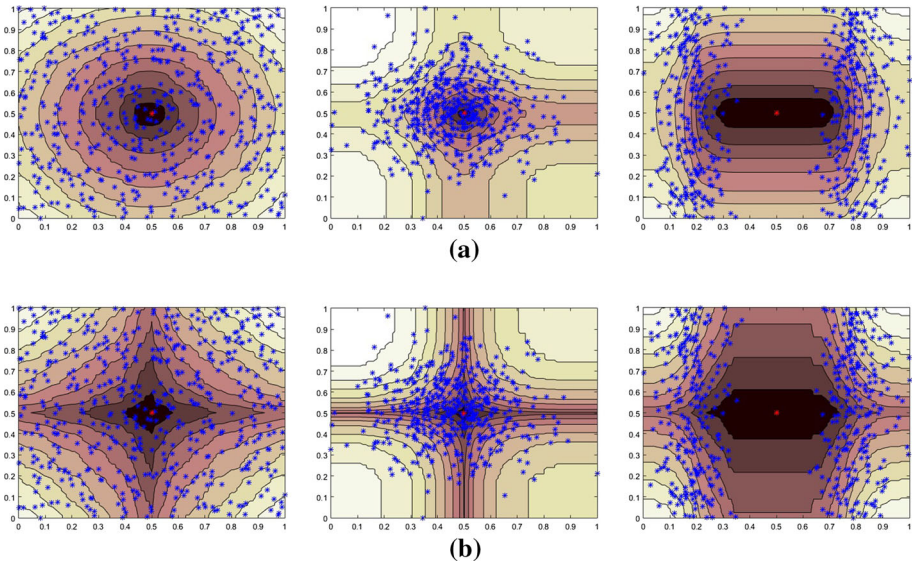

Fig. 2 Contour plots of dissimilarity of points in the space with reference to the centre (0.5, 0.5), based on m_p (with δ for each dimension i set to $\frac{\sigma_i}{2}$ where σ_i is the standard deviation of values of instances in dimension i) in three data distributions (uniform: *left column*, normal: *middle column*, and bimodal: *right column*). The darker the color, the lesser the dissimilarity (a) $p = 2.0$, (b) $p = 0.5$

where $|R_i(\mathbf{x}, \mathbf{y})|$ is the data mass in region $R_i(\mathbf{x}, \mathbf{y}) = [\min(x_i, y_i) - \delta, \max(x_i, y_i) + \delta]$ (i.e., $|R_i(\mathbf{x}, \mathbf{y})| = |\{z_i : \min(x_i, y_i) - \delta \leq z_i \leq \max(x_i, y_i) + \delta\}|$), $\delta \geq 0$, $p > 0$ and N is the total number of instances in D. An example of $R_i(\mathbf{x}, \mathbf{y})$ is shown in Fig. 1.

The region is extended by small $\delta > 0$ beyond x_i and y_i to consider the density distribution around them along with the distribution in-between them. The role of parameter p is similar to that in ℓ_p, i.e., p controls the influence of the dissimilarity in each dimension.

We call the proposed dissimilarity measure $m_p(\mathbf{x}, \mathbf{y})$ as 'm_p-`dissimilarity`'. This measure captures the essence of the distance density model proposed by psychologists [17] which prescribes that two instances in a sparse region are more similar than two instances in a dense region. Although m_p employs the same power mean formulation as ℓ_p, the core calculation is based on probability mass rather than distance. The proposed m_p-dissimilarity has a probabilistic interpretation which is provided in "Appendix 1".

The dissimilarity between a pair of instances using Eq. 8 depends on the distribution of data. Fig. 2 shows the contour plots of m_p-dissimilarity between the point (0.5,0.5) and any other point in the feature space in three different data distributions (uniform, normal and bimodal) for $p = 2.0$ and $p = 0.5$. In contrast, ℓ_p or d_{cos} would produce the same contour in all three distributions. Under uniform distribution and infinite samples, m_p will yield the same result as ℓ_p because the data mass in $R_i(\mathbf{x}, \mathbf{y})$ will be proportional to $abs(x_i - y_i)$. This

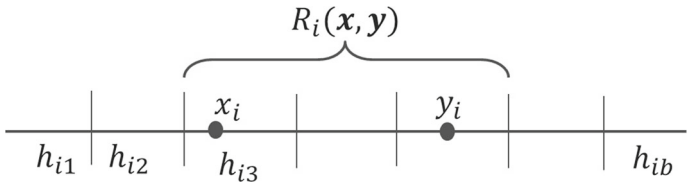

Fig. 3 Defining $R_i(\mathbf{x}, \mathbf{y})$ using bins

is depicted in the two contour plots in the first column in Fig. 2 where they exhibit similar contour plots to those of ℓ_2 and $\ell_{0.5}$.

3.1 Time complexity and efficient approximation

In continuous domains, estimating $m_p(\mathbf{x}, \mathbf{y})$ using Eq. 8 is expensive, especially when either $\mathbf{x}$ or $\mathbf{y}$ is an unseen instance, as it requires a range search in each dimension to estimate $|R_i(\mathbf{x}, \mathbf{y})|$. One-dimensional range search can be done in $O(\log N)$ using binary search trees resulting in the time complexity of $O(M \log N)$ to measure dissimilarity of a pair of instances against $O(M)$ of ℓ_p. It is expensive to compute in large data sets.

Alternatively, $|R_i(\mathbf{x}, \mathbf{y})|$ can be approximated efficiently by using a histogram, i.e., divide the range of real values in each dimension i into b bins $(h_{i1}, h_{i2}, \cdots, h_{ib})$. The number of instances in each bin can be computed in a preprocessing step. When two instances $\mathbf{x}$ and $\mathbf{y}$ are given for dissimilarity measurement, $R_i(\mathbf{x}, \mathbf{y})$ can be computed by using the bins in-between $\mathbf{x}$ and $\mathbf{y}$ as shown in Fig. 3. Even though the approximation using bins does not extend the range exactly by δ beyond x_i and y_i, the bins (where x_i and y_i fall into) provide a reasonable approximation of the distribution around x_i and y_i.

If h_{il} and h_{io} are the two bins in which $\min(x_i, y_i)$ and $\max(x_i, y_i)$ fall, respectively, then $|R_i(\mathbf{x}, \mathbf{y})|$ can be estimated as follows:

$$|R_i(\mathbf{x}, \mathbf{y})| = \sum_{q=l}^{o} |h_{iq}| \tag{9}$$

Note that the binning can be done in two ways: (i) equal width: Each bin is of the same size (bins in dense region have more data mass than those in the sparse region); (ii) equal frequency: Each bin has approximately the same number of instances as much as possible (bins are smaller in dense region than those in the sparse region). The former one is sensitive to outliers. If there is only one instance having significantly different value than others, it may affect the discrimination between the other instances as they all may fall in the same bin, and many bins in the middle will be left empty. Hence, we used the latter approach of binning where each bin has approximately the same number of instances with $b = 100$ using WEKA implementation[2] [13] in this paper. Note that bins in a dimension can have different data mass if many instances have the same values in that dimension making them impossible to split in b bins with the equal data mass.

The preprocessing requires a total of $O(NMb + Mb^2)$ time and $O(Mb^2)$ space complexities. It builds the histogram and the pairwise dissimilarity matrix of bins in each dimension. A histogram of b bins is built for each dimension and the number of instances falling in each bin can be calculated in $O(NMb)$ time. The dissimilarity matrix for $|R_i(\cdot, \cdot)|$ can be precomputed for each pair of bins in each dimension in $O(Mb^2)$ time and stored in $O(Mb^2)$

[2] We used sufficiently large b in order to discriminate instances well.

space. Having preprocessed, the dissimilarity between two instances in each dimension can be done as a table look-up in $O(1)$ time, resulting in $O(M)$ time to measure dissimilarity between a pair of instances, equivalent to those of ℓ_p and d_{cos}.

3.2 Handling discrete attributes

For ordinal discrete attributes, $|R_i(\mathbf{x}, \mathbf{y})|$ can be estimated as follows:

$$|R_i(\mathbf{x}, \mathbf{y})| = \sum_{z_i=\min(x_i,y_i)}^{\max(x_i,y_i)} f(z_i) \tag{10}$$

where $f(z_i)$ is the frequency of discrete label z_i in D.

Unlike d_{lin} that assigns dissimilarity in an ordinal attribute i based on the frequencies of labels if $x_i \neq y_i$ and assigns minimal dissimilarity of 0 regardless of the distribution of labels if $x_i = y_i$, m_p assigns dissimilarity based on the frequency of the label even in the case of $x_i = y_i$.

For nominal discrete attributes, $|R_i(\mathbf{x}, \mathbf{y})|$ can be estimated as follows:

$$|R_i(\mathbf{x}, \mathbf{y})| = \begin{cases} f(x_i) & \text{if } x_i = y_i \\ N & \text{otherwise} \end{cases} \tag{11}$$

It is interesting to note the difference between m_p and the existing dissimilarity measures for nominal domains such as d_{lin}, d_{of} and d_{iof} [7]. For a nominal attribute i, they use the frequency of labels if two instances have different labels ($x_i \neq y_i$), and assign the maximal similarity of 1 (or minimal dissimilarity of 0) if $x_i = y_i$. In contrast, m_p uses the opposite approach and uses the frequency of the label if $x_i = y_i$ and assigns the maximal dissimilarity of 1 otherwise. In the case of $x_i = y_i$, existing measures assign maximal similarity of 1 without considering the distribution of the label. It might be the case that all the other instances have the same label, and there is no discrimination between instances w.r.t the attribute.

The frequency of each discrete label can be computed in a preprocessing step which requires $O(NM)$ time and $O(Mu)$ (where u is the average number of discrete labels per dimension) space.

