A data-dependent dissimilarity measure: An effective alternative to distance measures

by

Sunil Aryal, MIT(Res)

Thesis

Submitted by Sunil Aryal

in fulfillment of the requirements for the degree of

Doctor of Philosophy (0190)

Main Supervisor: Prof. Kai Ming Ting

Associate Supervisor: Dr. Gholamreza Haffari

Associate Supervisor: Prof. Takashi Washio

School of Information Technology

Monash University

Clayton Campus

September, 2017

To my lovely wife Pratibha

Contents

List of Tables

List of Figures

A data-dependent dissimilarity measure: An effective alternative to distance measures

Sunil Aryal, MIT(Res)
[redacted]
Monash University, 2017

Main Supervisor: Prof. Kai Ming Ting
kaiming.ting@federation.edu.au
Associate Supervisor: Dr. Gholamreza Haffari
gholamreza.haffari@monash.edu
Associate Supervisor: Prof. Takashi Washio
washio@ar.sanken.osaka-u.ac.jp

Abstract

In data mining, the conventional approach to measuring similarities of data instances is primarily based on a geometric model, where data are assumed to be embedded in a multidimensional space and the similarity of two instances is estimated as the inverse of their distance in the space. Minkowski distance (also known as ℓ_p-norm with $p > 0$) and cosine distance are the most widely-used similarity measures. Their performances vary significantly in different data distributions for two main reasons: (i) the similarity of two instances is solely based on their spatial positions in the space and it is independent of the distribution of data; and (ii) the spatial distance is sensitive to units and scales of measurement.

This thesis investigates a (dis)similarity measure where data distribution is the key determinant of the measurement. It introduces a new data-dependent dissimilarity measure called m_p-dissimilarity (with $p \geq 0$). m_p-dissimilarity has exactly the same formulation as the traditional ℓ_p-norm, except that the spatial distance of two instances in each dimension is replaced with the probability data mass between them.

m_p-dissimilarity differs from traditional distance-based measures in two ways: (i) data-dependent dissimilarity: 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; and self-dissimilarity is also data-dependent, i.e., the dissimilarity of two instances is data-dependent when their spatial distance is zero; and (ii) it is robust to units and scales of measurement: the dissimilarity of two instances is based on the data mass between them, which is robust to monotonic transformation of data in each dimension. These two characteristics are particularly important in measuring similarities of instances in high-dimensional spaces.

High-dimensional data often lie in low-dimensional subspaces and many instances have the same value in many dimensions, e.g., bag-of-words (BoW) representation of text documents where many vector components are zeros. Data-dependent dissimilarity, more specifically data-dependent self-dissimilarity, provides a useful means to differentiate between instances. In addition, the properties of data objects in many applications are often measured by different sensors using different units and scales; this information may not be available after the data are collected. Therefore, the use of a measure which is robust to units and scales of measurement provides at least more consistent or better similarity results than the use of a measure which is not.

An analysis of m_p-dissimilarity reveals that it is a generic data-dependent measure. Existing data-dependent measures of rank difference and Lin's probabilistic similarity are special cases with $p > 0$ and $p = 0$, respectively, where the special cases have data-independent self-dissimilarities but m_p-dissimilarity has data-dependent self-dissimilarity.

The empirical evaluation conducted across a wide range of low-to high-dimensional datasets from different applications (e.g., text, image, music etc.) shows that m_p-dissimilarity produces at least more consistent or better task-specific performances than widely-used distance-based measures (e.g., ℓ_p-norm and cosine distance) and existing data-dependent measures (e.g., rank difference and Lin's probabilistic similarity).

This thesis shows that fully data-dependent similarity (which includes data-dependent self-similarity), and robustness to units and scales of measurement, are two important characteristics of a similarity measure in order to produce consistent task-specific performance across a wide range of datasets. The m_p-dissimilarity introduced in this thesis is one such measure, which has both of these characteristics.

Thesis including published works declaration

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

This thesis includes four papers published in peer-reviewed journals and conference proceedings, and two unpublished publications submitted to peer-reviewed journals. The core theme of the thesis is data-dependent dissimilarity measurement of data objects. The ideas, development and writing up of all the papers in this thesis were the principal responsibility of myself, the student, working within the Clayton School of Information Technology under the supervision of Prof. Kai Ming Ting, Dr. Gholamreza Haffari and Prof. Takashi Washio.

The inclusion of co-authors reflects the fact that the work came from active collaboration between researchers and acknowledges input into team-based research.

In the case of chapters of this thesis based on publications, my contribution to the work involved the following:

Thesis Chapter	**Publication title**	**Status**	**Nature and % of student contribution**	**Co-author name(s) Nature and % of Co-author's contribution**	**Co-author Monash Student (Yes/No)**
2	Improving iForest with relative mass	Published	Concept, implementation, experimentation, preparation of figures and manuscript, 75%	1) K.M. Ting, Input into manuscript, 15% 2) J.R. Wells, Input into experiments and preparing figures, 5% 3) T. Washio, Input into manuscript, 5%	No Yes No
3	Data-dependent dissimilarity measure: An effective alternative to geometric distance measures	Accepted Published online	Concept, implementation, experimentation, preparation of figures and manuscript, 75%	1) K.M. Ting, Input into manuscript, 15% 2) T. Washio, Input into manuscript, 5% 3) G. Haffari, Input into manuscript, 5%	No No No
4	A comparative study of data-dependent approaches to measuring similarities of data objects	Submitted	Concept, implementation, experimentation, preparation of figures and manuscript, 75%	1) K.M. Ting, Input into manuscript, 15% 2) T. Washio, Input into manuscript, 5% 3) G. Haffari, Input into manuscript, 5%	No No No
5	A new simple and effective measure for inter-document similarity measurement	Submitted	Concept, implementation, experimentation, preparation of figures and manuscript, 75%	1) K.M. Ting, Input into manuscript, 15% 2) T. Washio, Input into manuscript, 5% 3) G. Haffari, Input into manuscript, 5%	No No No
Appendix A	m_p-dissimilarity: A data-dependent dissimilarity measure	Published	Concept, implementation, experimentation, preparation of figures and manuscript, 75%	1) K.M. Ting, Input into manuscript, 15% 2) G. Haffari, Input into manuscript, 5% 3) T. Washio, Input into manuscript, 5%	No No No

Thesis Chapter	Publication title	Status	Nature and % of student contribution	Co-author name(s) Nature and % of Co-author's contribution	Co-author Monash Student (Yes/No)
Appendix B	Beyond tf-idf and cosine distance in document dissimilarity measures	Published	Concept, implementation, experimentation, preparation of figures and manuscript, 75%	1) K.M. Ting, Input into manuscript, 15% 2) T. Washio, Input into manuscript, 5% 3) G. Haffari, Input into manuscript, 5%	No No No

In order to generate a consistent presentation within the thesis, I have changed the format and some notations or symbols used, corrected minor grammar and spelling mistakes, and renumbered sections of published and submitted papers.

Student's signature: [redacted] **Date:** 01 / 09 / 2017

The undersigned hereby certifies that the above declaration correctly reflects the nature and extent of the student's contributions to this work. In instances where I am not the responsible author I have consulted with the responsible author to agree on the respective contributions of the authors.

Main Supervisor's signature: [redacted] **Date:** 01/09/2017

Acknowledgments

Firstly, I would like to express my sincere gratitude to my main supervisor Prof. Kai Ming Ting, and associate supervisors Dr. Gholamreza Haffari and Prof. Takashi Washio, for providing me the opportunity to do this research with them. I would like to thank them for all their continuous guidance, encouragement and critical comments throughout my candidature on conducting quality scientific research.

I would like to thank Prof. Bala Srinivasan, Dr. David Albrecht, Dr. Mark Carman and Dr. Arun Konagurthu for being panel members in my PhD milestone seminars and providing constructive comments and valuable suggestions.

I am indebted to the government of Australia and Monash University for supporting my PhD research with a scholarship.

I would like to thank Dr. Phil Smith and Dr. Alex McKnight for proofreading this thesis. A big thank you also goes to A/Prof. Peter Vamplew for providing useful comments to improve the quality of two papers included in this thesis.

I am grateful to my lovely wife Pratibha Sharma for her continuous moral support through the highs and lows of my PhD journey. Despite the immense stress and grief she has suffered in the last few years, she always stood by my side and motivated me to achieve my goal. Thank you so much sweetheart!

Finally, I would like to thank my parents, siblings, in-laws, friends, faculty members, administrative staff, colleagues at both Monash University and Federation University, and everyone who helped to make my study possible. Thank you all!

Sunil Aryal

Monash University
September 2017

Chapter 1

Introduction

In this information age, data are everywhere in our daily life, such as social media, sales transactions, health care and telecommunications. Massive volumes of data are being added to different databases. Data stored in databases have no value unless they are analysed to extract useful information and knowledge. The enormous volume of data is impossible to analyse manually. Over the last few decades, computers have been widely used to analyse these rapidly growing databases. *Knowledge discovery from databases* (KDD) (Tan et al., 2006; Han and Kamber, 2006) is the process of extracting interesting hidden patterns by analysing databases automatically. KDD comprises the following key steps:

1. Data pre-processing: collecting data from multiple sources and cleaning data to remove noise and irrelevant data.
2. Data mining: analysing pre-processed data using computers to extract useful information.
3. Pattern evaluation: evaluating extracted patterns based on some measure and presenting to users.

1.1 Data mining

Data mining is the process of discovering hidden patterns from data using computers and artificial intelligence (Tan et al., 2006; Han and Kamber, 2006). Data-mining systems use tools and techniques from computer science, mathematics and statistics. The nature and type of interesting patterns extracted from data depend on application domains. Some examples of data-mining tasks are as follows:

1. Anomaly detection: Identification of anomalous records in a given database, e.g., detection of fraudulent credit card transactions, intrusion detection in computer networks, identification of extreme conditions in natural systems such as hurricanes and earthquakes.
2. Classification: Classification of data into one of the predefined categories. This is a widely-used data-mining task in many applications, such as character recognition

(classifying hand-written characters into 26 alphabetical characters), email filtering (classifying emails as spam or not) and cancer diagnosis (predicting whether a tumour is benign or malignant).

3. Clustering: Automatic detection of data clusters. Some applications of clustering are market segmentation in marketing, detecting communities in social networks, identifying homologous gene sequences in bioinformatics.
4. Information retrieval: Retrieval of data records from a given database which are relevant to the information needs of a user. It has a wide range of applications, e.g., recommending new songs to a user based on the songs they like, search engines (presenting relevant information for a query), image retrieval (searching images similar to a given image).
5. Association analysis: Discovery of interesting relationships between variables in a given database. In market basket analysis, association rule mining is used to discover co-occurrences of products in transactions. This is useful for making marketing decisions such as pricing and product placement.
6. Regression: Predicting the value of a target variable of a data instance from the values of its other variables. Predicting share prices in a stock market, and predicting house prices in a real estate market are examples of regression tasks.

Different techniques and algorithms have been introduced to accomplish different data-mining tasks, and many rely on similarities between data instances.

Measuring similarities of data instances is an essential core computation in many data-mining tasks. For example, in content-based information retrieval (CBIR), also known as query-by-example, the task is to rank instances in a given database with respect to their similarities to a given query instance. Similarly, nearest neighbour (NN) search, which is a core process in many data-mining algorithms designed to solve different data-mining tasks, also uses (dis)similarities of data instances to find their nearest neighbour (most similar) instances.

NN-based data-mining techniques are simple and intuitive, and they have been shown to be effective in different data-mining tasks, such as kNN classification (Aha and Kibler, 1991), kMeans clustering (Macqueen, 1967), and kNN-based anomaly detection (Breunig et al., 2000; Bay and Schwabacher, 2003).

The subject of this thesis is measuring similarities between data instances. It primarily deals with the data-mining tasks of CBIR and kNN classification.

1.2 Similarity measures commonly used in data mining

In databases, a real-world entity is represented as a data instance defined by a fixed number of selected features or properties. Let D be a collection of N data instances $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(N)}\}$ where each instance $\mathbf{x}$ is represented as a vector of its values of M features $\langle x_1, x_2, \cdots, x_M \rangle$. Let $s(\mathbf{x}, \mathbf{y}) \in \mathbb{R}$ (where $\mathbb{R}$ is a real domain) be the measure of similarity of two instances $\mathbf{x}$ and $\mathbf{y}$.

The conventional approach to estimating $s(\mathbf{x}, \mathbf{y})$ is primarily based on a geometric model where D is assumed to be embedded in an M-dimensional metric space $\mathcal{X}$, and $s(\mathbf{x}, \mathbf{y})$ is estimated as the inverse of their spatial distance in $\mathcal{X}$—the higher the distance between $\mathbf{x}$ and $\mathbf{y}$, the less their similarity. Because $\mathcal{X}$ is assumed to be a metric space, $s(\mathbf{x}, \mathbf{y})$ has nice mathematical properties which can be exploited in solving many data-mining problems (Deza and Deza, 2009).

Minkowski distance (also known as ℓ_p-norm where $p > 0$) and cosine distance (also known as angular distance) (Deza and Deza, 2009) are the two most widely-used distance measures. The Minkowski distance of $\mathbf{x}$ and $\mathbf{y}$ is estimated by aggregating their spatial distances in every dimension. Euclidean distance (also known as ℓ_2-norm) is a popular choice of distance measure as it intuitively corresponds to the distance in the three-dimensional world humans experience in daily life. The cosine distance of two vectors is proportional to their Euclidean distance if the vectors are normalised to be of unit lengths.