3.3 Dissimilarity measure in bag-of-words vector representation

In the case of bag-of-words (bow) [26] vector representations, each component of a vector represents frequency of a feature (term in documents or a visual descriptor in images). Given any two vectors $\mathbf{x}$ and $\mathbf{y}$, many features have zero frequency i.e., $x_i = y_i = 0$ for many dimensions, because a document contains only a small proportion of words in the dictionary. Since the absence of a feature in both the instances does not provide any information about the (dis)similarity of $\mathbf{x}$ and $\mathbf{y}$, those features should be ignored. Hence, in the bow vector representation, m_p-dissimilarity of $\mathbf{x}$ and $\mathbf{y}$ is estimated using only those features that occur in either of $\mathbf{x}$ or $\mathbf{y}$ as follows:

$$m_p(\mathbf{x}, \mathbf{y}) = \left(\frac{1}{|F_{\mathbf{x},\mathbf{y}}|} \sum_{i \in F_{\mathbf{x},\mathbf{y}}} \left(\frac{|R_i(\mathbf{x}, \mathbf{y})|}{N} \right)^p \right)^{\frac{1}{p}} \tag{12}$$

where $F_{\mathbf{x}}$ is the set of features that occur in $\mathbf{x}$ (i.e., $F_{\mathbf{x}} = \{i : x_i > 0\}$), and $|F_{\mathbf{x},\mathbf{y}}| = |F_{\mathbf{x}} \cup F_{\mathbf{y}}|$ is the normalization term employed to account for different numbers of features used for measuring dissimilarity of any two instances.

It is important to ignore those features which have zero frequency in both instances ($x_i = y_i = 0$); otherwise, m_p would assign large dissimilarity w.r.t those features as many instances in the data set will have 0 values. This is not an issue for ℓ_p because it assigns 0 dissimilarity when $x_i = y_i = 0$.

3.4 Distinguishing properties of m_p

3.4.1 Data-dependent self-dissimilarity

The distinguishing characteristic of m_p against the geometry-based (ℓ_p and d_{cos}) and probabilistic (d_{lin}) dissimilarity measures discussed in Sect. 2 is the self-dissimilarity. The self-dissimilarity of m_p is not zero, and it ranges from the minimum of $\frac{1}{N}$ to the maximum of 1, depending on the data distribution in each dimension. In contrast, $\ell_p(\mathbf{x}, \mathbf{x}) = d_{cos}(\mathbf{x}, \mathbf{x}) = d_{lin}(\mathbf{x}, \mathbf{x}) = 0$ irrespective of the data distribution. Because of the data-dependent self-dissimilarity, m_p is non-metric.

The approximation of $|R_i(\mathbf{x}, \mathbf{y})|$ using equal-frequency bins will yield a nonzero constant self-dissimilarity for m_p only if each bin has the same number of instances in each dimension. This is often not be possible because there are duplicate values in many instances and this occurs in many dimensions. This is a common characteristic of many high-dimensional data sets because data often lie in a low-dimensional subspace. As a result, bins often have different numbers of instances resulting in data-dependent self-dissimilarity. The advantage of data-dependent self-dissimilarity of m_p over data-independent self-dissimilarity of existing measures is discussed in Sect. 5.

In discrete domains, unlike d_{lin}, d_{of} and d_{iof} based measures that use the probabilities of categorical labels only in the case of different labels, m_p uses the probability of the label even in the case of matching labels—data-dependent self-dissimilarity in action.

3.4.2 m_p is equivalent to ℓ_p only under uniform distribution

Under uniform distribution and infinite data, m_p is equivalent to ℓ_p as the data mass in the range is proportional to its length. This is the only condition under which m_p—a data-dependent measure—is equivalent to ℓ_p—a geometric model-based measure.

3.4.3 Robust to scale, units of measurement and outliers

As m_p is based on counts and does not use the feature values in the dissimilarity measure directly, it is robust to scale and units of measurements in continuous domains. It does not require preprocessing of data to address the scaling issue (min–max normalization) or difference in variance across different dimensions (standard deviation normalization). In many real-world applications, different properties of data may have been represented or measured in different scales (e.g., income is represented in dollars and age in normal integer scale: One unit difference is not the same in these two attributes). This can be the case in high-dimensional problems where different properties are measured by different sensors. Also for the same reason (i.e., based on the count and not the actual feature values), m_p is less sensitive to outliers. In the case of ℓ_p, outliers can have an adverse impact as they might change variance significantly.

4 Empirical evaluation

This section presents the results of experiments conducted to demonstrate that simply by replacing the geometric distance with the probability mass in each dimension, m_p produces better task-specific performances than ℓ_p and d_{cos} across a wide range of data sets. We have evaluated the performance of m_p against the general-purpose dissimilarity measures of Minkowski distance (ℓ_p), Minkowski distance after standard deviation normalization (s-ℓ_p), Cosine distance (d_{cos}) and Lin's probabilistic measure (d_{lin}) in k-nearest neighbor (k-NN) classification and content-based multimedia information retrieval (CBMIR) tasks. We used two settings of $p \in \{0.5, 2.0\}$ in ℓ_p, s-ℓ_p and m_p resulting in eight measures: d_{cos}, d_{lin}, $\ell_{0.5}$, ℓ_2, s-$\ell_{0.5}$, s-ℓ_2, $m_{0.5}$ and m_2. All dissimilarity measures and algorithms were implemented in Java using the WEKA platform [13].

We used moderately high to high-dimensional ($M \geq 20$) data sets from different application areas such as text, image, music, characters and digits recognition, medical and biology, games. In text collections, documents were represented by TF-IDF [25] weighted 'bag-of-words' [26] vectors. Feature values in each dimension in all other non-text data sets were normalized to be in the unit range. For d_{lin}, continuous attributes were converted into ordinal attributes using discretization as in the case of m_p.

The properties of the data sets are provided in Table 3. NG20, R52, R8, Webkb were from ([8])[3]; Ohscal, Wap, New3s and Fbis were from [14][4]; Caltech256 (sift bag-of-words features) from [30][5]; Corel and Gtzan were from [34]; HBA was from [2] and the rest of the other data sets were from UCI [4][6] and WEKA [13][7].

We discuss the experimental setups and results in k-NN classification and content-based multimedia information retrieval (CBMIR) tasks in the next two subsections.

4.1 k-NN classification

In the k-NN classification context, in order to predict a class label for a test instance $\mathbf{x}$, its k nearest neighbors were searched in the training set using all eight dissimilarity measures and the most frequent label in k-NNs was predicted as the class label for the test instance. All classification experiments were conducted using a tenfold cross-validation: 10 train-and-test trials using 90% of the given data set for training and 10% for testing. We set k to a commonly used value of 5 (i.e., $k = 5$). The average classification accuracy (%) over a tenfold cross-validation was reported. The accuracies of two algorithms were considered to be significantly different if their confidence intervals (based on two standard errors over the tenfold cross-validation) did not overlap. The average classification accuracies over the tenfold cross-validation of the eight dissimilarity measures in all data sets are provided in Table 4.

Out of 30 data sets, $m_{0.5}$ and m_2 produced the best result or equivalent to the best result in 23 and 16 data sets, respectively. Either $m_{0.5}$ or m_2 produced significantly better classification accuracy than any other contenders in the New3s, Ohscal, Wap, R52, NG20, R8, Webkb, Caltech, Corel, Connect-4 and Hypothyroid data sets. The summarized results in the last two rows in Table 4 show that both $m_{0.5}$ and m_2 produced consistently top or near top results across different data sets. $m_{0.5}$ and m_2 have average ranking of 1.97 and 2.37, respectively; whereas

[3] http://web.ist.utl.pt/acardoso/datasets/.

[4] http://www.cs.waikato.ac.nz/ml/weka/datasets.html.

[5] http://homes.esat.kuleuven.be/~tuytelaa/unsup_features.html.

[6] https://archive.ics.uci.edu/ml/datasets.html.

[7] Available with WEKA software http://www.cs.waikato.ac.nz/ml/weka/.

Table 3 Data sets used to compare the performance of m_p with other distance or dissimilarity measures. The number of nominal attributes (M_{nom}) is provided in bracket along with the total number of dimensions (M) and c is the number of classes in a data set

Name	N	M	(M_{nom})	c	Application area
New3s	9558	26,832	(0)	44	Text (TREC Collection)
Ohscal	11,162	11,465	(0)	10	Text (Ohsumed patients' information)
Arcene	200	10,000	(0)	2	Bioinformatics (Cancer)
Wap	1560	8460	(0)	20	Text (Yahoo web pages)
R52	9100	7369	(0)	52	Text (Reuters Collection)
NG20	18,821	5489	(0)	20	Text (20 Newsgroup)
Gisette	7000	5000	(0)	2	Digits Recognition
R8	7674	3497	(0)	8	Text (Reuters Collection)
Fbis	2463	2000	(0)	17	Text (TREC Collection)
Webkb	4199	1816	(0)	4	Text (University web pages)
Ads	3279	1558	(1555)	2	Internet Advertisements
Caltech	30,607	1000	(0)	257	Image
Mnist	70,000	784	(0)	10	Digits Recognition
Mfeat	2000	649	(0)	10	Digits Recognition
Isolet	7797	617	(0)	26	Spoken letters
Madelon	2600	500	(0)	2	Artificial data
Arrhythmia	452	279	(73)	2	Medical (Cardiac Arrhythmia)
Gtzan	1000	230	(0)	10	Music
Ismis	12,495	191	(0)	6	Music
Hba	1500	187	(0)	15	Music
Musk2	6598	166	(0)	2	Chemoinformatics
Corel	10,000	67	(0)	100	Image
Splice	3190	60	(60)	3	Bioinformatics (DNA)
Miniboone	129,596	50	(0)	2	Physics (particles)
Connect-4	67,557	42	(42)	3	Game (Connect-4)
Annealing	898	38	(32)	6	Steel annealing
Satellite	6435	36	(0)	7	Satellite Image
Chess	3196	36	(36)	2	Game
Hypothyroid	3772	29	(22)	4	Medical (Thyroid)
Credit-g	1000	20	(13)	2	Finance (Credit risks)

the average rank of the closest contender d_{cos} is 3.30. Table 5 provides the summarized result in terms of the win:loss:draw counts of $m_{0.5}$ and m_2 against the other six contenders using confidence interval based on the two standard errors in the tenfold cross-validation (standard errors are provided in Table 10 in "Appendix 3"). It shows that both $m_{0.5}$ and m_2 had significantly more wins than losses against all other contenders.