1.3 Thesis motivations: Limitations of distance-based similarity measures

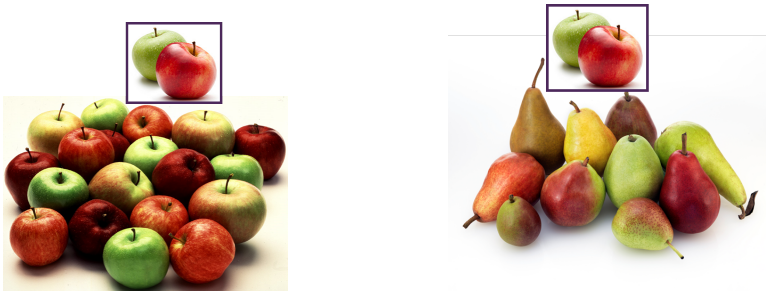
Although distance-based measures perform well in many problems, they have the following three limitations which motivated this thesis.

1.3.1 Task-specific performances vary significantly on different data distributions

The task-specific performances of distance-based similarity measures (e.g. Euclidean distance or cosine distance) depend on the distribution of data, and a distance measure that performs well in one distribution may perform poorly in others. A huge variation in performance can be observed when a distance measure is used in different data distributions. Therefore, a distance measure must be chosen carefully for the given dataset.

It has been suspected that the huge variation in the performance of a distance-based similarity measure in different data distributions occurs because the distance between two instances $\mathbf{x}$ and $\mathbf{y}$ is solely based on their geometric positions in $\mathcal{X}$, and the data distribution is not taken into consideration. Psychologists have expressed their concerns with the geometric model of similarity (Tversky, 1977; Krumhansl, 1978). They have argued that the judged similarity between two instances is influenced by the data distribution between the two instances, i.e., the similarity of two instances is data-dependent.

Krumhansl (1978) introduced a distance-density model of similarity and suggested that two instances in a relatively dense region would be less similar than two instances of equal distance but located in a less dense region. For example, consider evaluating the similarity between two apples (red and green) in two different contexts, where the two apples are among (a) apples of different colours or (b) pears of different colours, as shown in Figure 1.1. The two apples among apples of different colours in case (a) are perceptually less similar than the same two apples among pears of different colours in case (b). It is



(a) Apples (b) Pears

Figure 1.1: Judged similarity of the same red and green apples in two different contexts (Images source: Google Image, 2015).

interesting to note that the two apples are perceived to be less similar in case (a), mainly because there are more instances of the same kind in case (a) than in case (b).

In order to improve the task-specific performance of a distance measure, particularly Euclidean distance, in a given dataset, distance metric learning techniques (Weinberger et al., 2006; Yang, 2006; Weinberger and Saul, 2009; Wang and Sun, 2015) have been used. In distance metric learning, data in the original space $\mathcal{X}$ are projected to a new space $\mathcal{Z}$ (often lower dimensional than $\mathcal{X}$) where the task-specific performance of the distance measure can be maximised in the given dataset.

An appropriate $\mathcal{Z}$ is learned by optimising task-specific constraints. For example, in a classification problem, $\mathcal{Z}$ is learned such that instances belonging to the same class become closer to each other (similarity constraints) and instances belonging to different classes are separated further apart (dissimilarity constraints) (Weinberger et al., 2006).

Thus far, distance metric learning has been shown to produce better task-specific performance than using distance measures in the original space across different datasets in the classification task (Weinberger et al., 2006; Weinberger and Saul, 2009).

Because distance metric learning is tailored to the specific task at hand, it is not a general-purpose similarity measure like Euclidean distance. A distance metric learned for one task may not be good for other tasks in the same dataset. In addition, it is computationally expensive in high-dimensional and/or large datasets because it requires optimisation to find the best $\mathcal{Z}$.

1.3.2 Sensitivity to units and scales of measurement

In the geometric model, there is an implicit assumption that a unit distance implies the same degree of similarity everywhere in the space, regardless of scales or units of measurement. This is referred to as the *interval scale assumption* by Stevens (1946). However, this assumption is often violated in real-world problems where feature values are measured in different units and scales. For example, if the annual incomes of individuals are measured in the logarithmic scale of base 10, $w = \$50\text{k}$, $x = \$150\text{k}$, $y = \$1100\text{k}$ and $z = \$1200\text{k}$ become $w' = 4.70$, $x' = 5.18$, $y' = 6.04$ and $z' = 6.08$. Although $x - w = z - y$ in the original scale, $x' - w' > z' - y'$ in the logarithmic scale. In other words, distance-based similarity measures are sensitive to units and scales of measurement.

The impact of the interval scale assumption can be even worse in multidimensional datasets, where different feature values are often measured by different sensors using different scales and units of measurements. For example, in the previous example of individuals, income can be measured in dollars (which can be in the order of tens or hundreds of thousands) and age can be measured as years in normal integer scale (which is in the order of tens to a hundred). The unit distances in these two features do not provide the same amount of information about the similarity of two individuals. Such differences in many features can have a significant effect on the similarity of two individuals using distance measures.

In many data-mining problems, the units and scales of measurement may not be available where only magnitudes are provided. In order to address this issue to some extent, data pre-processing techniques such as min-max normalisation and standardisation are used (Duda et al., 2000). Min-max normalisation ensures that data in each dimension are in the same range, whereas standardisation ensures that the data in each dimension have zero mean and unit variance. However, these data pre-processing techniques are sensitive to outliers. It is difficult to choose the most appropriate pre-processing technique without any prior knowledge.

In order to deal with this issue, researchers have used different measures which do not rely on the interval scale assumption. One simple solution is to assume that data are ordinal and use measures such as (1) rank difference - distance after rank transformation (Conover and Iman, 1981); and (2) Lin's information theoretic measure (Lin, 1998). However, these methods have high time complexities, particularly when two instances given for the similarity measurement are not in the observed data, limiting them to small datasets only.

It is interesting to note that measures such as rank difference and Lin's probabilistic similarity are data-dependent for $x \neq y$ only, i.e., their similarity will be higher if they lie in a sparse region than in a dense region, as suggested by psychologists. For example, in the above example of annual incomes, two individuals earning $y = \$1100\text{k}$ and $z = \$1200\text{k}$ become more similar to each other than two other individuals earning $w = \$50\text{k}$ and $x = \$150\text{k}$, even though $z - y = x - w = 100\text{k}$ because there are many more individuals earning in [50k, 150k] than those earning in [1100k, 1200k].

However, for $x = y$, the similarity is data-independent even with these measures, i.e., self-similarity of data is a constant everywhere in the space. Because of the constant self-similarity of data, two individuals earning \$1m each become equally similar to each other as two other individuals earning \$50k each, even though there are many more individuals earning \$50k than those earning \$1m. Psychologists argue that the former are judged to be more similar than the latter by humans.

Other similarity measures which do not rely on the interval scale assumption are based on random forest (Shi and Horvath, 2006). Similarly, ReFeat (Zhou et al., 2012) uses a form of random forest called "Isolation Forest" (iForest) (Liu et al., 2008, 2012), which was originally introduced for anomaly detection, to solve the content-based multimedia information retrieval (CBMIR) task. Although these measures are shown to produce better

task-specific results than distance-based measures, they require large ensemble sizes to produce good results, and are not appropriate in high-dimensional datasets. Furthermore, a problem with ReFeat is that it does not guarantee that relevant instances lie in the same local neighbourhood, i.e., $\mathbf{x}$ can be more relevant to $\mathbf{q}$ than $\mathbf{y}$ even though $\mathbf{y}$ lies in closer proximity of $\mathbf{q}$ than $\mathbf{x}$.

1.3.3 Curse of dimensionality

The effectiveness of distance measures such as Minkowski distance decreases as the number of dimensions (M) increases. In high-dimensional space, data distribution becomes sparse, which makes the concepts of distance and nearest neighbour meaningless, i.e., all pairs of data instances are almost equidistant for a wide range of data distributions and distance measures. This is referred to as the *"curse of dimensionality"* (Beyer et al., 1999; Aggarwal et al., 2001; François et al., 2007; Radovanović et al., 2010).

In order to deal with the curse of dimensionality issue, different dimensionality reduction techniques (Fodor, 2002; Van der Maaten et al., 2009) have been used. These include feature selection or the projection of data into a lower-dimensional space.

In feature selection, the assumption is that all available features are not relevant to represent the underlying concept of given data. The task of feature selection is to remove irrelevant features and retain the most salient features (Guyon and Elisseeff, 2003). Feature selection requires a search strategy over the possible combinations of features and evaluation criteria to be optimised (Molina et al., 2002).

Projecting data into a lower-dimensional subspace is another widely-used dimensionality reduction technique. The central assumption here is that high-dimensional data often lie in a low-dimensional manifold. There are different linear and non-linear projection-based dimensionality reduction techniques (Fodor, 2002; Van der Maaten et al., 2009). Principal component analysis (PCA) (Jolliffe, 2005), kernel PCA (KPCA) (Cristianini and Shawe-Taylor, 2000) and random projection (Kaski, 1998; Achlioptas, 2001) are commonly-used projection-based dimensionality reduction techniques.

Note that distance metric learning (Weinberger et al., 2006; Yang, 2006; Wang and Sun, 2015) can also be viewed as a dimensionality reduction technique, because data are often projected into a lower-dimensional new space $\mathcal{Z}$ where the task-specific performance of a distance measure can be maximised in the given dataset.

Although the projection-based dimensionality reduction techniques including distance metric learning maintain the geometric interpretation of data in the projected space, it is difficult to interpret the meaning of the new dimensions. Furthermore, techniques such as KPCA have high computational complexity in large datasets as they require the calculation of pairwise distances of data instances.

1.4 Thesis aims

Motivated by the limitations of distance measures discussed in Section 1.3, psychologists' arguments and existing non-distance-based approaches, such as Random Forest (Shi and

Horvath, 2006), ReFeat (Zhou et al., 2012), rank difference (Conover and Iman, 1981) and Lin's probabilistic measure (Lin, 1998), this thesis investigates a fully data-dependent similarity measure (where even self-similarity is data-dependent), which is also robust to units and scales of measurement.

Data-dependent self-similarity and robustness to units and scales of measurement are important characteristics of a measure for the measurement of similarities between data instances in high-dimensional spaces for the following reasons:

1. High-dimensional data often lie in low-dimensional subspaces and many data instances have the same value in many dimensions. Data-dependent self-similarity is particularly useful in this case to differentiate between instances. In each dimension, having the same feature value which is very frequent (high probability) in a given dataset contributes less in the overall similarity of two instances than having the same feature value which is rare (low probability) in the dataset. However, they both contribute equally in distance-based similarity measures.
2. In a distance-based similarity measure, the effect of the difference in units and scales of measurement is more severe in high-dimensional datasets than in low-dimensional datasets. The effect of similarity in differentiating instances is weakened if the unit or scale of measurement is vastly different in different dimensions, and the degree of weakening increases as the number of dimensions increases.

A similarity measure which is fully data-dependent and robust to different units and scales will be less affected by the curse of dimensionality and is expected to produce better task-specific performance than distance-based similarity measures in high-dimensional datasets.

Therefore, this thesis aims to:

1. *Develop a new (dis)similarity measure which is fully data-dependent and robust to units and scales of measurement.*
2. *Verify that the new measure (a) produces better task-specific performance than widely used distance-based measures and existing data-dependent measures in different datasets from different application domains; and (b) is robust to units and scales of measurement.*
3. *Test the hypothesis that the new measure produces better task-specific performance than distance-based similarity measures in high-dimensional spaces (i.e., it is less affected by the curse of dimensionality).*

1.5 Thesis contributions

This thesis makes the following main contributions:

1. It introduces a generic data-dependent dissimilarity measure called "m_p-dissimilarity" ($p \geq 0$), where the similarity of two instances in each dimension is based on the probability data mass between them instead of their spatial distance. In comparison to

distance measures, m_p-dissimilarity has the following two distinguishing characteristics:

(a) Fully data-dependent dissimilarity including self-dissimilarity. 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 spatial distance and even if the spatial distance is zero.

(b) Robust to units and scales of measurement. In each dimension, the dissimilarity of two instances is based on the data mass between them which is robust to the monotonic transformation of data.

2. It analyses the characteristics and relationships of m_p-dissimilarity with existing data-dependent measures and reveals that m_p-dissimilarity is a generic data-dependent measure. Existing data-dependent measures of rank difference and Lin's probabilistic measure are its special cases with $p > 0$ and $p = 0$, respectively. These special cases have data-independent self-dissimilarities but the general version has data-dependent self-dissimilarity.

3. It evaluates the performance of m_p-dissimilarity with existing widely-used data-independent (distance-based) and data-dependent (dis)similarity measures in a wide range of low-to high-dimensional datasets from different application areas, including text, image and music. The empirical results show that m_p-dissimilarity often produces better or at least more consistent task-specific performances than other contenders across different datasets.

The superior performance of m_p-dissimilarity over distance-based measures in high-dimensional datasets, such as bag-of-words (BoW) text datasets, confirms that a measure which is fully data-dependent and robust to units and scales of measurement is less affected by the curse of dimensionality than distance-based similarity measures.

In addition to the primary contribution of developing the data-dependent dissimilarity measure of m_p-dissimilarity, this thesis makes the following secondary contributions:

i. It introduces the notion of "relative mass" to improve the performance of ReFeat. Using relative mass, the relevance of an instance $\mathbf{x}$ with respect to a query $\mathbf{q}$ is measured in each tree as the ratio of data mass in two nodes: (a) the leaf node in which $\mathbf{q}$ falls; and (b) the deepest node shared by both $\mathbf{x}$ and $\mathbf{q}$ in the tree. The relevance measure based on relative mass guarantees that two relevant instances lie in the same local neighbourhood. The concept of relative mass also overcomes a weakness of iForest in anomaly detection being not able to detect local anomalies. The similarity measure based on the relative mass motivated the work in m_p-dissimilarity.

ii. It improves the runtime complexities of rank difference and Lin's probabilistic similarity measure to be of the same order as those of distance-based measures and m_p-dissimilarity. This improvement makes them feasible to run in large datasets.

iii. It identifies the shortcomings of the underlying assumptions of term-weighting schemes (Salton and Buckley, 1988; Manning et al., 2008) employed in existing BoW documents similarity measures such as cosine (Salton and Buckley, 1988), and provides

an alternative assumption which is more congruous with the requirements of inter-document similarity measurement. Based on the new assumption, a simplified version of the m_p-dissimilarity measure called Sp is introduced for BoW inter-document similarity measurement. Sp does not require any term weighting and yet produces better or more consistent task-specific performance than existing measures using state-of-the-art term-weighting schemes.

The concept of a fully data-dependent dissimilarity measure based on probability mass has led to subsequent research. To estimate $s(\mathbf{x}, \mathbf{y})$, Ting et al. (2016) use data mass in the deepest node shared by both $\mathbf{x}$ and $\mathbf{y}$ in each tree in iForest and demonstrate that the mass-based similarity measure produces better task-specific performance than distance-based similarity measures in anomaly detection, clustering and multilabel classification.

1.6 Thesis structure

This is a thesis including publications and it is organised as follows:

Chapter 1 introduces the context of the thesis. It outlines the motivations and contributions of the thesis.

Chapter 2 presents a paper published in the proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) 2014 (Aryal et al., 2014a). The paper introduces the notion of "relative mass" and proposes two algorithms called ReMass-iForest and ReMass-ReFeat to overcome weaknesses of iForest (Liu et al., 2008, 2012) in anomaly detection and content-based information retrieval tasks, respectively. ReMass-ReFeat motivated the work in data-dependent dissimilarity measure presented in Chapter 3.

Chapter 3 presents a paper accepted for publication (published online) in the Knowledge and Information Systems (KAIS) journal in 2017 (Aryal et al., 2017), which is an extended version of the paper published in the proceedings of the IEEE International Conference on Data Mining (ICDM) 2014 (Aryal et al., 2014b) included in Appendix A. The paper proposes a data-dependent dissimilarity measure called m_p-dissimilarity (where $p > 0$) as an effective alternative to distance measures, particularly in high-dimensional spaces. It shows that the proposed m_p-dissimilarity measure produces similar or better task-specific performance than widely-used distance measures such as ℓ_p-norm and cosine distance across a wide range of medium-to high-dimensional datasets.

Chapter 4 presents a paper submitted to the Data Mining and Knowledge Discovery (DMKD) journal. This paper generalises m_p-dissimilarity where p is allowed to be 0 by introducing m_0-dissimilarity. By examining the relationships and characteristics of different data-dependent measures, the paper shows that m_p-dissimilarity is a generalised data-dependent similarity measure of which the rank difference and Lin's probabilistic measure are special cases with $p > 0$ and $p = 0$, respectively. It also improves the runtime complexities of rank difference and Lin's probabilistic measure to be of the same order as that of distance-based measures and m_p-dissimilarity. Empirical evaluation reveals that the fully data-dependent measure of m_p-dissimilarity, which is robust to units and scales of

measurement, is more effective than other existing data-dependent and data-independent similarity measures.

Chapter 5 presents a paper submitted to the Computational Intelligence (COIN) journal which is an extended version of the paper published in the proceedings of the 11th Asia Information Retrieval Society (AIRS) conference 2015 (Aryal et al., 2015) included in Appendix B. This paper identifies the shortcomings of the underlying assumptions of term-weighting schemes employed in existing BoW document similarity measures, and provides an alternative assumption which is more congruous with the requirements of inter-document similarity measurement. Based on the new assumption, it introduces a simple but effective BoW inter-document similarity measure called Sp. Unlike existing measures, the explicit adjustment of document vectors through term weighting is not required in Sp. It is a simplified version of m_0-dissimilarity.

Chapter 6 concludes the thesis and provides some directions for future research.

References

Achlioptas, D. (2001). Database-friendly random projections, *In Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, ACM, New York, USA, pp. 274–281.

Aggarwal, C. C., Hinneburg, A. and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space, *In Proceedings of the International Conference on Database Theory*, Springer, Berlin Heidelberg, pp. 420–434.

Aha, D. and Kibler, D. (1991). Instance-based learning algorithms, *Machine Learning* **6**: 37–66.

Aryal, S., Ting, K. M., Haffari, G. and Washio, T. (2014b). Mp-dissimilarity: A data dependent dissimilarity measure, *In Proceedings of the IEEE International Conference on Data Mining*, IEEE, pp. 707–712.

Aryal, S., Ting, K. M., Haffari, G. and Washio, T. (2015). Beyond tf-idf and cosine distance in document dissimilarity measures, *In Proceedings of the 11th Asia Information Retrieval Societies Conference*, Springer, Cham, pp. 400–406.

Aryal, S., Ting, K. M., Washio, T. and Haffari, G. (2017). Data-dependent dissimilarity measure: an effective alternative to geometric distance measures, *Knowledge and Information Systems* pp. 1–28, doi:10.1007/s10115–017–1046–0.

Aryal, S., Ting, K. M., Wells, J. R. and Washio, T. (2014a). Improving iForest with Relative Mass, *In Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, Cham, pp. 510–521.

Bay, S. D. and Schwabacher, M. (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule, *In Proceedings of the Ninth ACM*

SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 29–38.

Beyer, K. S., Goldstein, J., Ramakrishnan, R. and Shaft, U. (1999). When is "nearest neighbour" meaningful?, *In Proceedings of the 7th International Conference on Database Theory*, Springer-Verlag, London, UK, pp. 217–235.

Breunig, M. M., Kriegel, H.-P., Ng, R. T. and Sander, J. (2000). LOF: Identifying Density-Based Local Outliers, *In Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 93–104.

Conover, W. J. and Iman, R. L. (1981). Rank transformations as a bridge between parametric and nonparametric statistics, *The American Statistician* **35**(3): 124–129.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, New York, USA.

Deza, M. M. and Deza, E. (2009). *Encyclopedia of Distances*, Springer, Berlin Heidelberg.

Duda, R. O., Hart, P. E. and Stork, D. G. (2000). *Pattern Classification (2nd Edition)*, Wiley-Interscience, New York, USA.

Fodor, I. (2002). A survey of dimension reduction techniques, *Technical Report UCRL-ID-148494*, Lawrence Livermore National Laboratory, University of California, USA.

François, D., Wertz, V. and Verleysen, M. (2007). The concentration of fractional distances, *IEEE Transactions on Knowledge and Data Engineering* **19**(7): 873–886.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection, *Journal of Machine Learning Research* **3**: 1157–1182.

Han, J. and Kamber, M. (2006). *Data mining concepts and techniques*, Morgan Kaufmann Publishers, San Francisco, USA.

Jolliffe, I. (2005). *Principal component analysis*, Wiley Online Library.

Kaski, S. (1998). Dimensionality reduction by random mapping: fast similarity computation for clustering, *In Proceedings of the IEEE World Congress on Computational Intelligence, IEEE International Joint Conference on Neural Networks.*, Vol. 1, pp. 413–418.

Krumhansl, C. L. (1978). Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density, *Psychological Review* **85**(5): 445–463.

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, USA, pp. 296–304.

Liu, F. T., Ting, K. M. and Zhou, Z.-H. (2012). Isolation-based anomaly detection, *ACM Transactions on Knowledge Discovery from Data* **6**(1): 3:1–3:39.

Liu, F., Ting, K. M. and Zhou, Z.-H. (2008). Isolation forest, *In Proceedings of the Eighth IEEE International Conference on Data Mining*, pp. 413–422.

Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations, *In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297.

Manning, C. D., Raghavan, P. and Schtze, H. (2008). *Introduction to Information Retrieval*, Cambridge University Press, New York, USA.

Molina, L. C., Belanche, L. and Nebot, A. (2002). Feature selection algorithms: A survey and experimental evaluation, *In Proceedings of the IEEE International Conference on Data Mining*, IEEE Computer Society, Washington DC, USA, pp. 306–313.

Radovanović, M., Nanopoulos, A. and Ivanović, M. (2010). Hubs in space: Popular nearest neighbours in high-dimensional data, *Journal of Machine Learning Research* **11**: 2487–2531.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval, *Information Processing and Management* **24**(5): 513–523.

Shi, T. and Horvath, S. (2006). Unsupervised learning with random forest predictors, *Journal of Computational and Graphical Statistics* **15**(1): 118–138.

Stevens, S. S. (1946). On the theory of scales of measurement, *Science* **103**(2684): 677–680.

Tan, P.-N., Steinbach, M. and Kumar, V. (2006). *Introduction to Data Mining*, Addison-Wesley Longman Publishing Corporation, Boston, USA.

Ting, K. M., Zhu, Y., Carman, M., Zhu, Y. and Zhou, Z.-H. (2016). Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure, *In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1205–1214.

Tversky, A. (1977). Features of Similarity, *Psychological Review* **84**(2): 327–352.

Van der Maaten, L., Postma, E. O. and Van den Herik, H. J. (2009). Dimensionality reduction: A comparative review, *Technical Report TiCC TR 2009-005*, Tilburg centre for Creative Computing, Tilburg University, The Netherlands.

Wang, F. and Sun, J. (2015). Survey on distance metric learning and dimensionality reduction in data mining, *Data Mining and Knowledge Discovery* **29**(2): 534–564.

Weinberger, K., Blitzer, J. and Saul, L. (2006). Distance metric learning for large margin nearest neighbour classification, *In Proceedings of the 18th International Conference on Neural Information Processing Systems*, MIT Press, pp. 1473–1480.

Weinberger, K. Q. and Saul, L. K. (2009). Distance metric learning for large margin nearest neighbour classification, *Journal of Machine Learning Research* **10**: 207–244.

Yang, L. (2006). Distance metric learning: A comprehensive survey, *Technical report*, Michigan State University, USA.

Zhou, G.-T., Ting, K. M., Liu, F. T. and Yin, Y. (2012). Relevance feature mapping for content-based multimedia information retrieval, *Pattern Recognition* **45**(4): 1707–1720.

Chapter 2

Improving the performance of ReFeat using relative mass

The idea of the data-dependent dissimilarity measure discussed in this thesis is motivated by the superior performance of ReFeat over distance-based similarity measures in the content-based multimedia information retrieval (CBMIR) task. ReFeat uses a non-distance-based ranking measure based on iForest, which was originally developed to solve anomaly detection (AD) problems. Although iForest has been shown to be effective in both CBMIR and AD tasks, this thesis has identifies its limitations in both tasks.

This chapter introduces the notion of "relative mass" and proposes two algorithms, ReMass-iForest and ReMass-ReFeat, as effective alternatives to iForest and ReFeat in AD and CBMIR tasks, respectively. ReMass-ReFeat motivated the work on data-dependent dissimilarity measure to be discussed in Chapter 3.

The work on relative mass is reported in the following published paper:

Aryal, S., Ting, K. M., Wells, J. R. and Washio, T. (2014), Improving iForest with Relative mass, In *Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) 2014*, Springer, Cham, pp. 510-521.

This chapter is a copy of the paper published in the conference proceedings. In order to generate a consistent presentation within the thesis, the format and some notations or symbols used have been changed, minor grammar and spelling mistakes have been corrected, and sections of the published paper have been renumbered.

The original published version of the paper is available at Springer via `https://doi.org/10.1007/978-3-319-06605-9_42`

Improving iForest with Relative Mass

Sunil Aryal[†], Kai Ming Ting[†], Jonathan R. Wells[†] and Takashi Washio[‡]

[†]Monash University, Victoria, Australia
[‡]Osaka University, Osaka, Japan

Abstract:

iForest uses a collection of isolation trees to detect anomalies. While it is effective in detecting global anomalies, it fails to detect local anomalies in datasets with multiple clusters of normal instances, because the local anomalies are masked by normal clusters of similar density and they become less susceptible to isolation. In this paper, we propose a very simple but effective solution to overcome this limitation by replacing the global ranking measure based on path length with a local ranking measure based on relative mass that takes local data distribution into consideration. We demonstrate the utility of relative mass by improving the task-specific performance of iForest in anomaly detection and information retrieval tasks.

Keywords: Relative mass, iForest, ReFeat, anomaly detection

2.1 Introduction

Data-mining tasks such as anomaly detection (AD) and information retrieval (IR) require a ranking measure in order to rank data instances. Distance-or density-based methods are widely used to rank instances in these tasks. The main problem of these methods is that they are computationally expensive in large datasets because of their high time complexities.

Isolation Forest (iForest) (Liu et al., 2008) is an anomaly detector that does not use distance or density measures. It performs an operation to isolate each instance from the rest of the instances in a given dataset. Because anomalies have characteristics of being 'few and different', they are more susceptible to isolation in a tree structure than normal

instances. Therefore, anomalies have shorter average path lengths than those of normal instances over a collection of isolation trees (iTrees).

Although iForest has been shown to perform well (Liu et al., 2008), we have identified a weakness in detecting local anomalies in datasets with multiple clusters of normal instances, because the local anomalies are masked by normal clusters of similar density; thus they become less susceptible to isolation using iTrees. In other words, iForest cannot detect local anomalies because the path length measures the degree of anomaly globally. It does not consider how isolated an instance is from its local neighbourhood.

iForest has its foundation in mass estimation (Ting et al., 2013a). Ting et al. (2013a) have shown that the path length is a proxy for mass in a tree-based implementation. On this basis, we consider that iForest's inability to detect local anomalies can be overcome by replacing the global ranking measure based on path length with a local ranking measure based on relative mass using the same iTrees. In general, the relative mass of an instance is the ratio of data mass in two regions covering the instance, where one region is a subset of the other. The relative mass measures the degree of anomaly locally by considering the data distribution in the local regions (superset and subset) covering an instance.

In addition to AD, we show the generality of relative mass in IR that overcomes the limitation of a recent IR system called ReFeat (Zhou et al., 2012), which uses iForest as a core ranking model. Even though ReFeat performs well in content-based multimedia information retrieval (CBMIR) (Zhou et al., 2012), the ranking scheme based on path length does not guarantee that two instances with a similar ranking score are in the same local neighbourhood. The new ranking scheme based on relative mass provides such a guarantee.