Note that in data sets with nominal attributes only (e.g., Connect-4, Chess, and Splice), d_{cos}, ℓ_p and s-ℓ_p produced exactly the same results because they are effectively the same measure. Because of the one-of-all transformation, all the vectors are of the same length of M (as each instance has exactly M 1s) in which case d_{cos} is proportional to ℓ_2. Since the difference in each dimension is either 0 or 1, the parameter p is meaningless.

Table 4 Average accuracy of 5-NN classification over a tenfold cross-validation. The average accuracy and average rank of measures in 30 data sets are included in the last two rows

Data set	d_{cos}	$\ell_{0.5}$	ℓ_2	s-$\ell_{0.5}$	s-ℓ_2	d_{lin}	$m_{0.5}$	m_2
New3s	76.28	24.35	64.78	27.53	36.72	1.97	**80.11***	79.51
Ohscal	60.30	19.57	42.64	15.51	26.62	6.84	**73.22***	71.94
Arcene	82.00	84.00	83.50	83.50	80.00	82.00	84.00	79.50
Wap	73.53	22.95	35.06	19.62	25.90	29.42	82.82*	82.50*
R52	87.44	74.79	76.18	72.09	62.74	0.97	**90.07***	88.63
NG20	83.44	27.05	58.07	27.50	58.37	4.82	**84.63***	81.80
Gisette	97.76*	94.59	96.50	95.16	95.77	92.30	96.77	97.73*
R8	90.36	81.89	79.59	80.02	70.11	51.90	**94.94***	93.72
Fbis	77.91*	48.18	70.40	49.61	60.74	56.23	79.21*	78.85*
Webkb	73.40	51.28	63.85	51.34	62.47	47.30	**85.23***	84.31
Ads	96.43	96.49	96.46	97.26*	97.07*	94.54	96.59*	97.04*
Caltech	11.40	2.90	8.46	3.49	8.74	1.52	13.83	**14.68***
Mnist	**97.66***	95.62	97.19	95.49	94.79	41.58	95.77	97.23
Mfeat	98.00	98.15	98.20	98.20	98.15	97.78	97.85	98.20
Isolet	88.37*	83.71	89.16*	83.42	87.51	81.49	79.68	82.42
Madelon	57.27	60.92*	56.88	60.23*	53.92	58.65	59.23*	55.00
Arrhythmia	63.93	64.83	63.93	65.48	68.15*	71.00*	71.90*	69.88*
Gtzan	70.90*	65.00	70.40*	63.10	65.20	70.40*	72.00*	68.80
Ismis	94.53	94.35	94.41	94.10	94.14	95.42*	95.54*	94.48
Hba	50.20	59.07	52.00	59.40	53.67	65.27*	67.07*	60.73
Musk2	96.45	95.35	96.62	95.18	**97.03***	95.01	95.01	95.47
Corel	24.59	35.66	23.68	36.82	28.80	37.67	**39.76***	35.30
Splice	78.21	78.21	78.21	78.21	78.21	85.52*	84.64*	83.17
Miniboone	92.65	93.03*	92.63	92.84	92.89	76.76	92.77	92.94*
Connect-4	74.85	74.85	74.85	74.85	74.85	30.29	76.62	**77.11***
Annealing	84.65	87.88	85.09	88.65	85.53	89.76*	89.64*	85.87
Satellite	84.86	90.68*	90.97*	90.54*	91.03*	90.65*	90.97*	90.85*
Chess	96.24*	96.24*	96.24*	96.24*	96.24*	96.31*	93.52	95.87*
Hypothyroid	93.43	93.72	93.45	94.30	94.25	94.94	**95.71***	94.19
Credit-g	72.40	72.20	72.40	71.60	72.80	70.80	73.20	71.80
Avg. Acc.	77.65	68.92	73.39	68.71	70.41	60.64	81.08	79.98
Avg. Rank	3.30	4.07	3.60	3.97	3.83	4.67	1.97	2.37

Boldface represents a measure which has significantly better performance than all other competitors and represents the best or equivalent to the best performance (it is not used when all the measures produced the best or equivalent to the best results, e.g., Arcene, Mfeat, and Credit-g)

All eight measures had run time in the same order of magnitude. For example, predicting class labels for instances in one fold of train-and-test in NG20 took 21,458 seconds for m_2 and 26,484 seconds for $m_{0.5}$ in comparison to 16,296 (d_{cos}), 28,168 ($\ell_{0.5}$), 24,380 (ℓ_2), 29,210 (s-$\ell_{0.5}$), 25,944 (s-ℓ_2) and 20,515 (d_{lin}) seconds. In Corel, m_2 and $m_{0.5}$ took 37 and 47 seconds whereas d_{cos} took 22 s followed by 32 (ℓ_2), 45 ($\ell_{0.5}$), 47 (s-ℓ_2), 59 (s-$\ell_{0.5}$) and 90 (d_{lin}) seconds.

Table 5 Win:loss:draw counts of $m_{0.5}$ and m_2 w.r.t other measures in 5-NN classification

	$m_{0.5}$	m_2
d_{cos}	18:5:7	16:5:9
$\ell_{0.5}$	18:4:8	17:3:10
ℓ_2	19:4:7	17:2:11
s-$\ell_{0.5}$	19:3:8	16:4:10
s-ℓ_2	21:4:5	16:2:12
d_{lin}	17:2:11	16:7:7

Table 6 Average P@10 over N queries. The average P@10 and average rank of measures in 10 data sets are included in the last two rows

Data set	d_{cos}	$\ell_{0.5}$	ℓ_2	s-$\ell_{0.5}$	s-ℓ_2	d_{lin}	$m_{0.5}$	m_2
New3s	0.66	0.16	0.47	0.14	0.15	0.03	**0.69**$*$	0.68
Ohscal	0.48	0.17	0.27	0.15	0.15	0.10	**0.61**$*$	0.59
Wap	0.64	0.18	0.24	0.16	0.16	0.20	0.73$*$	0.72$*$
R52	0.81	0.69	0.70	0.66	0.59	0.33	**0.85**$*$	0.83
NG20	**0.71**$*$	0.19	0.42	0.19	0.40	0.06	0.697	0.65
Fbis	0.68$*$	0.36	0.57	0.34	0.45	0.41	0.68$*$	0.67
Caltech	0.08	0.02	0.06	0.03	0.06	0.01	0.09	**0.10**$*$
Gtzan	0.53$*$	0.49	0.53$*$	0.48	0.49	0.53$*$	0.54$*$	0.51
Hba	0.37	0.44	0.38	0.45	0.40	0.50$*$	0.51$*$	0.46
Corel	0.16	0.24	0.16	0.25	0.19	0.253	**0.27**$*$	0.24
Avg. P@10	0.51	0.29	0.38	0.29	0.30	0.24	0.57	0.55
Avg. Rank	3.20	5.30	4.40	5.80	5.80	5.50	1.20	2.70

Boldface represents a measure which has significantly better performance than all other competitors and $*$ represents the best or equivalent to the best performance

4.2 Content-based multimedia information retrieval (CBMIR)

Given a query instance **q** for a retrieval task, all the instances in a data set were ranked in ascending order of their dissimilarity to **q** based on a dissimilarity measure; and the first k instances were presented as the relevant instances to **q**. For performance evaluation, an instance was considered to be relevant to **q** if they have the same category label. A good information retrieval system returns relevant instances at the top. Hence, the precision in the top 10 (P@10) retrieved results was used as the performance measure. The same process was repeated for each instance in a data set as a query and the rest of the instances were ranked. The average P@10 of N queries was reported. For information retrieval task, we used 10 data sets with 10 or more classes from multimedia (text, music and image) applications: New3s, Ohscal, Wap, R52, NG20, Fbis, Caltech, Gtzan, Hba and Corel. The average P@10 of d_{cos}, $\ell_{0.5}$, ℓ_2, s-$\ell_{0.5}$, s-ℓ_2, d_{lin}, $m_{0.5}$ and m_2 are provided in Table 6.

Table 7 presents the summarized result in terms of the win:loss:draw counts of $m_{0.5}$ and m_2 using confidence interval based on the two standard errors over N queries (standard errors are provided in Table 11 in "Appendix 3"). It shows that both $m_{0.5}$ and m_2 produced significantly better retrieval results than the other six contenders in many data sets: $m_{0.5}$ had only 1 loss and between 7 and 10 wins; m_2 has at least 7 wins and at most 3 losses. The

Table 7 Win:loss:draw counts of $m_{0.5}$ and m_2 w.r.t other measures in CBMIR

	$m_{0.5}$	m_2
d_{cos}	7:1:2	7:2:1
$\ell_{0.5}$	10:0:0	7:1:2
ℓ_2	9:0:1	9:1:0
$s\text{-}\ell_{0.5}$	10:0:0	8:1:1
$s\text{-}\ell_2$	10:0:0	9:0:1
d_{lin}	8:0:2	7:3:0

detailed result in Table 6 shows that, out of 10 data sets used, $m_{0.5}$ and m_2 produced the best result or equivalent to the best result in 9 and 6 data sets, respectively. They have the average ranking of 1.20 and 2.70, respectively whereas the closest contender d_{cos} has an average ranking of 3.2.