The contributions of this paper are as follows:

1. It introduces relative mass as a ranking measure.
2. It proposes ways to apply relative mass instead of path length to overcome the weaknesses of iForest in AD and IR.
3. It demonstrates the utility of relative mass in AD and IR by improving the task-specific performance of iForest and ReFeat using exactly the same implementation of iTrees as that employed in iForest.

The rest of the paper is organised as follows. Section 2.2 introduces the notion of relative mass and proposes ways to apply it to AD and IR. Section 2.3 provides the empirical evaluation, followed by conclusions in the last section.

2.2 Relative mass: A mass-based local ranking measure

Rather than using the global ranking measure based on path length in iForest, an instance can be ranked using a local ranking measure based on relative mass w.r.t its local neighbourhood. In a tree structure, the relative mass of an instance is computed as the ratio of mass in two nodes along the path the instance traverses from the root to a leaf node. The two nodes used in the calculation of relative mass depend on the task-specific requirements.

- In AD, we are interested in the relative mass of $\mathbf{x}$ w.r.t its local neighbourhood. Hence, the relative mass is computed as the ratio of mass in the immediate parent node and the leaf node where $\mathbf{x}$ falls.
- In IR, we are interested in the relative mass of $\mathbf{x}$ w.r.t to a query $\mathbf{q}$. Hence, the relative mass is computed as the ratio of mass of the leaf node where $\mathbf{q}$ falls and the lowest node where $\mathbf{x}$ and $\mathbf{q}$ shared along the path $\mathbf{q}$ traverses.

We converted iForest (Liu et al., 2008) and ReFeat (Zhou et al., 2012) using the relative mass, and named the resultant relative mass versions ReMass-iForest and ReMass-ReFeat, respectively. We describe iForest and ReMass-iForest in AD in Section 2.2.1, and ReFeat and ReMass-ReFeat in IR in Section 2.2.2.

2.2.1 Anomaly Detection: iForest and ReMass-iForest

In this subsection, we first discuss iForest and its weakness in detecting local anomalies and introduce the new anomaly detector, ReMass-iForest, based on relative mass to overcome the weakness.

iForest

Given an M-variate database of N instances ($D = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(N)}\}$), iForest (Liu et al., 2008) constructs t iTrees ($\Gamma_1, \Gamma_2, \cdots, \Gamma_t$). Each Γ_i is constructed from a small random sub-sample ($\mathcal{D}_i \subset D$, $|\mathcal{D}_i| = \psi < N$) by recursively dividing it into two non-empty nodes through a randomly-selected attribute and split point. A branch stops splitting when the height reaches the maximum (H_{max}) or the number of instances in the node is less than $MinPts$. The default values used in iForest are $H_{max} = \log_2(\psi)$ and $MinPts = 1$. The anomaly score is estimated as the average path length over t iTrees as follows:

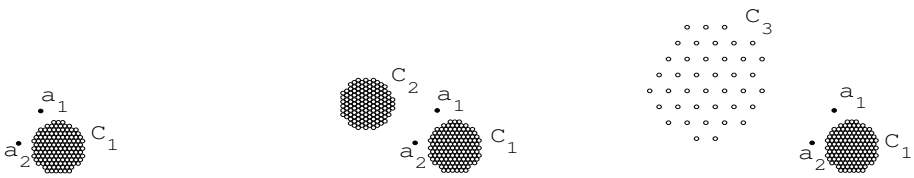
$$L(\mathbf{x}) = \frac{1}{t} \sum_{i=1}^{t} l_i(\mathbf{x}) \tag{2.1}$$

where, $l_i(\mathbf{x})$ is the path length of $\mathbf{x}$ in Γ_i

As anomalies are likely to be isolated early, they have shorter average path lengths. Once all instances in the given dataset have been scored, the instances are sorted in ascending order of scores. The instances at the top of the list are reported as anomalies.

iForest runs very fast because it does not require distance calculation and each iTree is constructed from a small random sub-sample of data.

iForest is effective in detecting global anomalies (e.g., a_1 and a_2 in Figures 2.1(a) and 2.1(b)) because they are more susceptible to isolation in iTrees. However, it fails to detect local anomalies (e.g., a_1 and a_2 in Figure 2.1(c)) as they are less susceptible to isolation in iTrees. This is because the local anomalies and the normal cluster C_3 have about the same density. Some fringe instances in the normal cluster C_3 will have shorter average path lengths than those for a_1 and a_2.



(a) Global anomalies (b) Global anomalies (c) Local anomalies

Figure 2.1: Global and local anomalies. Note that both anomalies a_1 and a_2 are exactly the same instances in Figure 2.1(a), 2.1(b) and 2.1(c). In Figure 2.1(a) and Figure 2.1(b), a_1 and a_2 have lower density than that in the normal clusters C_1 and C_2. In Figure 2.1(c), a_1, a_2 and the normal cluster C_3 have the same density but a_1 and a_2 are anomalies relative to the normal cluster C_1 with a higher density.

ReMass-iForest

In each iTree Γ_i, the anomaly score of an instance $\mathbf{x}$ w.r.t its local neighbourhood, $s_i(\mathbf{x})$, can be estimated as the ratio of data mass as follows:

$$s_i(\mathbf{x}) = \frac{m(\breve{\Gamma}_i(\mathbf{x}))}{m(\Gamma_i(\mathbf{x})) \times \psi} \tag{2.2}$$

where $\Gamma_i(\mathbf{x})$ is the leaf node in Γ_i in which $\mathbf{x}$ falls, $\breve{\Gamma}_i(\mathbf{x})$ is the immediate parent of $\Gamma_i(\mathbf{x})$, and $m(\cdot)$ is the data mass of a tree node. ψ is a normalisation term which is the training data size used to generate Γ_i.

$s_i(\cdot)$ is in $(0, 1]$. The higher the score, the greater the likelihood of $\mathbf{x}$ being an anomaly. Unlike $l_i(\mathbf{x})$ in iForest, $s_i(\mathbf{x})$ measures the degree of anomaly locally.

The final anomaly score can be estimated as the average of local anomaly scores over t iTrees as follows:

$$S(\mathbf{x}) = \frac{1}{t}\sum_{i=1}^{t} s_i(\mathbf{x}) \tag{2.3}$$

Once every instance in the given dataset has been scored, instances can be ranked in descending order of their anomaly scores. The instances at the top of the list are reported as anomalies.

Relation to LOF and DEMass-LOF

The idea of relative mass in ReMass-iForest has some relation to the idea of relative density in Local Outlier Factor (LOF) (Breunig et al., 2000). LOF uses k nearest neighbours to estimate density $\bar{f}_k(\mathbf{x}) = \frac{|NN(\mathbf{x},k)|}{n\sum_{\mathbf{x}' \in NN(\mathbf{x},k)} distance(\mathbf{x},\mathbf{x}')}$, where $NN(\mathbf{x},k)$ is the set of k nearest neighbours of $\mathbf{x}$. It estimates its anomaly score as the ratio of the average density of $\mathbf{x}$'s k nearest neighbours to $\bar{f}_k(\mathbf{x})$. In LOF, the local neighbourhood is defined by k nearest neighbours which requires distance calculation. In contrast, in ReMass-iForest, the local neighbourhood is the immediate parent in iTrees. It does not require distance calculation.

Table 2.1: Ranking measure and complexities (time and space) of ReMass-iForest, iForest, DEMass-LOF and LOF.

	ReMass-iForest	iForest	DEMass-LOF	LOF
Ranking Measure	$\frac{1}{t\psi}\sum_{i=1}^{t}\frac{m(\breve{\Gamma}_i(\mathbf{x}))}{m(\Gamma_i(\mathbf{x}))}$	$\frac{1}{t}\sum_{i=1}^{t} l_i(\mathbf{x})$	$\frac{\sum_{i=1}^{t}\frac{m(\breve{\Gamma}_i(\mathbf{x}))}{\breve{v}_i}}{\sum_{i=1}^{t}\frac{m(\Gamma_i(\mathbf{x}))}{v_i}}$	$\frac{\sum_{\mathbf{x}'\in NN(\mathbf{x},k)}\frac{\bar{f}_k(\mathbf{x}')}{\lvert NN(\mathbf{x},k)\rvert}}{\bar{f}_k(\mathbf{x})}$
Time Complexity	$O(t(N+\psi)\log\psi)$	$O(t(N+\psi)\log\psi)$	$O(t(N+\psi)bM)$	$O(MN^2)$
Space Complexity	$O(t\psi)$	$O(t\psi)$	$O(tM\psi)$	$O(MN)$

$\breve{v}_i$ and v_i are the volumes of nodes $\breve{\Gamma}_i(\mathbf{x})$ and $\Gamma_i(\mathbf{x})$, respectively.

DEMass-LOF (Ting et al., 2013b) computes the same anomaly score as LOF from trees, without distance calculation. The idea of relative density of parent and leaf nodes is used in DEMass-LOF. It constructs a forest of t balanced binary trees where the height of each tree is $b \times M$ (b is a parameter that determines the level of division on each attribute and M is the number of attributes). It estimates its anomaly score as the ratio of average density of the parent node to the average density of the leaf node where $\mathbf{x}$ falls. The density of a node is estimated as the ratio of mass to volume. It uses mass to estimate density and ranks instances based on the density ratio. Like iForest, it is fast because no distance calculation is involved, but it has limitation in dealing with problems with even a moderate number of dimensions, because each tree has $2^{(b\times M)}$ leaf nodes.

In contrast to LOF and DEMass-LOF, ReMass-iForest does not require density estimation, but uses relative mass directly in order to estimate the local anomaly score from each iTree.

The ranking measure and complexities (time and space) of ReMass-iForest, iForest, DEMass-LOF and LOF are summarised in Table 2.1.

2.2.2 Information Retrieval: ReFeat and ReMass-ReFeat

In this subsection, we first describe how ReFeat uses iForest in IR and its weakness. Then, we introduce a new IR system, ReMass-ReFeat, based on the relative mass to overcome the weakness.

ReFeat

Given a query instance $\mathbf{q}$, ReFeat (Zhou et al., 2012) assigns a weight $w_i(\mathbf{q}) = \frac{l_i(\mathbf{q})}{c} - 1$ (where c is a normalisation constant) to each Γ_i. The relevance feedback process (Rui et al., 1998) allows the user to refine the retrieved result by providing some 'relevant' and 'irrelevant' examples for the query. Let $\mathcal{Q} = \mathcal{P} \cup \mathcal{N}$ be a set of feedback instances to the query $\mathbf{q}$ where $\mathcal{P}$ and $\mathcal{N}$ are the sets of positive and negative feedbacks, respectively. Note

that $\mathcal{P}$ includes $\mathbf{q}$. In a relevance feedback round, ReFeat assigns a weight to Γ_i using positive and negative feedback instances as: $w_i(\mathcal{Q}) = \frac{1}{|\mathcal{P}|}\sum_{\mathbf{y}^+\in\mathcal{P}} w_i(\mathbf{y}^+) - \gamma\frac{1}{|\mathcal{N}|}\sum_{\mathbf{y}^-\in\mathcal{N}} w_i(\mathbf{y}^-)$, where $0 \leq \gamma \leq 1$ is a trade-off parameter for the relative contribution of positive and negative feedbacks. The relevance of $\mathbf{x}$ w.r.t $\mathcal{Q}$ is estimated as the weighted average of its path lengths over t iTrees as follows:

$$R_{ReFeat}(\mathbf{x}|\mathcal{Q}) = \frac{1}{t}\sum_{i=1}^{t}(w_i(\mathcal{Q}) \times l_i(\mathbf{x})) \tag{2.4}$$

Although ReFeat has been shown to have superior retrieval performance over other existing methods in CBMIR, the ranking scheme does not guarantee that two instances having similar ranking scores are in the same local neighbourhood. Two instances can have a similar score if they have equal path lengths in an iTree, even though they lie in two different branches which share few common nodes. This effect will degrade the performance of ReFeat, especially when the tree height (h) is increased. Hence, ReFeat must use a low h (2 or 3) in order to reduce this weakness. The superior performance of ReFeat is mainly due to its large ensemble size ($t = 1000$). We discuss the effect of h and t in ReFeat in Section 2.3.2. ReFeat does not consider the positions of instances in the feature space, as it computes the path length in iTrees.

ReMass-ReFeat

In each iTree Γ_i, the relevance of $\mathbf{x}$ w.r.t. $\mathbf{q}$, $r_i(\mathbf{x}|\mathbf{q})$, is estimated using relative mass as follows:

$$r_i(\mathbf{x}|\mathbf{q}) = \frac{m(\Gamma_i(\mathbf{q}))}{m(\Gamma_i(\mathbf{x},\mathbf{q}))} \tag{2.5}$$

where $\Gamma_i(\mathbf{x},\mathbf{q})$ is the smallest region in Γ_i where $\mathbf{x}$ and $\mathbf{q}$ appear together.

In Eqn 2.5, the numerator corresponds with $w_i(\mathbf{q})$ in ReFeat. The denominator term measures how relevant $\mathbf{x}$ is to $\mathbf{q}$. In contrast, ReFeat's $l_i(\mathbf{x})$ is independent of $\mathbf{q}$ (it does not examine whether $\mathbf{x}$ and $\mathbf{q}$ are in the same locality (Zhou et al., 2012)), whereas $m(\Gamma_i(\mathbf{x},\mathbf{q}))$ measures how close $\mathbf{x}$ and $\mathbf{q}$ are in the feature space. In each Γ_i, $r_i(\mathbf{x}|\mathbf{q})$ is in the range of $(0, 1]$. The higher the score, the more the relevance of $\mathbf{x}$ w.r.t $\mathbf{q}$. If $\mathbf{x}$ and $\mathbf{q}$ lie in the same leaf node in Γ_i, $r_i(\mathbf{x}|\mathbf{q})$ is 1. This relevance measure gives a high score to an instance which lies deeper in the branch where $\mathbf{q}$ lies.

The final relevance score of $\mathbf{x}$ w.r.t $\mathbf{q}$, $R(\mathbf{x}|\mathbf{q})$, is the average over t iTrees:

$$R(\mathbf{x}|\mathbf{q}) = \frac{1}{t}\sum_{i=1}^{t} r_i(\mathbf{x}|\mathbf{q}) \tag{2.6}$$

Once the relevance score of each instance is estimated, the scores can be sorted in descending order. The instances at the top of the list are regarded as the most relevant instances to $\mathbf{q}$.

ReMass-ReFeat estimates the relevance score with relevance feedback as follows:

Table 2.2: Time and space complexities of ReMass-ReFeat and ReFeat

	ReMass-ReFeat	ReFeat
Time Complexity	$O(t(N+\psi)\log\psi)$ (Model building) $O(t(N+\log\psi))$ (On-line query)	$O(t(N+\psi)\log\psi)$ (Model building) $O(t(N+\log\psi))$ (On-line query)
Space Complexity	$O(t(N+\psi))$	$O(N+t\psi)$

$$R(\mathbf{x}|\mathcal{Q}) = \frac{1}{|\mathcal{P}|}\sum_{\mathbf{y}^+\in\mathcal{P}} R(\mathbf{x}|\mathbf{y}^+) - \gamma\frac{1}{|\mathcal{N}|}\sum_{\mathbf{y}^-\in\mathcal{N}} R(\mathbf{x}|\mathbf{y}^-) \tag{2.7}$$

Note that Eqns 2.5 and 2.6 do not make use of any distance or similarity measure, and $R(\mathbf{x}|\mathbf{q})$ is not a metric as it does not satisfy all metric axioms. It has the following characteristics. For $\mathbf{x}, \mathbf{y} \in D$,

i. $0 < R(\mathbf{x}|\mathbf{y}) \leq 1$ (Non-negativity)

ii. $R(\mathbf{x}|\mathbf{x}) = R(\mathbf{y}|\mathbf{y}) = 1$ (Equal self-similarity; maximal similarity)

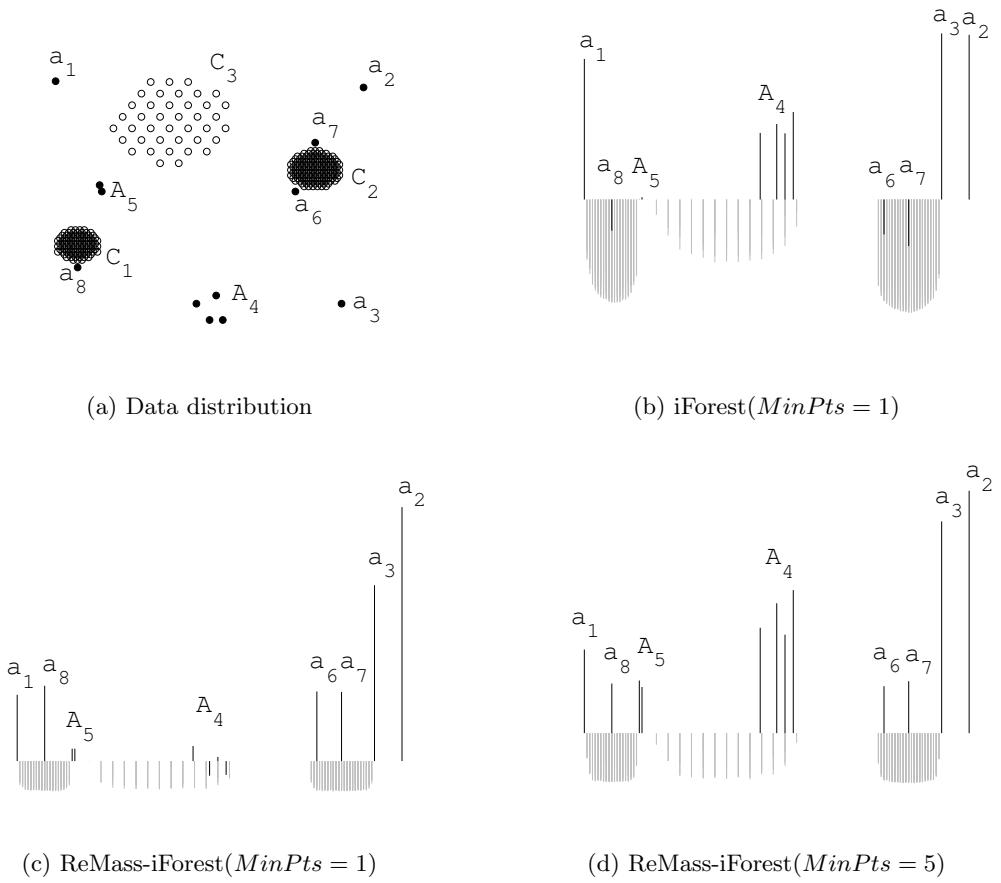
iii. $R(\mathbf{x}|\mathbf{y}) \neq R(\mathbf{y}|\mathbf{x})$ (Asymmetric)

Note that ReMass-ReFeat and ReFeat have the same time complexities. If the indices of data instances falling in each node are recorded in the modelling stage, the joint mass of $\mathbf{q}$ and every $\mathbf{x} \in D$ can be estimated in one search from the root to $\Gamma_i(\mathbf{q})$ in each tree. However, it will increase the space complexity as it requires to store N indices in each iTree. The time and space complexities of ReMass-ReFeat and ReFeat are provided in Table 2.2.

2.3 Empirical evaluation

In this section, we evaluate the utility of relative mass in AD and CBMIR tasks. In AD, we compared ReMass-iForest with iForest (Liu et al., 2008), DEMass-LOF (Ting et al., 2013b) and LOF (Breunig et al., 2000). In CBMIR, we compared ReMass-ReFeat with ReFeat (Zhou et al., 2012) and the other existing CBMIR systems: MRBIR (He et al., 2004), InstRank (Giacinto and Roli, 2005) and Qsim (Zhou and Dai, 2006). Both the AD and CBMIR experiments were conducted in unsupervised learning settings. The labels of instances were not used in the model building process. They were used as the ground truth in the evaluation stage. The AD results were measured in terms of the area under the ROC curve (AUC). In CBMIR, the precision at the top 50 retrieved results (P@50) (Zhou et al., 2012) was used as the performance measure. The presented result was the average over 20 runs for all randomised algorithms. A two-standard-error significance test was conducted to check whether the difference in performance of the two methods was significant.

We used the same MATLAB implementation of iForest provided by the authors of ReFeat (Zhou et al., 2012), the JAVA implementation of DEMass-LOF in the WEKA (Hall et al., 2009) platform, and the JAVA implementation of LOF in the ELKI (Achtert et al., 2011) platform.



(a) Data distribution

(b) iForest($MinPts = 1$)

(c) ReMass-iForest($MinPts = 1$)

(d) ReMass-iForest($MinPts = 5$)

Figure 2.2: Anomaly scores by iForest and ReMass-iForest using $t = 100, \psi = 256$. Note that in anomaly score plots, instances are represented by their values in x_1 dimension. Anomalies are represented by black lines and normal instances are represented by grey lines. The height of lines represents the anomaly scores. In order to differentiate the scores of normal and anomaly instances, the maximum score for normal instances is subtracted from the anomaly scores so that all normal instances have scores of zero or less.

We present the empirical evaluation results in the following two subsections.

2.3.1 Anomaly Detection: ReMass-iForest versus iForest

In the first experiment, we used a synthetic dataset to demonstrate the strength of ReMass-iForest over iForest to detect local anomalies. The dataset has 263 normal instances in three clusters and 12 anomalies representing global, local and clustered anomalies. The data distribution is shown in Figure 2.2(a). Instances a_1, a_2 and a_3 are global anomalies; four instances in A_4 and two instances in A_5 are clustered anomalies; and a_6, a_7 and a_8 are local anomalies; C_1, C_2 and C_3 are normal instances in three clusters of varying densities.

Figures 2.2(b)-2.2(d) show the anomaly scores of all data instances obtained from iForest and ReMass-iForest. With iForest, local anomalies a_6, a_7 and a_8 had lower anomaly scores than some normal instances in C_3; and it produced an AUC of 0.98. In contrast, ReMass-iForest had ranked local anomalies a_6, a_7, a_8 higher than any instances in normal clusters C_1, C_2 and C_3 along with global anomalies a_1, a_2 and a_3. However, ReMass-iForest with $MinPts = 1$ had some problem with ranking clustered anomalies in A_4 and produced

Table 2.3: AUC and runtime (seconds) of ReMass-iForest (RM), iForest (IF), DEMass-LOF (DM), and LOF in benchmark datasets.

Dataset	N	M	AUC				Runtime			
			RM	IF	DM	LOF	RM	IF	DM	LOF
Http	567K	3	1.00	1.00	0.99	1.00	71	99	19	19965
ForestCover	286K	10	0.96	0.88	0.87	0.94	42	56	4	2918
Mulcross	262K	4	1.00	1.00	0.99	1.00	20	23	16	2169
Smtp	95K	3	0.88	0.88	0.78	0.95	10	12	16	373
Shuttle	49K	9	1.00	1.00	0.95	0.98	4	9	7	656
Mammography	11K	6	0.86	0.86	0.86	0.68	1	1	5	127
Satellite	6K	36	0.71	0.70	0.55	0.79	1	4	0.6	24
Breastw	683	9	0.99	0.99	0.98	0.96	0.1	0.4	0.3	0.4
Arrhythmia	452	274	0.80	0.81	0.52	0.80	0.3	0.5	5	1
Ionosphere	351	32	0.89	0.85	0.85	0.90	2	3	0.5	0.3

an AUC of 0.99. One fringe instance in the cluster C_3 was ranked higher than two clustered anomalies in A_4. This is because cluster anomalies have similar mass ratios w.r.t their parents to those for the instances in sparse normal cluster C_3. Clustered anomalies were correctly ranked and an AUC of 1.0 was achieved when $MinPts$ was increased to 5. The performance of iForest did not improve when $MinPts$ was increased to any values in the range of 2, 3, 4, 5 and 10.

In the second experiment, we used the ten benchmark datasets previously employed by Liu et al. (2008). In ReMass-iForest, iForest and DEMass-LOF, the parameter t was set to 100 as default and the best value for the sub-sample size ψ was searched from 8, 16, 32, 64, 128 to 256. In ReMass-iForest, $MinPts$ was set to 5 as default. iForest uses the default settings as specified in (Liu et al., 2008), i.e, $MinPts = 1$. The level of subdivision (b) for each attribute in DEMass-LOF was searched from 1, 2, 3, 4, 5, and 6. In LOF, the best k was searched between 5 and 4000 (or to $\frac{N}{4}$ for small datasets), with steps from 5, 10, 20, 40, 60, 80, 150, 250, 300, 500, 1000, 2000, 3000 to 4000. The best results were reported. The characteristics of the datasets, AUC and runtime (seconds) of ReMass-iForest, iForest, DEMass-LOF and LOF are presented in Table 2.3.

In terms of AUC, ReMass-iForest had better or at least similar results to iForest. Based on the two-standard-error significance test, it produced better results than iForest in the ForestCover and Ionosphere datasets. Most of these datasets do not have local anomalies. Therefore, both methods had similar AUCs in eight datasets. Note that iForest did not improve AUC when $MinPts$ was set to 5. ReMass-iForest produced significantly better AUC than DEMass-LOF in relatively high dimensional datasets (Arrhythmia - 274, Satellite - 36, Ionosphere - 32, ForestCover - 10, Shuttle - 9). These results show that DEMass-LOF has problems in handling datasets with a moderate number of dimensions (9 or 10). ReMass-iForest was competitive with LOF. It was better than LOF in the Mammography dataset, worse in the Smtp and Satellite datasets, and had equal performance in the other seven datasets.

As shown in Table 2.3, the runtimes of ReMass-iForest, iForest and DEMass-LOF were of the same order of magnitude, whereas LOF was up to three orders of magnitude

slower in large datasets. Note that we cannot conduct a head-to-head comparison of the runtimes of ReMass-iForest and iForest with DEMass-LOF and LOF because they were implemented in different platforms (MATLAB versus JAVA). The results are included here to provide an idea of the order of magnitude of runtime. The difference in runtime of ReMass-iForest and iForest was due to the difference in ψ and $MinPts$. $MinPts = 5$ results in smaller sized iTrees in ReMass-iForest than those in iForest ($MinPts = 1$). Hence, ReMass-iForest runs faster than iForest, even though the same ψ is used.