5 Relation to ℓ_p with rank transformation

In the first glance, it appears that m_p (Eq. 8 with $\delta = 0$) is equivalent to ℓ_p with rank transformation [9] in continuous domains because they both measure dissimilarity based on the number of instances in-between the two instances under measurement. In rank transformation [9], instances in each dimension are ranked in ascending order with the smallest value having ranked 1, the second smallest value having ranked 2, and so on. The values of instances are then replaced by their ranks. If there are $n < N$ instances which have the same value and the value has rank r, then all instances are assigned the same rank r; and the next available rank is $r + n$ (i.e, the minimum rank is assigned in the case of tie)[8].

The distance between two instances in each dimension after the rank transformation as discussed above can be defined as: $abs(rank(x_i) - rank(y_i)) = |\{z_i : \min(x_i, y_i) \leq z_i < \max(x_i, y_i)\}|$. In m_p (with $\delta = 0$) using the implementation based on the range search, $|R_i(x_i, y_i)| = |\{z_i : \min(x_i, y_i) \leq z_i \leq \max(x_i, y_i)\}|$[9].

These two formulations are equivalent only if all values in dimension i are distinct, i.e., $|R_i(x_i, y_i)| = abs(rank(x_i) - rank(y_i)) + 1$. They are different when there are duplicate values; and the degree of difference is proportional to the number of duplicates.

It is interesting to note that the self-dissimilarity of x_i if there are duplicate x_i: $abs(rank(x_i) - rank(x_i)) = 0$ versus $|R_i(x_i, x_i)| = f(x_i)$ where $f(x_i)$ is the frequency of x_i. Even though rank difference between x_i and y_i is density (data) dependent when $x_i \neq y_i$ (i.e., the rank difference between x_i and y_i is larger in denser region than in sparse region even if the geometric distance is the same), it is zero irrespective of the distribution when $x_i = y_i$. In the extreme case where all the instances have the same value in dimension i, the self-dissimilarity is 1 (maximum) in the case of m_p, whereas the self-dissimilarity of ℓ_p after rank transformation is 0 (minimum). Often in high-dimensional real-world problems, many instances can have the same value in many dimensions, e.g., many documents in a collection can have the same occurrence frequency of a term; or different individuals can have the same age, etc.

We have compared the performances of m_p and ℓ_p with rank transformation (ℓ_p^{rank}) in the k-NN classification task using data sets with continuous attributes only (as rank trans-

[8] Another approach of assigning rank in the case of tie is to assign the average rank, i.e., $\frac{r+(r+1)+\cdots+(r+n)}{n}$.

[9] We used the implementation based on the range search and not the approximation using binning in order to have similar formulation as ℓ_p with rank transformation.

Table 8 The average accuracy of k-NN ($k = 5$) classification in a tenfold cross-validation. The distinct values statistic α is provided in the second column

Data set	α	ℓ_2^{rank}	m_2	$\ell_{0.5}^{rank}$	$m_{0.5}$
Hba	0.973	60.40	60.73	66.67	66.93
Gtzan	0.966	68.40	68.50	71.50	71.50
Arcene	0.378	84.50	79.50	80.00	84.00
Mfeat	0.320	97.95	98.20	97.80	97.90
Madelon	0.054	55.08	55.23	59.23	59.73
Satellite	0.011	90.72	90.80	90.69	90.94
Fbis	0.005	64.60	**78.85**	59.40	**79.21**
Wap	0.002	26.54	**82.82**	25.19	**82.50**
Webkb	0.002	61.28	**84.31**	59.18	**85.23**
R8	0.001	85.80	**93.72**	87.35	**94.94**

Boldface represents significantly better performance than the corresponding contender

formation is applicable only in continuous domains). Both ℓ_p^{rank} and m_p (since the efficient approximation of $R_i(\cdot, \cdot)$ as discussed in Sect. 3.1 is not used) have high time complexities. Estimating $|R_i(\cdot, \cdot)|$ in m_p and computing the rank of an unseen value of a test instance in ℓ_p^{rank} in each dimension requires $O(\log N)$ time using binary search resulting in the total time complexity of measuring dissimilarity of a pair instances to be $O(M \log N)$ which is very expensive in large data sets. We only managed to get a tenfold cross-validation of k-NN classification completed in 24 h in ten relatively small data sets only: Hba, Gtzan, Arcene, Mfeat, Madelon, Satellite, Fbis, Wap, Webkb, and R8.

In order to provide an idea about the number of duplicate values per dimension in a data set, the factor of distinct values averaged over all dimensions, i.e., α, is calculated as:

$$\alpha = \frac{1}{M} \sum_{i=1}^{M} \frac{w_i}{N} \tag{13}$$

where w_i is the number of distinct values in dimension i. $\alpha = 1$ indicates that the data set has unique values in all dimensions (no duplicate at all) and $\alpha = \frac{1}{N}$ indicates that all instances have the same value in each and every dimension.

The average accuracies of 5-NN classification over a tenfold cross-validation using ℓ_2^{rank}, $\ell_{0.5}^{rank}$, m_2 and $m_{0.5}$ are provided in Table 8. Based on two standard error confidence interval significance test, ℓ_2^{rank} & m_2 and $\ell_{0.5}^{rank}$ & $m_{0.5}$ produced similar results in Hba, Gtzan, Arcene, MFeat, Madelon and Satellite; but both m_2 and $m_{0.5}$ produced significantly better accuracies than ℓ_2^{rank} and $\ell_{0.5}^{rank}$ in Fbis, Wap, Webkb, and R8. These results show that m_p performs better than ℓ_p^{rank} in the case where many instances have the same values (i.e., there are only a very few distinct values) in many dimensions.

In order to further demonstrate this difference, we conducted experiments with the Hba and Gtzan data sets (having large α) by increasing the number of duplicate values in many dimensions. The range of values in dimension i was divided into 10 equal-width bins represented by bin id $1, 2, \cdots, 10$ and an instance's value was replaced by the id of the bin in which the instance falls into, resulting in many duplicate values in dimension i. In order to control the number of dimensions with duplicate values, we introduced a parameter a that determines the proportion of attributes to be converted into bins, i.e., $a = 0$ indicates that no attribute was converted into bins (i.e., values in all attributes were left as they were and no duplicate values were introduced) and $a = 1.0$ indicates that all attributes were converted into bins (i.e., many instances have duplicate values in all dimensions). The k-NN ($k = 5$)

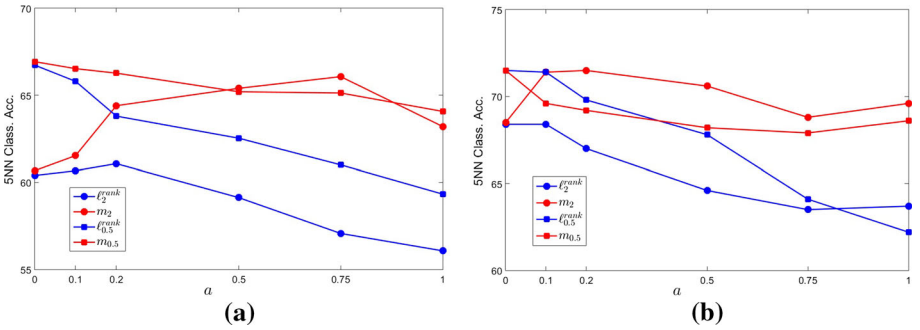

Fig. 4 5-NN classification accuracies of ℓ_2^{rank}, $\ell_{0.5}^{rank}$, m_2 and $m_{0.5}$ for different values of a

Table 9 The distinct values statistic α for different values of a

Data set	$a = 0$	$a = 0.1$	$a = 0.2$	$a = 0.5$	$a = 0.75$	$a = 1.0$
Hba	0.973	0.881	0.783	0.485	0.241	0.006
Gtzan	0.966	0.870	0.774	0.486	0.246	0.009

classification accuracies of ℓ_2^{rank}, $\ell_{0.5}^{rank}$, m_2 and $m_{0.5}$ in the Hba and Gtzan data sets with $a = 0, 0.1, 0.2, 0.5, 0.75$ and 1.0 are shown in Fig. 4 and corresponding α values are provided in Table 9.

Figure 4 shows that there is a significant difference between the classification accuracies of m_2 and $m_{0.5}$ in compare to those of ℓ_2^{rank} and $\ell_{0.5}^{rank}$ for $a \geq 0.75$ in both the Hba and Gtzan data sets. This indicates that m_p can provide more reliable nearest neighbors than ℓ_p^{rank} if many instances have duplicate values in many dimensions. This superior performance of m_p over ℓ_p^{rank} is primarily due to the data-dependent self-dissimilarity.

Furthermore, the rank transformation is possible in continuous domains only. In contrast, m_p not only can apply to both continuous and discrete domains, but has a seamless treatment of mixed attribute type domains.

6 Discussion

In a high-dimensional space, the most widely used Euclidean distance (ℓ_2-norm) becomes ineffective. Many researchers have argued that it is due to the 'concentration' effect of ℓ_p, i.e., pairwise distances become almost equal or similar and the contrast between the nearest and farthest instances diminishes [1,6,12]. Let $dmax(\mathbf{x}, d)$ and $dmin(\mathbf{x}, d)$ be the dissimilarity of $\mathbf{x}$ to its farthest and nearest neighbors in D using dissimilarity measure d, respectively. For a given instance, the distance between the nearest and farthest instances does not increase as fast as the distance to the nearest instance for many distributions [6], i.e., the 'relative contrast' $\left(\frac{dmax(\mathbf{x},\ell_p)-dmin(\mathbf{x},\ell_p)}{dmin(\mathbf{x},\ell_p)}\right)$ vanishes as the number of dimensions increases.