2.3.2 CBMIR: ReMass-ReFeat versus ReFeat

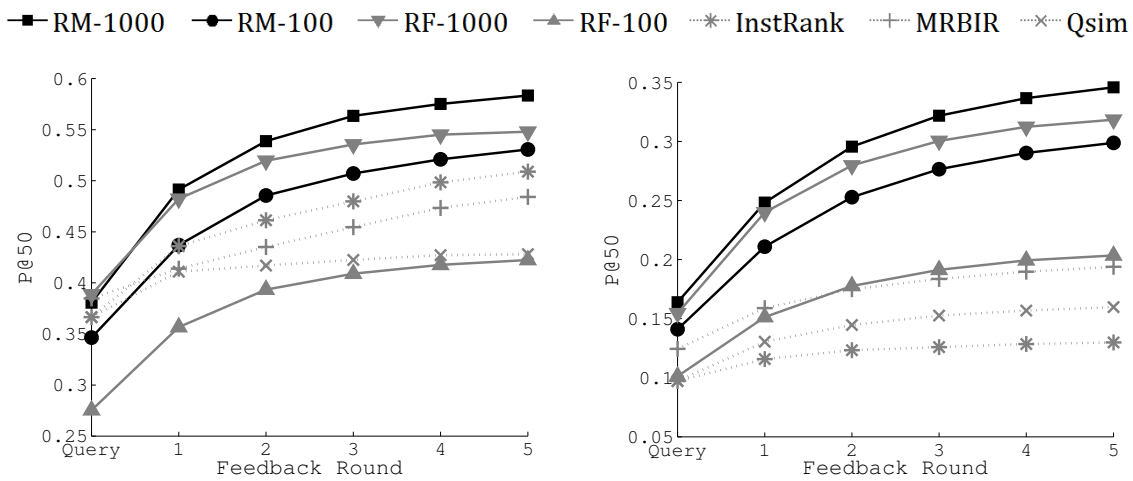
The performance of ReMass-ReFeat was evaluated against that of ReFeat in music and image retrieval tasks with the GTZAN music dataset (Tzanetakis and Cook, 2002) and the COREL image dataset (Zhou et al., 2006), respectively. GTZAN is a dataset of 1000 songs uniformly distributed in 10 genres. Each song is represented by 230 features. COREL is a dataset of 10,000 images uniformly distributed over 100 categories. Each image is represented by 67 features. These are the same datasets used in Zhou et al. (2012) to evaluate the performance of ReFeat. The results of the existing CBMIR systems InstRank, Qsim and MRBIR were taken from Zhou et al. (2012).

We conducted our experiments using the same experimental design as that in Zhou et al. (2012). Initially five queries were chosen randomly from each class. For each query, instances from the same class were regarded as relevant and the other classes were irrelevant. At each round of feedback, two relevant (instances from the same class) and two irrelevant (instances from the other classes) instances were provided. Up to five rounds of feedback were conducted for each query. The instance was not used in ranking if it was used as a feedback instance. The feedback process was repeated five times with different relevant and irrelevant feedbacks. The above process was repeated 20 times and average P@50 was reported.

In ReMass-ReFeat, the parameters ψ and $MinPts$ were set as default to 256 and 1, respectively. In ReFeat, ψ was set to 4 for GTZAN and 8 for COREL, as reported in Zhou et al. (2012). Other settings of ψ in ReFeat were found to perform worse than these settings. In order to show how their retrieval performance varies when ensemble size was increased, we used two settings for t: ReMass-ReFeat and ReFeat with (i) $t = 100$ (RM-100 and RF-100) and (ii) $t = 1000$ (RM-1000 and RF-1000). The feedback parameter γ was set as default to 0.5 in ReMass-ReFeat and 0.25 in ReFeat (as used in Zhou et al. (2012)).

P@50 of ReMass-ReFeat (RM-100 and RM-1000), ReFeat (RF-100 and RF-1000), InstRank, MRBIR and Qsim in the GTZAN and COREL datasets are shown in Figure 2.3. P@50 curves in both the datasets show that ReMass-ReFeat (RM-1000) has better retrieval performance than all contenders, especially in feedback rounds. In round 1 or no feedback (query only), ReMass-ReFeat (RM-1000) and ReFeat (RF-1000) produced similar retrieval performance but in later feedback rounds, RM-1000 produced better results than RF-1000.

It is interesting to note that the performance of RF-100 was worse than that of RM-100 in all feedback rounds including query only (no feedback). In GTZAN, RF-100 had worse



(a) GTZAN (b) COREL

Figure 2.3: Precision at top 50 returned results (P@50)

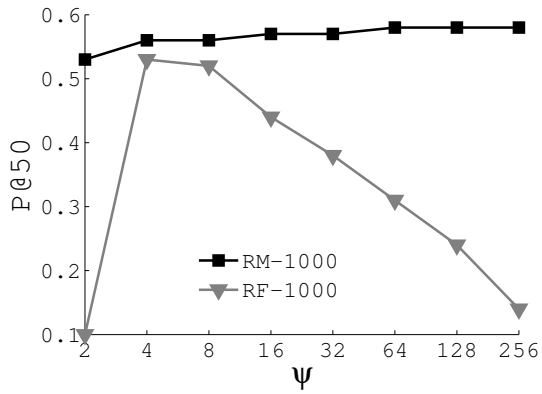


Figure 2.4: P@50 at feedback round 5 with varying sample size (ψ) in the GTZAN dataset.

performance than all other contenders. The increase in P@50 from RF-100 to RF-1000 was much larger than that of RM-100 to RM-1000. This result shows that the retrieval performance of ReFeat is mainly due to the large ensemble size of 1000. The difference in P@50 of RM-100 and RF-1000 decreased in subsequent feedback rounds. This indicates that ReMass-ReFeat produces better results than ReFeat, even with a smaller ensemble size if more feedback instances are available.

In terms of runtime, ReMass-ReFeat had slightly higher runtime than ReFeat because of the higher ψ that allows trees to grow deeper (256 vs. 4 in GTZAN and 8 in COREL). The model-building time of RM-1000 was 21 seconds (vs. 4 seconds for RF-1000) in COREL and 20 seconds (vs. 2 seconds for RF-1000) in GTZAN. The on-line retrieval time for one query of RM-1000 was 0.9 seconds (vs. 0.3 seconds for RF-1000) in COREL and 0.2 seconds (vs. 0.2 seconds for RF-1000) in GTZAN.

Figure 2.4 shows the effect of ψ on the P@50 of ReMass-ReFeat and ReFeat at feedback round 5 (one run) in the GTZAN dataset. In ReFeat, when ψ was increased above 4, the retrieval performance degraded. This is due to the increase in the height of iTrees ($h = \log_2(\psi)$) and instances falling in two distinct branches having similar relevance

scores based on the same path lengths. In contrast, ReMass-ReFeat improved its retrieval performance up to 64 and then remained almost flat beyond that. A similar effect was observed in the COREL dataset where the performance of ReFeat degraded when ψ was set above 8.

2.4 Conclusions

While the relative mass was motivated to overcome the weakness of iForest in detecting local anomalies, we have shown that the idea has a wider application. In information retrieval, we applied it to overcome the weakness of a state-of-the-art system called ReFeat. Our empirical evaluations show that ReMass-iForest and ReMass-ReFeat perform better than iForest and ReFeat, respectively, in terms of task-specific performance. In comparison with other state-of-the-art systems in both tasks, ReMass-iForest and ReMass-ReFeat are found to be either competitive or better.

The idea of relative mass in ReMass-iForest is similar to that of relative density in LOF and our empirical results show that ReMass-iForest and LOF have similar anomaly detection performance. However, ReMass-iForest runs significantly faster than LOF in large datasets because it does not require distance or density calculations.

Acknowledgements

This work is partially supported by the U.S. Air Force Research Laboratory, under agreement #FA2386-13-1-4043. Sunil Aryal is supported by an Australian Postgraduate Award (APA) at Monash University. The paper on mass-based similarity measure (Ting et al., 2013) inspired us to create the relevance score based on relative mass used in ReMass-ReFeat, although the motivations of the two papers differ.

References

Achtert, E., Hettab, A., Kriegel, H.-P., Schubert, E. and Zimek, A. (2011). Spatial outlier detection: Data, algorithms, visualizations, *In Proceedings of the 12th International Conference on Advances in Spatial and Temporal Databases*, Springer, Berlin Heidelberg, pp. 512–516.

Breunig, M. M., Kriegel, H.-P., Ng, R. T. and Sander, J. (2000). LOF: Identifying Density-Based Local Outliers, *In Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 93–104.

Giacinto, G. and Roli, F. (2005). Instance-based relevance feedback for image retrieval, *In Proceedings of the 17th International Conference on Neural Information Processing Systems*, MIT Press, pp. 489–496.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. (2009). The WEKA Data Mining Software: An Update, *SIGKDD Exploration Newsletter* **11**(1): 10–18.

He, J., Li, M., Zhang, H.-J., Tong, H. and Zhang, C. (2004). Manifold-ranking based image retrieval, *In Proceedings of the 12th annual ACM international conference on Multimedia*, ACM, New York, USA, pp. 9–16.

Liu, F., Ting, K. M. and Zhou, Z.-H. (2008). Isolation forest, *In Proceedings of the Eighth IEEE International Conference on Data Mining*, pp. 413–422.

Rui, Y., Huang, T., Ortega, M. and Mehrotra, S. (1998). Relevance feedback: a power tool for interactive content-based image retrieval, *IEEE Transactions on Circuits and Systems for Video Technology* **8**(5): 644–655.

Ting, K. M., Fernando, T. L. and Webb, G. I. (2013). Mass-based Similarity Measure: An Effective Alternative to Distance-based Similarity Measures, *Technical Report 2013/276*, Clayton School of IT, Monash University, Australia.

Ting, K. M., Washio, T., Wells, J., Liu, F. T. and Aryal, S. (2013b). DEMass: A new density estimator for big data, *Knowledge and Information Systems* **35**(3): 493–524.

Ting, K. M., Zhou, G.-T., Liu, F. and Tan, S. (2013a). Mass estimation, *Machine Learning* **90**(1): 127–160.

Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals, *IEEE Transactions on Speech and Audio Processing* **10**(5): 293–302.

Zhou, G.-T., Ting, K. M., Liu, F. T. and Yin, Y. (2012). Relevance feature mapping for content-based multimedia information retrieval, *Pattern Recognition* **45**(4): 1707–1720.

Zhou, Z.-H., Chen, K.-J. and Dai, H.-B. (2006). Enhancing relevance feedback in image retrieval using unlabelled data, *ACM Transactions on Information Systems* **24**(2): 219–244.

Zhou, Z.-H. and Dai, H.-B. (2006). Query-sensitive similarity measure for content-based image retrieval, *In Proceedings of the Sixth International Conference on Data Mining*, pp. 1211–1215.

Chapter 3

m_p-dissimilarity: A data-dependent dissimilarity measure

This chapter introduces a data-dependent dissimilarity measure called m_p-dissimilarity (where $p > 0$). It has the same formulation as the traditional ℓ_p-norm, except that the spatial distance of two instances in each dimension is replaced with the probability data mass between them. This chapter shows that by simply replacing the distance with the probability mass, m_p-dissimilarity produces better and more consistent task-specific performance than other widely-used distance measures such as ℓ_p-norm and cosine distance across a wide range of medium-to high-dimensional datasets.

The work on m_p-dissimilarity is reported in the following papers:

Aryal, S., Ting, K. M., Haffari, G. and Washio, T. (2014), m_p-dissimilarity: A data-dependent dissimilarity measure, In *Proceedings of the IEEE International conference on data mining (ICDM) 2014*, IEEE, pp. 707-712.

Aryal, S., Ting, K. M., Washio, T. and Haffari, G. (2017), Data-dependent dissimilarity measure: an effective alternative to geometric distance measures, *Knowledge and Information Systems*, Springer, London, pp. 1-28. doi: 10.1007/s10115-017-1046-0 (published online, paper format in press).

The journal paper is an extended version of the conference paper. This chapter is a copy of the paper published in the journal and a copy of the conference paper is attached in Appendix A. In order to generate a consistent presentation within the thesis, the format and some notations or symbols used have been changed, minor grammar and spelling mistakes have been corrected, and sections of the published paper have been renumbered.

The original published version of the journal paper is available at Springer via `https://doi.org/10.1007/s10115-017-1046-0`

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

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

[1]School of Engineering and Information Technology, Federation University, Australia
[2]Clayton School of Information Technology, Monash University, Australia
[3]The Institute of Scientific and Industrial Research, Osaka University, Japan

Abstract:

Nearest neighbour search is a core process in many data-mining algorithms. Finding reliable closest matches of a test instance remains 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 kNN 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 a wide range of moderate-to high-dimensional datasets with continuous only, discrete only, and mixed attributes.

Keywords: Distance measure, ℓ_p-norm, cosine distance, m_p-dissimilarity

3.1 Introduction

In order to make a prediction for a test instance, many data-mining algorithms search for its k closest matches or nearest neighbours (kNNs) in the given training set, and make a prediction based on the kNNs. They use a (dis)similarity or distance measure to find kNNs. However, finding reliable kNNs 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 (Beyer et al., 1999; Aggarwal et al., 2001; François et al., 2007).

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 $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ (where $\mathbb{R}$ is a real domain) be a measure of dissimilarity between two vectors in $\mathcal{X}$. The most common approach to 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 (also known as ℓ_p-norm) (Deza and Deza, 2009) is a widely-used distance measure. It estimates the dissimilarity between two M-dimensional vectors $\mathbf{x}$ 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 has been shown to produce more reliable kNNs than ℓ_2-norm (Salton and McGill, 1986). Cosine distance is proportional to the Euclidean distance of the length normalised 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.

Psychologists have expressed concerns about the geometric model of dissimilarity measure (Tversky, 1977; Krumhansl, 1978), arguing that the judged dissimilarity between two objects is influenced by the context of dissimilarity measurement and other objects in proximity. Krumhansl (1978) has suggested a distance-density model of dissimilarity measure, arguing that two objects in a relatively dense region are 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 fewer 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, consider an example of a dataset with distributions in dimensions i and j as shown in Table 3.1. 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 in

Table 3.1: An example of data distribution in two dimensions

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

dimension j is very common (9 out of 10 instances have the same value). In geometric distance measures such as ℓ_p-norm, 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 psychologists is that the same value in dimension j (where probability of the value is very high) does not provide the same amount of information about the (dis)similarity between $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ as the same value in dimension i (where the probability of the value is very small). This scenario, where many instances have the same value in many dimensions, can be very common in high-dimensional spaces, as data often lie 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 few 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 the 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 between and around them in many dimensions. In 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 suggested by psychologists.

Our empirical evaluation in kNN 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 datasets with continuous only, discrete only and mixed attributes.