In our investigation, we observed that m_p is more concentrated than ℓ_p and d_{cos}, i.e., the relative contrast of m_p is smaller than that of ℓ_p and d_{cos}. Despite having a higher concentration effect, m_p provides more reliable nearest neighbors than ℓ_p and d_{cos} in many data sets, particularly in high-dimensional problems (see the experimental results in Sects. 4.1 and 4.2). This indicates that the negative impact of the concentration phenomenon in practice may not be as severe as it is thought theoretically in the literature. This finding is consistent

with that suggested by François et al. [12]. The detailed empirical result of the phenomenon of concentration of m_2, ℓ_2 and d_{cos} is provided in the "Concentration" section of "Appendix 2".

Another issue of distance measures in high-dimensional spaces discussed in the literature is "hubness" [24]. Let $N_k(\mathbf{y})$ be the set of k nearest neighbors of $\mathbf{y}$; and k-occurrences of $\mathbf{x}$, $O_k(\mathbf{x}) = |\{\mathbf{y} : \mathbf{x} \in N_k(\mathbf{y})\}|$, be the number of other instances in the given data set where $\mathbf{x}$ is one of their k nearest neighbors. As the number of dimensions increases, the distribution of $O_k(\mathbf{x})$ becomes considerably skewed (i.e., there are many instances with zero or small O_k and only a few instances have large O_k) for many widely used distance measures [24]. The instances with large $O_k(\cdot)$ are considered as "hubs," i.e., the popular nearest neighbors. Hubness becomes prominent in high-dimensional space, and it affects the performance of k-NN based algorithms. For example, if $\mathbf{x}$ is a hub, it appears in the k-NN sets of many test instances and contributes in the prediction decisions, but it may not be relevant to make predictions for all of those test instances.

We observed that the hubness phenomenon in m_p is not as severe as in the case of ℓ_p and d_{cos} when the number of dimensions is increased particularly in non-uniform distributions. This may contribute to the superior performance of m_p over ℓ_p and d_{cos}. The detailed empirical result of the phenomenon of hubness of m_2, ℓ_2 and d_{cos} is provided in the "Hubness" section of "Appendix 2".

In order to circumvent the high-dimensionality issue, dimensionality reduction [11] techniques are used before using distance measures. In continuous domain, Principal Component Analysis (PCA) [16] is commonly used to project data into a lower-dimensional space defined by principal components with high variance. The principal components are computed by the eigen decomposition of covariance or correlation matrix which is computationally expensive in the case of large M and N. It relies on variance of data in each dimension which may not be enough to capture the characteristics of local data distribution. As it selects the dimensions with high variance, we may lose differences between instances in the dimensions with low variance.

In a nutshell, the main purpose of PCA is dimensionality reduction that enables an application to high-dimensional data sets; and it usually does not improve predictive accuracy. This is exactly what we observed in the 5-NN classification task. For example, 5-NN classification accuracies of d_{cos} and ℓ_2 were increased in Corel and Hba but that of $\ell_{0.5}$ was decreased in both data sets. Similarly, the classification accuracies of all three measures decreased significantly in Mnist and R52. In general, m_2 and $m_{0.5}$ in the original space (without dimensionality reduction) produced better and consistent results across different data sets. The detailed results of this comparison are provided in Table 12 in "Appendix 4".

Note that PCA changes the distribution of data to maximize the variance (which is defined by inter-point distances). Thus, it does not make sense to apply PCA when using m_p.

Different data-dependent distance metric adaptation techniques are discussed in the literature to improve task-specific performance of distance measures in a given data set. Weighted Minkowski distance [10] assigns weight to the distance in each dimension based on the observed data. Note that standardized Euclidean distance (s-ℓ_2) is a simple weighted Euclidean distance where the distance in each dimension is weighted by the inverse of data variance in that dimension. Assigning weights more intelligently requires some learning or optimization. In transductive learning, Lundell and Ventura [21] corrected the Euclidean distance between two instances based on meta-clustering which itself relies on pairwise Euclidean distances and can be computationally expensive in large and high-dimensional problems.

Metric learning [32,33] projects data from the original space to a new low-dimensional space that best suits the Euclidean distance to solve the task at hand. Rather than projecting

data in low-dimensional space by ignoring dimensions with small eigen values, regularized matrix relevance learning [28] uses a regularization scheme which inhibits decays in the eigen profile. Both of these techniques require intensive learning which is computationally expensive in large and/or high-dimensional data sets. They optimize distance metric specifically for the given task which may not be good for other tasks using the same data set. They are not a general-purpose measures like m_p, ℓ_p or d_{cos}.

All the adaptive metric learning techniques discussed in the literature attempt to adjust the inter-point distances in the space based on the data distribution that satisfies some optimality constraints. Because the transformed space is still embedded in the Euclidean space, the self-similarity is still constant regardless of the data distribution, and they still rely on geometric model and metric assumptions. Even though metric-based measures have a nice mathematical properties, their assumptions might be inappropriate to model some problems. Recently, Schleif and Tino [27] discussed issues of metric-based proximity learning and provided a comprehensive review of non-metric proximity learning.

In this paper, we focus on general-purpose distance or dissimilarity measures which requires no learning. We have evaluated the performance of the proposed data-dependent general-purpose dissimilarity measure m_p against the geometric general-purpose distance measures ℓ_p and d_{cos}. In the future, it would be interesting to investigate how learning can be applied to data-dependent dissimilarity measure such as m_p to produce non-metric learning and then compare non-metric learning with metric learning.

Because of the implementation of m_p using bins, one can see some similarity with Locality Sensitive Hashing (LSH) [15]. The aims of binning are different in the two cases. In LSH, bins are used to find a small set of candidate nearest neighbors of a test instance quickly where the k-NNs are searched using the Euclidean distance. In contrast, m_p probability data mass in bins is used as a measure of dissimilarity directly. It is an open question whether LSH can be used to generate candidate set quickly for m_p. LSH has a nice theoretical bounds for the Euclidean distance but it is not clear if similar bounds can be derived for m_p.

7 Conclusions and future work

In this paper, we proposed a new dissimilarity measure called "m_p-dissimilarity". It estimates the dissimilarity between two instances in each dimension as a probability data mass in the region enclosing the two instances. The final dissimilarity between the two instances is estimated by combining all single-dimensional dissimilarities as in the case of ℓ_p. The fundamental difference between the formulations of m_p and ℓ_p is the replacement of the geometric distance with the probability mass in each dimension.

Our empirical evaluations in k-NN classification and content-based multimedia information retrieval tasks show that m_p provides better closest matches than those provided by ℓ_p and cosine distance in high-dimensional spaces. Its performance is more consistent across different data sets. By simply replacing the geometric distance in each dimension with the probability mass, k-NN using m_p significantly improves the performance of k-NN using ℓ_p in many high-dimensional data sets.

In contrast to the commonly used distance measures, m_p is not using the values of instances in each dimension in the measure directly. Because it is based on data mass, it is insensitive to units and scale of measurement and the difference in variance of values of instances between dimensions. Thus, it does not require any preprocessing such as min–max normalization to rescale values in the same range, or standard deviation normalization to ensure unit variance across all dimensions, or TF-IDF weighting to adjust the importance of a term in a document.

Even though ℓ_p can be made data dependent though rank transformation, it is applicable only in the case where all instances have distinct values (or a few duplicates only) in each dimension. However, the data-dependent characteristics of m_p is applicable in both cases of with and without many instances having duplicate values in many dimensions. Many instances having duplicate values in many dimensions are a common characteristic of high-dimensional data sets where data lies in a low-dimensional subspace. In such high-dimensional data sets, m_p produces better task-specific performance than ℓ_p with the rank transformation.

Future work includes investigating learning for m_p and compare the non-metric learning with metric learning; examining the effectiveness of m_p in other data mining tasks such as clustering, anomaly detection, vector quantization and SVM kernel learning; and developing indexing schemes for m_p to speed up the nearest neighbor search in the case of large N.

Acknowledgements The preliminary version of this paper is published in Proceedings of the IEEE International conference on data mining (ICDM) 2014 [3]. We would like to thank anonymous reviewers for their useful comments. Kai Ming Ting is partially supported by the Air Force Office of Scientific Research (AFOSR), Asian Office of Aerospace Research and Development (AOARD) under Award Number FA2386-13-1-4043. Takashi Washio is partially supported by the AFOSR AOARD Award Number 15IOA008-154005 and JSPS KAKENHI Grant Number 2524003.

Appendix 1: Probabilistic interpretation of m_p

The formulation of $m_p(\mathbf{x}, \mathbf{y})$ (Eq. 8) has a probabilistic interpretation. The simplest form of data-dependent dissimilarity measure is to define an M-dimensional region $R(\mathbf{x}, \mathbf{y})$ that encloses $\mathbf{x}$ and $\mathbf{y}$, and to estimate the probability of a randomly selected point $\mathbf{t}$ from the distribution of data, $\phi(\mathbf{x})$, falling in $R(\mathbf{x}, \mathbf{y})$, $P(\mathbf{t} \in R(\mathbf{x}, \mathbf{y})|\phi(\mathbf{x}))$. Let $R(\mathbf{x}, \mathbf{y})$ has length of $R_i(\mathbf{x}, \mathbf{y})$ in dimension i. Assuming that the dimensions are independent, $P(\mathbf{t} \in R(\mathbf{x}, \mathbf{y})|\phi(\mathbf{x}))$ can be approximated as:

$$P(\mathbf{t} \in R(\mathbf{x}, \mathbf{y})|\phi(\mathbf{x})) \approx \prod_{i=1}^{M} P(t_i \in R_i(\mathbf{x}, \mathbf{y})|\phi_i(\mathbf{x})) \tag{14}$$

where $P(t_i \in R_i(\mathbf{x}, \mathbf{y})|\phi_i(\mathbf{x}))$ is the probability of t_i falling in $R_i(\mathbf{x}, \mathbf{y})$ for dimension i.

The approximation in Eq. 14 is sensitive to outliers. An approximation which is tolerant to outliers can be estimated by replacing the product with the summation [23]. The sum-based approximation relates to the probability of $\mathbf{t}$ in Eq. 14 under the following *outlier model*. Consider a data generation process in which in order to sample t_i, a coin with probability of turning head $(1 - \epsilon)$ is flipped. If the coin turns head, t_i is drawn from the distribution of data in dimension i, $\phi_i(\mathbf{x})$, where the probability of sampling t_i is $P_i(t_i|\phi_i(\mathbf{x}))$, otherwise it is sampled from the uniform distribution with probability $1/A$, and A is a constant.

Lemma 1 *[23] Under the data generation process described above, the probability of a data point $P'(\cdot)$ can be approximated as*

$$P'(\mathbf{t}|\phi(\mathbf{x}), \epsilon) \approx C_1 + C_2 \times \sum_{i=1}^{M} P_i(t_i|\phi_i(\mathbf{x}))$$

where C_1 and C_2 are constants.

Proof Under the outlier model, the probability of generating the value of the i'th dimension t_i is

$$P'(t_i|\phi(\mathbf{x}),\epsilon) = \epsilon/A + (1-\epsilon)P(t_i|\phi_i(\mathbf{x})) \tag{15}$$

We assume that each dimension is generated independently, hence

$$P'(\mathbf{t}|\phi(\mathbf{x}),\epsilon) \approx \prod_{i=1}^{M} P'(t_i|\phi(\mathbf{x}),\epsilon) = \prod_{i=1}^{M} (\epsilon/A + (1-\epsilon)P(t_i|\phi_i(\mathbf{x})))$$
$$= (\epsilon/A)^M + (\epsilon/A)^{M-1}(1-\epsilon)\sum_{i=1}^{M} P(t_i|\phi_i(\mathbf{x})) + O\left((1-\epsilon)^2\right)$$

In the extreme case where the probability of generating t_i from the uniform distribution (i.e., the outlier component) is high, i.e., ϵ is close to 1, only the first two terms matter. Assuming $C_1 := (\epsilon/A)^M$ and $C_2 := (\epsilon/A)^{M-1}(1-\epsilon)$, the lemma follows. □

In addition to the above approximation given by Minka [23], we propose that the chance of t_i being drawn from the outlier model can be further reduced by sampling from $\phi_i(\mathbf{x})^p$, $p > 1$ when coin turns up head in the above mentioned data generation process. The probability of sampling t_i from $\phi_i(\mathbf{x})^p$ is $\frac{P(t_i|\phi_i(\mathbf{x}))^p}{Z_{i,p}}$, where $P(\cdot)^p$ is the probability of a random event occurring in p successive trials and $Z_{i,p}$ is the normalization constant to ensure the total probability sums up to 1 in the i^{th} dimension.

Lemma 2 *Under the data generation process of sampling from exponential distribution described above, the probability of a data point $P''(\cdot)$ can be approximated as*

$$P''(\mathbf{t}|\phi(\mathbf{x}),\epsilon,p) \approx C_1 + C_2 \times \sum_{i=1}^{M} \frac{P_i(t_i|\phi_i(\mathbf{x}))^p}{Z_{i,p}}$$

where C_1, C_2, and $\{Z_{i,p}\}_{i=1}^{M}$ are constants.

Proof This follows from Lemma 1 by drawing t_i from $\phi_i(\mathbf{x})^p$ $p > 1$ when coin turns up head in the data generation process. □

As a result of Lemma 2 (by considering the outlier tolerant model), $P(\mathbf{t} \in R(\mathbf{x},\mathbf{y}))$ can be approximated as:

$$P(\mathbf{t} \in R(\mathbf{x},\mathbf{y})) \approx C_1 + C_2 \times \sum_{i=1}^{M} \frac{P_i(t_i \in R_i(\mathbf{x},\mathbf{y}))^p}{Z_{i,p}} \tag{16}$$

Note that $P(\mathbf{t} \in R(\mathbf{x},\mathbf{y}))$ is a data-dependent dissimilarity measure for $\mathbf{x}$ and $\mathbf{y}$. All the constants on RHS of Eq. 16 are independent of $\mathbf{x}$ and $\mathbf{y}$ and they are just the scaling factors of the dissimilarity measure. Particularly, in order to find the nearest neighbor of $\mathbf{x}$ among a collection of data instances, the only important term in the measure is $\sum_{i=1}^{M} P_i(t_i \in R_i(\mathbf{x},\mathbf{y}))^p$. The constants can be ignored as they do not change the ranking of data points. Hence, by ignoring the constants in Eq. 16, $m_p(\mathbf{x},\mathbf{y})$ can be expressed as its rescaled version as follows:

$$m_p(\mathbf{x},\mathbf{y}) = \left(\frac{1}{M}\sum_{i=1}^{M} P_i\,(t_i \in R_i(\mathbf{x},\mathbf{y}))^p\right)^{\frac{1}{p}} \tag{17}$$

where the outer power of $\frac{1}{p}$ is just a rescaling factor and $\frac{1}{M}$ is a constant.

In practice, $P_i\ (t_i \in R_i(\mathbf{x}, \mathbf{y}))$ can be estimated from D as:

$$\hat{P}_i\ (t_i \in R_i(\mathbf{x}, \mathbf{y})) = \frac{|R_i(\mathbf{x}, \mathbf{y})|}{N} \tag{18}$$

Hence, Eqs. 17 and 18 lead to m_p defined in Eq. 8.

Appendix 2: Analysis of concentration and hubness

In order to examine the concentration and hubness of the three dissimilarity measures m_2, ℓ_2 and d_{cos} in different data distributions with the increase in the number of dimensions, we used synthetic data sets with uniform (each dimension is uniformly distributed between [0,1]) and normal (each dimension is normally distributed with zero mean and unit variance) distributions with $M = 10$ and $M = 200$. Feature vectors were normalized to be in unit range in each dimension.

Concentration

The relative contrast between the nearest and farthest neighbor is computed for all $N = 1000$ instances in each data set using m_2, ℓ_2 and d_{cos}. The relative contrast for each instance in uniform and normal distributions with $M = 10$ and $M = 200$ are shown in Fig. 5.

The relative contrast of all three measures decreased substantially (note that the y-axes have different scales in Fig. 5) when the number of dimensions was increased from $M = 10$

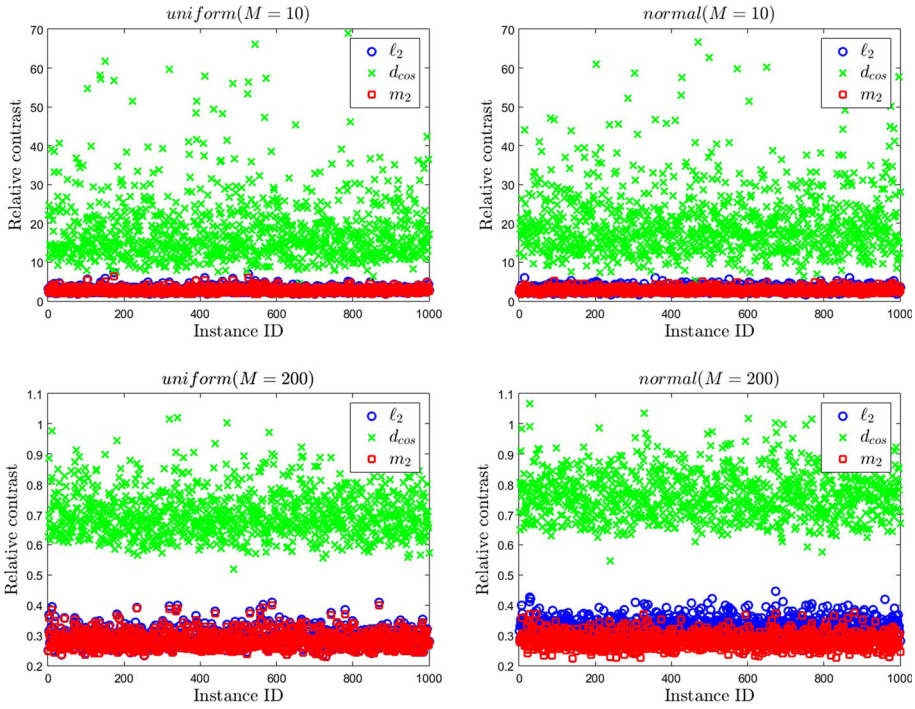

Fig. 5 Relative contrast $\left(\frac{dmax(\mathbf{x},d)-dmin(\mathbf{x},d)}{dmin(\mathbf{x},d)}\right)$ of m_2, ℓ_2 and d_{cos}. Note that x axis is instance id and corresponding y axis value is the relative contrast of that instance