The rest of the paper is organised as follows. Previous work related to this paper is discussed in Section 3.2. The proposed m_p-dissimilarity is presented in Section 3.3,

followed by empirical results in Section 3.4. The relationship of m_p-dissimilarity with ℓ_p-norm after rank transformation of data is discussed in Section 3.5, followed by the related discussion in Section 3.6. Finally, we conclude the paper with conclusions and future work in the last section. Hereafter, we refer to m_p-dissimilarity and ℓ_p-norm as m_p and ℓ_p, respectively.

3.2 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.

3.2.1 Dissimilarity measures in continuous domain

In the continuous domain where each dimension is numeric, i.e., $\forall_i\ x_i \in \mathbb{R}$, 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:

$$d_{mink,p}(\mathbf{x},\mathbf{y}) = \ell_p(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_p = \left(\sum_{i=1}^{M} abs(x_i - y_i)^p\right)^{\frac{1}{p}} \tag{3.1}$$

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 normalisation ($x'_i = \frac{x_i - min_i}{max_i - min_i}$, 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]). Although min-max normalisation takes account of scale differences between different dimensions, it does not take account 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 equal variance in each dimension, standard deviation normalisation ($x''_i = \frac{x_i}{\sigma_i}$ 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 normalised vectors standardised ℓ_p (s-ℓ_p) i.e., s-$\ell_p(\mathbf{x},\mathbf{y}) = \ell_p(\mathbf{x}'',\mathbf{y}'')$. Standardised ℓ_p with $p = 2$ (s-ℓ_2) is the simplest variant of Mahalanobis distance (Deza and Deza, 2009), where the covariance matrix is a diagonal matrix of variance of values in each dimension.

The Mahalanobis distance (Mahalanobis, 1936; Deza and Deza, 2009) of $\mathbf{x}$ and $\mathbf{y}$ is defined as follows:

$$d_{mah}(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T \Sigma^{-1} (\mathbf{x}-\mathbf{y})} \tag{3.2}$$

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

Rather than using the inverse of the sample covariance matrix, the distance metric learning literature focuses on learning a generalised Mahalanobis distance (Yang, 2006; Kulis, 2013; Bellet et al., 2013; Wang and Sun, 2015) from D defined as follows:

$$d_{genMah}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \Omega (\mathbf{x} - \mathbf{y})} \tag{3.3}$$

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

Since Ω is positive semi-definite, it can be factorised as $\Omega = \Lambda^T \Lambda$ where $\Lambda \in \mathbb{R}^{\omega \times M}$ and ω is a positive integer and $d_{genMah}(\mathbf{x}, \mathbf{y})$ can be written as: $d_{genMah}(\mathbf{x}, \mathbf{y}) = \|\Lambda \mathbf{x} - \Lambda \mathbf{y}\|_2$ (Kulis, 2013; Bellet et al., 2013; Wang and Sun, 2015). The generalised Mahalanobis distance is the Euclidean distance of vectors transformed by matrix Λ. The goal of distance 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 from each other (dissimilarity constraints). Learning the best Λ requires learning intensive optimisation, which is expensive in high-dimensional and/or large datasets. Furthermore, Λ is optimised specifically for the given task, and it may not be good for other tasks using the same dataset. 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 (also known as cosine distance) (Deza and Deza, 2009) is a more sensible choice to measure dissimilarity between two documents. The cosine distance between two vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as follows (Deza and Deza, 2009):

$$d_{cos}(\mathbf{x}, \mathbf{y}) = 1 - \frac{\sum_{i=1}^{M} x_i \times y_i}{\sqrt{\sum_{i=1}^{M} x_i^2} \times \sqrt{\sum_{i=1}^{M} y_i^2}} \tag{3.4}$$

Cosine distance is proportional to Euclidean distance when the vectors are length normalised to be of unit lengths, which is referred to as cosine normalisation 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 cosine distance (Salton and Buckley, 1988; Lan et al., 2009). Cosine distance with Term Frequency - Inverse Document Frequency (TF-IDF)-based term weighting (Salton and Buckley, 1988) has been shown to perform well in many text mining problems, such as text categorisation, text clustering and text retrieval tasks.

In both distance metric learning and term weighting, the focus is to transform data so that the task-specific performance of Euclidean or cosine distance is maximised in the given dataset. Some aspects of data distribution are taken into consideration in the transformation in metric learning and in term weighting, but still restricted to being

Table 3.2: $s(x_i, y_i)$ 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	$[1 + \log f(x_i) \times \log f(y_i)]^{-1}$

a metric in the transformed space, i.e., dissimilarity is still computed based solely on geometrical positions in the transformed space.

3.2.2 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 $x_i \neq y_i$ (Tanimoto, 1958; Boriah et al., 2008). Other approaches such as occurrence frequency (OF) and inverse occurrence frequency (IOF) (Boriah et al., 2008) 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 definitions of $s(x_i, y_i)$ based on overlap, OF and IOF (Boriah et al., 2008) are provided in Table 3.2.

Lin (1998) 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:

$$s_{lin,ord}(x_i, y_i) = \frac{2 \times \log \sum_{z_i=\min(x_i,y_i)}^{\max(x_i,y_i)} P(z_i)}{\log P(x_i) + \log P(y_i)} \tag{3.5}$$

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. (2008) used Lin's information theoretic definition of similarity in nominal discrete domain as follows:

$$s_{lin,nom}(x_i, y_i) = \frac{2 \times \log P(x_i \vee y_i)}{\log P(x_i) + \log P(y_i)} \tag{3.6}$$

In multivariate discrete domain, dissimilarity[1] between two instances $\mathbf{x}$ and $\mathbf{y}$ using Lin's measure can be estimated as follows (Boriah et al., 2008):

$$d_{lin}(\mathbf{x}, \mathbf{y}) = 1 - \frac{1}{M} \sum_{i=1}^{M} s_{lin}(x_i, y_i) \tag{3.7}$$

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

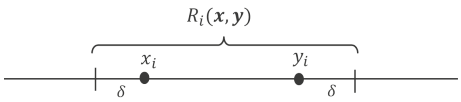


Figure 3.1: $R_i(\mathbf{x}, \mathbf{y})$

Boriah et al. (2008) have shown that d_{lin} performs better than d_{of} and d_{iof} in discrete domains. Although 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.

3.2.3 Dissimilarity measures in mixed domain

Many real-world applications have both continuous and discrete attributes, resulting in a 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.

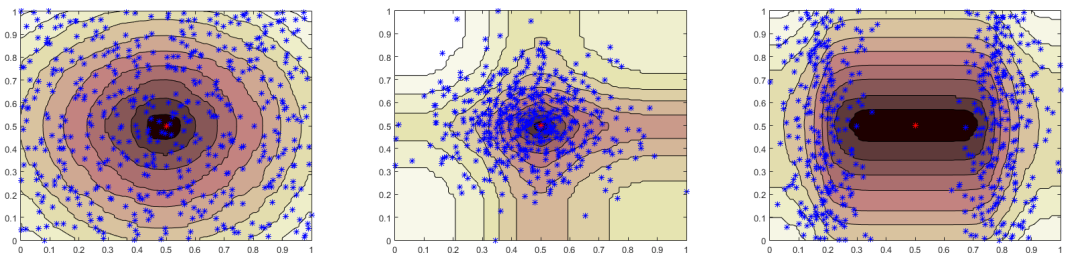
Other approaches include converting attributes into continuous only or discrete only and using (dis)similarity measures designed for continuous or discrete domains. A continuous attribute can be converted into a discrete attribute through discretisation (Hall et al., 2009). A discrete attribute with u discrete labels can be converted into u continuous attributes by converting each discrete label into a binary attribute, where 0 represents the absence of the label and 1 represents the presence. All converted u binary attributes are treated as continuous attributes (Hall et al., 2009).

3.3 Data-dependent dissimilarity measure

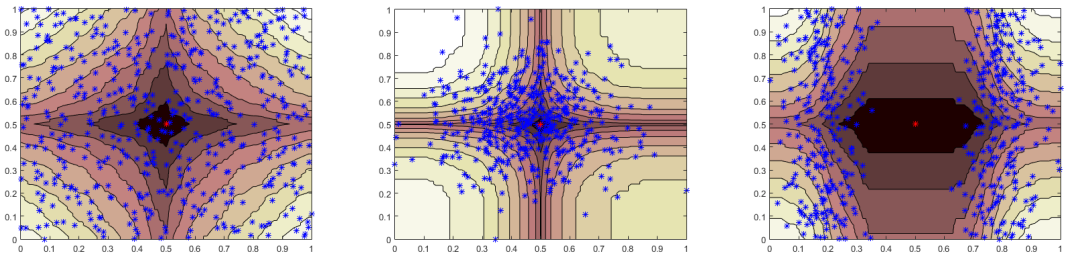
In order to measure dissimilarity between $\mathbf{x}$ and $\mathbf{y}$, instead of using $abs(x_i - y_i)$ in Eqn 3.1, we propose to consider the relative positions of $\mathbf{x}$ and $\mathbf{y}$ with respect to the rest of the data distribution in each dimension. The dissimilarity between $\mathbf{x}$ 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 can be defined as:

$$m_p(\mathbf{x}, \mathbf{y}) = \left(\frac{1}{M} \sum_{i=1}^{M} \left(\frac{|R_i(\mathbf{x}, \mathbf{y})|}{N} \right)^p \right)^{\frac{1}{p}} \tag{3.8}$$

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 Figure 3.1.



(a) $p = 2.0$



(b) $p = 0.5$

Figure 3.2: Contour plots of dissimilarity of points in the space with reference to the centre ($\langle 0.5, 0.5 \rangle$), based on m_p (with δ for each dimension i set to $\frac{\sigma_i}{2}$) in three data distributions (uniform: left column, normal: middle column, and bimodal: right column). The darker the colour, the smaller the dissimilarity.

The region is extended by small $\delta > 0$ beyond x_i and y_i to consider the density distribution around them and the distribution 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 $m_p(\mathbf{x}, \mathbf{y})$ 'm_p-`dissimilarity`'. This measure captures the essence of the distance-density model proposed by psychologists (Krumhansl, 1978), which suggests 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 3.A.

The dissimilarity between a pair of instances using Eqn 3.8 depends on the distribution of data. Figure 3.2 shows 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 with 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 is depicted in the two contour plots in the first column in Figure 3.2 which exhibit similar contour plots to those of ℓ_2 and $\ell_{0.5}$.

3.3.1 Time complexity and efficient approximation

In continuous domains, estimating $m_p(\mathbf{x}, \mathbf{y})$ using Eqn 3.8 is expensive, especially when either $\mathbf{x}$ or $\mathbf{y}$ is an unseen instance, as it requires a range search in each dimension to

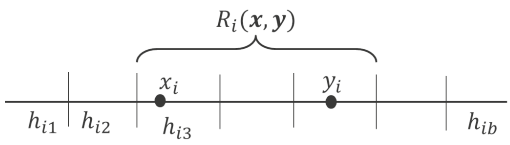


Figure 3.3: Defining $R_i(\mathbf{x}, \mathbf{y})$ using bins

estimate $|R_i(\mathbf{x}, \mathbf{y})|$. One-dimensional range search can be done in $O(\log N)$ using a binary search tree 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 datasets.

Alternatively, $|R_i(\mathbf{x}, \mathbf{y})|$ can be approximated efficiently by using a histogram, i.e., dividing 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 pre-processing 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 between $\mathbf{x}$ and $\mathbf{y}$, as shown in Figure 3.3. Although the approximation using bins does not extend the range exactly by δ beyond x_i and y_i, the bins in which 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:

$$|R_i(\mathbf{x}, \mathbf{y})| = \sum_{q=l}^{o} |h_{iq}| \tag{3.9}$$

Note that the binning can be done in two ways: (i) equal-width: each bin is of the same size (bins in the dense region have more data mass than those in the sparse region); (ii) equal-frequency: each bin has approximately the same number of instances (bins are smaller in the dense region than in the sparse region). The former is sensitive to outliers. If there is only one instance with significantly different value than others, it may affect the discrimination between the other instances, as they may all 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] (Hall et al., 2009). 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 equal data mass.

The pre-processing 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 pre-computed for each pair of bins in each dimension in $O(Mb^2)$ time and stored in $O(Mb^2)$ space. Following pre-processing, 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}.

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

3.3.2 Handling discrete attributes

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

$$|R_i(\mathbf{x}, \mathbf{y})| = \sum_{z_i=\min(x_i,y_i)}^{\max(x_i,y_i)} f(z_i) \tag{3.10}$$

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

Unlike d_{lin} which 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:

$$|R_i(\mathbf{x}, \mathbf{y})| = \begin{cases} f(x_i) & \text{if } x_i = y_i \\ N & \text{otherwise} \end{cases} \tag{3.11}$$

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} (Boriah et al., 2008). 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 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 pre-processing step which requires $O(NM)$ time and $O(Mu)$ (where u is the average number of discrete labels per dimension) space.

3.3.3 Dissimilarity measure in bag-of-words vector representation

In the case of bag-of-words (BoW) (Salton and McGill, 1986) vector representations, each component of a vector represents the 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 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:

$$m_p(\mathbf{x}, \mathbf{y}) = \left(\frac{1}{|F_{\mathbf{x},\mathbf{y}}|} \sum_{i \in F_{\mathbf{x},\mathbf{y}}} \left(\frac{|R_i(\mathbf{x}, \mathbf{y})|}{N} \right)^p \right)^{\frac{1}{p}} \tag{3.12}$$

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 normalisation term employed to account for different numbers of features used for measuring the 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 dataset will have 0 values. This is not an issue for ℓ_p because it assigns 0 dissimilarity when $x_i = y_i = 0$.

3.3.4 Distinguishing properties of m_p

Data-dependent self dissimilarity. The distinguishing characteristic of m_p compared with the geometry-based (ℓ_p and d_{cos}) and probabilistic (d_{lin}) dissimilarity measures discussed in Section 3.2 is self-dissimilarity. The self-dissimilarity of m_p is not zero; 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 non-zero constant self-dissimilarity for m_p only if each bin has the same number of instances in each dimension. This is often not possible, because there are duplicate values in many instances, and this occurs in many dimensions. This is a common characteristic of many high-dimensional datasets 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 the data-dependent self-dissimilarity of m_p over data-independent self-dissimilarity of existing measures is discussed in Section 3.5.