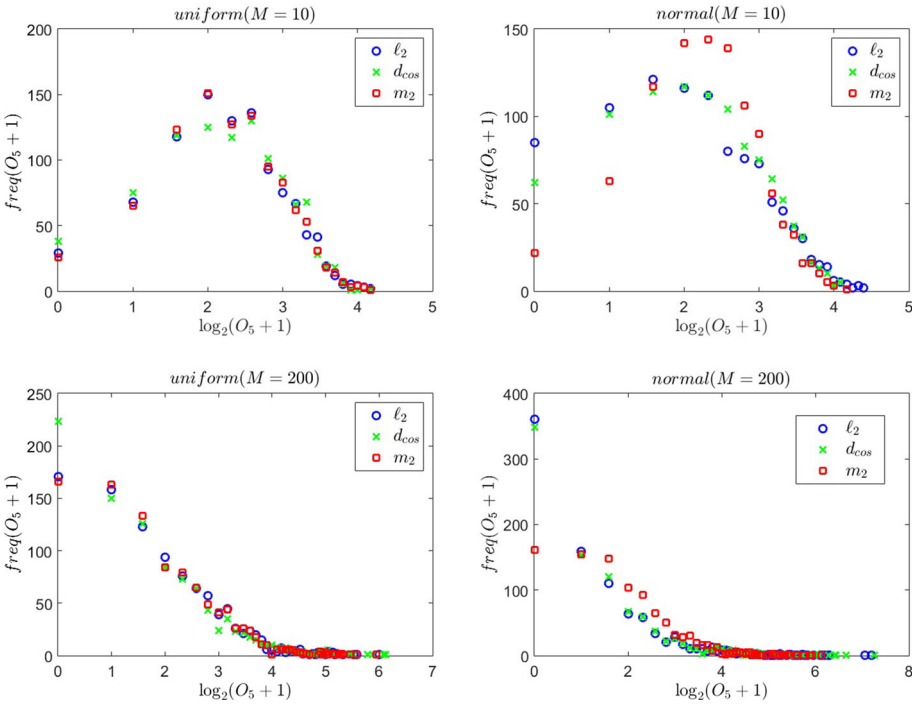

Fig. 6 The O_5 distributions of m_2, ℓ_2 and d_{cos} in synthetic data sets. Note that x axis is in the log scale hence x axis value is $\log(O_5 + 1)$ to consider the case of $O_5 = 0$

to $M = 200$ in both distributions. It is interesting to note that m_2 has the least relative contrast in both distributions with $M = 10$ and $M = 200$; and d_{cos} has the maximum relative contrast in all cases. The relative contrasts of ℓ_2 and m_2 are almost the same except in the case of normal ($M = 200$), where the relative contrast of ℓ_2 is slightly higher than that of m_2 for many instances.

This suggests that m_2 is more concentrated than ℓ_2 and d_{cos}. Even in real-world data sets, we observed that m_2 is more concentrated than ℓ_2 and d_{cos}.

Hubness

In order to examine the hubness phenomenon, 5-Occurrences of each instance $\mathbf{x} \in D$ is estimated, i.e., $O_5(\mathbf{x}) = |\{\mathbf{y} : \mathbf{x} \in N_5(\mathbf{y})\}|$, where $N_5(\mathbf{y})$ is the set of 5-NN of $\mathbf{y}$. Then, the O_5 distribution is plotted for each measure (m_2, ℓ_2 and d_{cos}) in all four synthetic data sets which is shown in Fig. 6.

The O_5 distributions of all three measures become skewed when the number of dimensions was increased from $M = 10$ to $M = 200$ in both distributions. It is interesting to note that the O_5 distributions of m_2 in uniform and normal distributions are almost similar for both $M = 10$ and $M = 200$, whereas those of ℓ_2 and d_{cos} in the case of normal distribution are more skewed than those in uniform distribution for both $M = 10$ and $M = 200$. Note that the O_5 distributions of m_2 and ℓ_2 in uniform distribution are similar for both $M = 10$ and $M = 200$. This is because of the fact that m_2 is proportional to ℓ_2 under uniform distribution (also reflected in Fig. 2a). In the case of normal distribution and $M = 200$, the O_5 distribution of m_2 is less skewed than those of ℓ_2 and d_{cos}. There are 361 and 348 (out of 1000) instances

with $O_5 = 0$ (which do not occur in the 5-NN set of any other instance) in the case of ℓ_2 and d_{cos}, respectively; whereas there are only 161 instances with $O_5 = 0$ in the case of m_2. Similarly, the most popular nearest neighbors using ℓ_2 and d_{cos} have $O_5 = 146$ and 152, respectively; whereas the most popular nearest neighbor using m_2 has $O_5 = 69$.

We observed similar behavior in many real-world data sets as well where the O_5 distribution of m_2 is less skewed than that of ℓ_2 and d_{cos}.

Appendix 3: Standard error

Table 10 shows the standard error of classification accuracies (in %) of k-NN classification ($k = 5$) over a tenfold cross-validation (average classification accuracy is presented in Table 4 in Sect. 4.1).

Table 10 Standard error of accuracies of k-NN classification ($k = 5$) over a tenfold cross-validation. Average classification accuracy is presented in Table 4 in Sect. 4.1

Data set	d_{cos}	$\ell_{0.5}$	ℓ_2	s-$\ell_{0.5}$	s-ℓ_2	d_{lin}	$m_{0.5}$	m_2
New3s	0.35	0.59	0.67	0.67	0.66	0.03	0.34	0.36
Ohscal	0.57	0.48	0.49	0.81	0.70	0.01	0.26	0.26
Arcene	2.49	1.45	1.83	1.98	2.11	2.00	2.96	2.17
Wap	0.78	0.65	0.72	0.66	0.77	1.07	0.83	1.08
R52	0.51	0.23	0.44	0.45	0.42	0.13	0.31	0.25
NG20	0.22	0.42	0.39	0.30	0.24	0.05	0.19	0.23
Gisette	0.16	0.23	0.19	0.27	0.23	0.44	0.22	0.14
R8	0.29	0.40	0.51	0.42	0.45	0.08	0.25	0.29
Fbis	0.80	2.90	1.04	1.91	1.30	1.47	0.71	0.75
Webkb	0.53	0.43	0.75	0.30	0.70	0.28	0.51	0.40
Ads	0.28	0.24	0.30	0.20	0.29	0.23	0.28	0.29
Caltech	0.15	0.09	0.20	0.09	0.14	0.06	0.10	0.11
Mnist	0.08	0.09	0.08	0.09	0.08	0.28	0.07	0.06
Mfeat	0.30	0.32	0.29	0.25	0.32	0.40	0.37	0.38
Isolet	0.45	0.27	0.40	0.23	0.48	0.22	0.33	0.33
Madelon	1.04	1.30	1.18	1.46	1.35	0.78	0.79	0.98
Arrhythmia	2.00	1.32	2.01	1.76	1.42	1.68	1.89	2.34
Gtzan	1.68	1.32	1.61	1.41	1.49	1.48	1.20	1.67
Ismis	0.23	0.24	0.20	0.28	0.24	0.17	0.16	0.19
Hba	1.18	1.31	0.88	1.20	1.21	1.50	1.12	1.36
Musk2	0.18	0.15	0.21	0.15	0.14	0.16	0.13	0.09
Corel	0.44	0.38	0.41	0.41	0.38	0.43	0.49	0.38
Splice	0.67	0.67	0.67	0.67	0.67	0.59	0.41	0.54
Miniboone	0.07	0.07	0.07	0.07	0.05	0.05	0.06	0.07
Connect-4	0.11	0.11	0.11	0.11	0.11	0.19	0.17	0.12
Annealing	1.24	1.48	1.22	1.30	1.35	1.38	1.46	1.46
Satellite	0.29	0.38	0.27	0.35	0.22	0.28	0.39	0.34
Chess	0.39	0.39	0.39	0.39	0.39	0.33	0.29	0.34
Hypothyroid	0.15	0.13	0.15	0.23	0.17	0.21	0.27	0.13
Credit-g	1.41	1.12	1.37	1.12	1.26	1.10	0.89	1.25

Table 11 Standard error of P@10 over N queries. Average P@10 is presented in Table 6 in Sect. 4.2

Data set	d_{cos}	$\ell_{0.5}$	ℓ_2	s-$\ell_{0.5}$	s-ℓ_2	d_{lin}	$m_{0.5}$	m_2
New3s	0.004	0.002	0.004	0.002	0.002	0.002	0.003	0.003
Ohscal	0.003	0.002	0.002	0.002	0.001	0.001	0.003	0.003
Wap	0.009	0.006	0.007	0.006	0.006	0.006	0.008	0.008
R52	0.003	0.004	0.004	0.004	0.004	0.004	0.003	0.003
NG20	0.002	0.001	0.002	0.001	0.002	0.002	0.002	0.002
Fbis	0.006	0.005	0.007	0.005	0.006	0.006	0.006	0.006
Caltech	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
Gtzan	0.009	0.010	0.010	0.010	0.010	0.009	0.010	0.010
Hba	0.007	0.007	0.007	0.007	0.007	0.008	0.008	0.007
Corel	0.002	0.003	0.002	0.003	0.002	0.003	0.003	0.003

Table 11 shows the standard error of precision at top 10 retrieved results (P@10) over N queries in content-based multimedia information retrieval (average P@10 is presented in Table 6 in Sect. 4.2).

Appendix 4: Comparison with geometric distance measures after dimensionality reduction

Average 5-NN classification accuracies over a tenfold cross-validation of d_{cos}, $\ell_{0.5}$ and ℓ_2 before and after dimensionality reduction through PCA along with those of $m_{0.5}$ and m_2 in

Table 12 Average accuracy of 5-NN classification over a tenfold cross-validation

Data set	Dim. red. with PCA			Orginal dimensions				
	d_{cos}	$\ell_{0.5}$	ℓ_2	d_{cos}	$\ell_{0.5}$	ℓ_2	$m_{0.5}$	m_2
Caltech	12.76	03.80	07.17	11.40	02.90	08.46	13.83	14.68
Corel	28.32	26.84	28.50	24.59	35.66	23.68	39.76	35.30
Fbis	71.87	56.44	65.00	77.91	48.18	70.40	79.21	78.85
Gissette	96.66	72.24	95.50	97.76	94.59	96.50	96.77	97.73
Gtzan	72.40	51.10	65.90	70.90	65.00	70.40	72.00	68.80
Hba	57.47	41.70	55.20	50.20	59.07	52.00	67.07	60.73
Ismis	94.86	92.41	93.96	94.53	94.35	94.41	95.54	94.48
Isolet	87.43	84.28	87.77	88.37	83.71	89.16	79.68	82.42
Madelon	57.85	51.62	55.14	57.27	60.92	56.88	59.23	55.00
Mfeat	97.90	97.20	98.10	98.00	98.15	98.20	97.85	98.20
Miniboone	92.47	92.36	92.79	92.65	93.03	92.63	92.77	92.94
Mnist	94.99	92.07	95.24	97.66	95.62	97.19	95.77	97.23
Musk2	96.15	97.42	96.54	96.45	95.35	96.62	95.01	95.47
R8	80.87	61.17	65.77	90.36	81.89	79.59	94.94	93.72
Satellite	88.98	90.09	90.74	84.86	90.68	90.97	90.97	90.85
Webkb	72.02	51.25	59.14	73.40	51.28	63.85	85.23	84.31
Avg.	75.19	66.37	72.03	75.39	71.90	73.81	78.48	77.54

the original space in 16 out of 22 data sets with continuous only attributes are provided in Table 12. With PCA, the number of dimensions was reduced by projecting data in the lower-dimensional space defined by the principal components capturing 95% of the variance in data. The principal components were computed by the eigen decomposition of the correlation matrix of the training data to ensure that the projection is robust to scale differences in the original dimensions. Note that PCA did not complete in 24 h in the remaining six data sets with $M > 5000$: New3s (26,832), Ohscal (11,465), Arcene (10,000), Wap (8460), R52 (7369) and NG20 (5489).

References

1. Aggarwal CC, Hinneburg A, Keim DA (2001) On the surprising behavior of distance metrics in high dimensional space. In: Proceedings of the 8th international conference on database theory. Springer, Berlin, pp. 420–434
2. Ariyaratne HB, Zhang D (2012) A novel automatic hierachical approach to music genre classification. In: Proceedings of the 2012 IEEE international conference on multimedia and expo workshops, IEEE Computer Society, Washington DC, USA, pp. 564–569
3. Aryal S, Ting KM, Haffari G, Washio T (2014) Mp-dissimilarity: a data dependent dissimilarity measure. In: Proceedings of the IEEE international conference on data mining. IEEE, pp. 707–712
4. Bache K, Lichman M (2013) UCI machine learning repository, http://archive.ics.uci.edu/ml. University of California, Irvine, School of Information and Computer Sciences
5. Bellet A, Habrard A, Sebban M (2013) A survey on metric learning for feature vectors and structured data, Technical Report, arXiv:1306.6709
6. Beyer KS, Goldstein J, Ramakrishnan R, Shaft U (1999) When Is "Nearest Neighbor" meaningful? Proceedings of the 7th international conference on database theory. Springer, London, pp. 217–235
7. Boriah S, Chandola V, Kumar V (2008) Similarity measures for categorical data: a comparative evaluation. In: Proceedings of the eighth SIAM international conference on data mining, pp. 243–254
8. Cardoso-Cachopo A (2007) Improving methods for single-label text categorization, PhD thesis, Instituto Superior Tecnico, Technical University of Lisbon, Lisbon, Portugal
9. Conover WJ, Iman RL (1981) Rank transformations as a bridge between parametric and nonparametric statistics. Am Stat 35(3):124–129
10. Deza MM, Deza E (2009) Encyclopedia of distances. Springer, Berlin
11. Fodor I (2002) A survey of dimension reduction techniques. Technical Report UCRL-ID-148494, Lawrence Livermore National Laboratory
12. François D, Wertz V, Verleysen M (2007) The concentration of fractional distances. IEEE Trans Knowl Data Eng 19(7):873–886
13. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The weka data mining software: an update. SIGKDD Explor Newsl 11(1):10–18
14. Han E-H, Karypis G (2000) Centroid-based document classification: Analysis and experimental results. In: Proceedings of the 4th European conference on principles of data mining and knowledge discovery. Springer, London, pp. 424–431
15. Indyk P, Motwani R (1998) Approximate nearest neighbors: towards removing the curse of dimensionality. Proceedings of the thirtieth annual ACM symposium on theory of computing, STOC '98, ACM, New York, pp. 604–613
16. Jolliffe I (2005) Principal component analysis. Wiley Online Library, Hoboken
17. Krumhansl CL (1978) Concerning the applicability of geometric models to similarity data: the interrelationship between similarity and spatial density. Psychol Rev 85(5):445–463
18. Kulis B (2013) Metric learning: a survey. Found Trends Mach Learn 5(4):287–364
19. Lan M, Tan CL, Su J, Lu Y (2009) Supervised and traditional term weighting methods for automatic text categorization. IEEE Trans Pattern Anal Mach Intell 31(4):721–735
20. Lin D (1998) An information-theoretic definition of similarity. In: Proceedings of the fifteenth international conference on machine learning. Morgan Kaufmann Publishers Inc., San Francisco, pp. 296–304
21. Lundell J, Ventura D (2007) A data-dependent distance measure for transductive instance-based learning. In: Proceedings of the IEEE international conference on systems, man and cybernetics, pp. 2825–2830
22. Mahalanobis PC (1936) On the generalized distance in statistics. Proc Natl Inst Sci India 2:49–55
23. Minka TP (2003) The 'summation hack' as an outlier model, http://research.microsoft.com/en-us/um/people/minka/papers/minka-summation.pdf. Microsoft Research

24. Radovanović M, Nanopoulos A, Ivanović M (2010) Hubs in space: popular nearest neighbors in high-dimensional data. J Mach Learn Res 11:2487–2531
25. Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Inf Process Manage 24(5):513–523
26. Salton G, McGill MJ (1986) Introduction to modern information retrieval. McGraw-Hill Inc, New York
27. Schleif F-M, Tino P (2015) Indefinite proximity learning: a review. Neural Comput 27(10):2039–2096
28. Schneider P, Bunte K, Stiekema H, Hammer B, Villmann T, Biehl M (2010) Regularization in matrix relevance learning. IEEE Trans Neural Netw 21(5):831–840
29. Tanimoto TT (1958) Mathematical theory of classification and prediction, International Business Machines Corp
30. Tuytelaars T, Lampert C, Blaschko MB, Buntine W (2010) Unsupervised object discovery: a comparison. Int J Comput Vision 88(2):284–302
31. Tversky A (1977) Features of similarity. Psychol Rev 84(4):327–352
32. Wang F, Sun J (2015) Survey on distance metric learning and dimensionality reduction in data mining. Data Min Knowl Disc 29(2):534–564
33. Yang L (2006) Distance metric learning: a comprehensive survey, *Technical report*, Michigan State Universiy
34. Zhou G-T, Ting KM, Liu FT, Yin Y (2012) Relevance feature mapping for content-based multimedia information retrieval. Pattern Recogn 45(4):1707–1720

Sunil Aryal is a lecturer in the School of Engineering and Information Technology at Federation University, Australia. He is also a PhD candidate at Monash University, Australia. His research interests include data mining, machine learning, health informatics and Information systems. He received a BIT degree from Purbanchal University, Nepal in 2005; an MIT (Coursework) degree from the University of Southern Queensland, Australia in 2008; and an MIT (Research) degree from Monash University, Australia, in 2012. Before joining academia, he worked in IT industry for a number of years.

Kai Ming Ting is a professor in the Faculty of Science and Technology at Federation University, Australia. His current research interests are in the areas of mass estimation, anomaly detection, ensemble approaches, data streams, data mining and machine learning in general. After receiving his PhD from the University of Sydney, he had worked at the University of Waikato, Deakin University and Monash University. He has served as a program committee co-chair for the Twelfth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-2008). He was a member of the program committee for a number of international conferences including ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, and International Conference on Machine Learning. He has received research funding from Australian Research Council, US Air Force of Scientific Research (AFOSR/AOARD), Toyota InfoTechnology Center, and Australian Institute of Sports. Awards received include the Runner-up Best Paper Award in 2008 IEEE ICDM, and the Best Paper Award in 2006 PAKDD.

Takashi Washio is a full professor in Department of Reasoning for Intelligence, The Institute of Scientific and Industrial Research, Osaka University, which is located at Ibaraki City, Osaka, Japan. His department in Osaka University focuses on basic studies of machine learning and data mining and is a leading research group in Japan. His current main research interests are machine learning principles for high dimensional big data in the basic study and machine learning techniques for scientific advanced sensing in the application study. Takashi Washio obtained his M.E. and PhD in the field of nuclear measurement and instrumentation control at Tohoku University, Miyagi, Japan, in 1985 and 1988, respectively. He was previously a visiting researcher of a Nuclear Reactor Laboratory of MIT (Massachusetts Institute of Technology), USA, from 1988 to 1990, a senior researcher of Mitsubishi Research Institute from 1990 to 1996, and an associate professor in Osaka University from 1996 to 2006.

Gholamreza Haffari is a Senior Lecturer in the Faculty of Information Technology, Monash University, Australia. His research interests lie in the intersection of natural language processing and machine learning, including topics such as machine translation, sentiment analysis and text classification, language modelling, parsing, structured prediction, deep learning, probabilistic graphical models, and scalable machine learning. He earned his PhD from Simon Fraser University in 2009, and was a Postdoctoral Fellow at University of British Columbia in 2010–2011.