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 in the case of matching labels: data-dependent self-dissimilarity in action.

m_p is equivalent to ℓ_p only under uniform distribution. Under uniform distribution and with 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.

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 the scales and units of measurement in continuous domains. It does not require preprocessing of data to address the scaling issue (min-max normalisation) or differences in variance across different dimensions (standard deviation normalisation). 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. 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.

3.4 Empirical evaluation

This section presents the results of experiments conducted to demonstrate that by simply 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 datasets. We evaluated the performance of m_p against the general-purpose dissimilarity measures of Minkowski distance (ℓ_p), Minkowski distance after standard deviation normalisation (s-ℓ_p), cosine distance (d_{cos}) and Lin's probabilistic measure (d_{lin}) in k-nearest neighbour (kNN) 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: $d_{cos}, d_{lin}, \ell_{0.5}, \ell_2, s\text{-}\ell_{0.5}, s\text{-}\ell_2, m_{0.5}$ and m_2. All dissimilarity measures and algorithms were implemented in Java using the WEKA platform (Hall et al., 2009).

We used moderately high to high-dimensional ($M \geq 20$) datasets from different application areas such as text, image, music, character and digit recognition, medical and biology, games, etc. In text collections, documents were represented by TF-IDF (Salton and Buckley, 1988) weighted 'bag-of-words' (Salton and McGill, 1986) vectors. Feature values in each dimension in all other non-text datasets were normalised to be in the unit range. For d_{lin}, continuous attributes were converted into ordinal attributes using discretisation, as in the case of m_p.

The properties of the datasets are provided in Table 3.3. NG20, R52, R8, Webkb were from Cardoso-Cachopo (2007)[3]; Ohscal, Wap, New3s and Fbis were from Han and Karypis (2000)[4]; Caltech256 (sift bag-of-words features) from Tuytelaars et al. (2010)[5]; Corel and Gtzan were from Zhou et al. (2012); HBA was from Ariyaratne and Zhang (2012) and the rest of the other datasets were from UCI (Bache and Lichman, 2013)[6] and WEKA (Hall et al., 2009)[7].

We discuss the experimental set-ups and results in kNN classification and content-based multimedia information retrieval (CBMIR) tasks in the next two subsections.

3.4.1 kNN classification

In the kNN classification context, in order to predict a class label for a test instance $\mathbf{x}$, its k nearest neighbours were searched in the training set using all eight dissimilarity measures and the most frequent label in kNNs was predicted as the class label for the test instance. All classification experiments were conducted using 10-fold cross-validation: 10 train-and-test trials using 90% of the given dataset for training and 10% for testing. We set k to the commonly-used value of 5. The average classification accuracy (%) over a 10-fold 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 10-fold cross-validation) did not overlap. The average classification accuracies over

[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 3.3: 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 brackets along with the total number of dimensions (M) and c is the number of classes in a dataset

Name	N	M (M_{nom})	c	Application area
New3s	9558	26832 (0)	44	Text (TREC Collection)
Ohscal	11162	11465 (0)	10	Text (Ohsumed patients' information)
Arcene	200	10000 (0)	2	Bioinformatics (Cancer)
Wap	1560	8460 (0)	20	Text (Yahoo web pages)
R52	9100	7369 (0)	52	Text (Reuters Collection)
NG20	18821	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	30607	1000 (0)	257	Image
Mnist	70000	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	12495	191 (0)	6	Music
Hba	1500	187 (0)	15	Music
Musk2	6598	166 (0)	2	Chemoinformatics
Corel	10000	67 (0)	100	Image
Splice	3190	60 (60)	3	Bioinformatics (DNA)
Miniboone	129596	50 (0)	2	Physics (particles)
Connect-4	67557	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 10-fold cross-validation of the eight dissimilarity measures in all datasets are provided in Table 3.4.

Out of 30 datasets, $m_{0.5}$ and m_2 produced the best result or equivalent to the best result in 23 and 16 datasets, 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 datasets. The results summarised in the last two rows in Table 3.4 show that both $m_{0.5}$ and m_2 produced consistently top or near-top results across different datasets. $m_{0.5}$ and m_2 have average ranking of 1.97 and 2.37, respectively, whereas the average rank of the closest contender d_{cos} is 3.30. Table 3.5 provides the results summarised in terms of the win:loss:draw counts of $m_{0.5}$ and m_2 against the other six contenders using confidence intervals based on the two standard errors in the 10-fold cross-validation (standard errors are provided in Table 3.10

Table 3.4: Average accuracy of 5NN classification over a 10-fold cross-validation. The average accuracy and average rank of measures in 30 datasets 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).

in Appendix 3.C). Table 3.5 shows that both $m_{0.5}$ and m_2 had significantly more wins than losses against all other contenders.

Note that in datasets 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 3.5: Win:loss:draw counts of $m_{0.5}$ and m_2 against other measures in 5NN 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

All eight measures had runtimes of the same order of magnitude. For example, predicting class labels for instances in one fold of train-and-test in NG20 took 21458 seconds for m_2 and 26484 seconds for $m_{0.5}$, in comparison to 16296 (d_{cos}), 28168 ($\ell_{0.5}$), 24380 (ℓ_2), 29210 (s-$\ell_{0.5}$), 25944 (s-ℓ_2) and 20515 (d_{lin}) seconds. In Corel, m_2 and $m_{0.5}$ took 37 and 47 seconds, whereas d_{cos} took 22 seconds followed by 32 (ℓ_2), 45 ($\ell_{0.5}$), 47 (s-ℓ_2), 59 (s-$\ell_{0.5}$) and 90 (d_{lin}) seconds.

3.4.2 Content-based multimedia information retrieval (CBMIR)

Given a query instance $\mathbf{q}$ for a retrieval task, all the instances in a dataset were ranked in ascending order of their dissimilarity to $\mathbf{q}$, based on a dissimilarity measure, and the first k instances were presented as the relevant instances to $\mathbf{q}$. For performance evaluation, an instance was considered to be relevant to $\mathbf{q}$ if they had 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 dataset as a query and the rest of the instances were ranked. The average P@10 of N queries was reported. For the information retrieval task, we used 10 datasets 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 $d_{cos}, \ell_{0.5}, \ell_2, s$-$\ell_{0.5}, s$-$\ell_2, d_{lin}, m_{0.5}$ and m_2 are provided in Table 3.6.

Table 3.7 presents the results summarised 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 3.11 in Appendix 3.C). The table shows that both $m_{0.5}$ and m_2 produced significantly better retrieval results than the other six contenders in many datasets: $m_{0.5}$ has only 1 loss and between 7 and 10 wins; m_2 has at least 7 wins and at most 3 losses. The detailed results in Table 3.6 show that, out of the 10 datasets used, $m_{0.5}$ and m_2 produced the best result or equivalent to the best result in 9 and 6 datasets, respectively. They have average rankings of 1.20 and 2.70, respectively whereas the closest contender d_{cos} has an average ranking of 3.2.

Table 3.6: Average P@10 over N queries. The average P@10 and average rank of measures in 10 datasets 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.

Table 3.7: Win:loss:draw counts of $m_{0.5}$ and m_2 against 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-$\ell_{0.5}$	10:0:0	8:1:1
s-ℓ_2	10:0:0	9:0:1
d_{lin}	8:0:2	7:3:0

3.5 Relation to ℓ_p with rank transformation

It might appear that m_p (Eqn 3.8 with $\delta = 0$) is equivalent to ℓ_p with rank transformation (Conover and Iman, 1981) in continuous domains because they both measure dissimilarity based on the number of instances between the two instances under measurement. In rank transformation (Conover and Iman, 1981), instances in each dimension are ranked in ascending order with the smallest value having rank 1, the second smallest value having rank 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 $abs(rank(x_i) - rank(y_i)) = |\{z_i : \min(x_i, y_i) \leq z_i <$

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

$\max(x_i, y_i)\}|$. 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. Although the rank difference between x_i and y_i is data-dependent when $x_i \neq y_i$ (i.e., the rank difference between x_i and y_i is larger in a denser region than in a sparse region, even if the geometric distance is the same), but 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 transformed 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 compared the performances of m_p and ℓ_p with rank transformation (ℓ_p^{rank}) in the kNN classification task using datasets with continuous attributes only because rank transformation 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 Section 3.3.1 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 being $O(M \log N)$, which is very expensive in large datasets. We managed a 10-fold cross-validation of kNN classification completed in 24 hours in ten relatively small datasets only: Hba, Gtzan, Arcene, Mfeat, Madelon, Satellite, Fbis, Wap, Webkb and R8.

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

$$\alpha = \frac{1}{M} \sum_{i=1}^{M} \frac{w_i}{N} \tag{3.13}$$

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

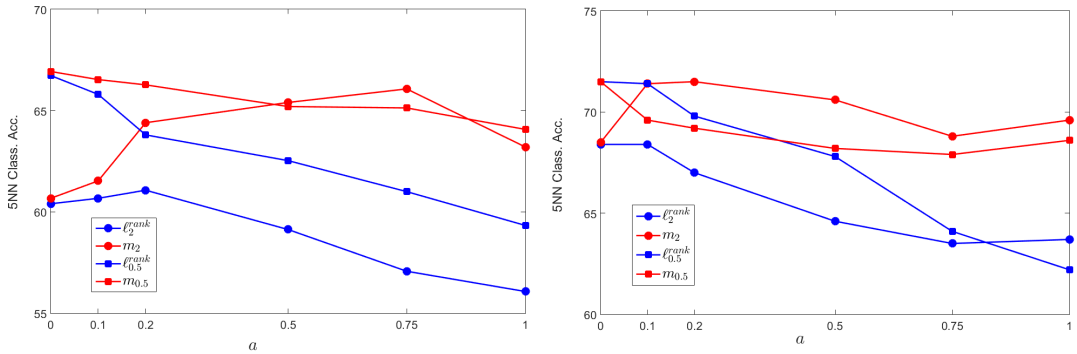
The average accuracies of 5NN classification over a 10-fold cross-validation using ℓ_2^{rank}, $\ell_{0.5}^{rank}$, m_2 and $m_{0.5}$ are provided in Table 3.8. Based on the two standard error confidence interval significance test, ℓ_2^{rank} & m_2 and $\ell_{0.5}^{rank}$ & $m_{0.5}$ produced similar 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

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

Table 3.8: The average accuracy of 5NN classification in a 10-fold 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.



(a) Hba (b) Gtzan

Figure 3.4: 5NN classification accuracies of $\ell_2^{rank}, \ell_{0.5}^{rank}, m_2$ and $m_{0.5}$ for different values of a.

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 datasets (with 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, 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 5NN classification accuracies of $\ell_2^{rank}, \ell_{0.5}^{rank}, m_2$ and $m_{0.5}$ in the Hba and Gtzan datasets with $a = 0, 0.1, 0.2, 0.5, 0.75$ and 1.0 are shown in Figure 3.4 and the corresponding α values are provided in Table 3.9.

Table 3.9: 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

Figure 3.4 shows that there is a significant difference between the classification accuracies of m_2 and $m_{0.5}$ compared to those of ℓ_2^{rank} and $\ell_{0.5}^{rank}$ for $a \geq 0.75$ in both the Hba and Gtzan datasets. This indicates that m_p can provide more reliable nearest neighbours 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 cannot only be applied to both continuous and discrete domains, but has a seamless treatment of mixed attribute-type domains.

3.6 Discussion

In a high-dimensional space, the most widely-used Euclidean distance (ℓ_2-norm) becomes ineffective. Many researchers have argued that this 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 (Beyer et al., 1999; Aggarwal et al., 2001; François et al., 2007). Let $dmax(\mathbf{x}, d)$ and $dmin(\mathbf{x}, d)$ be the dissimilarity of $\mathbf{x}$ to its farthest and nearest neighbours 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 (Beyer et al., 1999) 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 neighbours than ℓ_p and d_{cos} in many datasets, particularly in high-dimensional problems (see the experimental results in Sections 3.4.1 and 3.4.2). This indicates that the negative impact of the concentration phenomenon may not be as severe in practice as it is believed to be theoretically. This finding is consistent with that suggested by François et al. (2007). The detailed empirical result of the phenomenon of concentration of m_2, ℓ_2 and d_{cos} is provided in Appendix 3.B.1.

Another issue of distance measures in high-dimensional spaces discussed in the literature is 'hubness' (Radovanović et al., 2010). Let $N_k(\mathbf{y})$ be the set of k nearest neighbours of $\mathbf{y}$, and k-occurrences of $\mathbf{x}$, $O_k(\mathbf{x}) = |\{\mathbf{y} : \mathbf{x} \in N_k(\mathbf{y})\}|$, i.e., the number of other instances in the given dataset where $\mathbf{x}$ is one of their k nearest neighbours. 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 (Radovanović et al., 2010). The instances with large $O_k(\cdot)$ are considered as 'hubs', i.e., the popular nearest neighbours. Hubness becomes prominent

in high-dimensional space, and it affects the performance of kNN-based algorithms. For example, if $\mathbf{x}$ is a hub, it appears in the kNN sets of many test instances and contributes to the prediction decisions, but it may not be relevant to make predictions for all 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 Appendix 3.B.2.

In order to circumvent the high-dimensionality issue, dimensionality reduction (Fodor, 2002) techniques are used before using distance measures. In continuous domain, principal component analysis (PCA) (Jolliffe, 2005) 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 the 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.

The main purpose of PCA is dimensionality reduction, which enables the application of distance measures to high-dimensional datasets. It usually does not improve predictive accuracy. This is exactly what we observed in the 5NN classification task. For example, the 5NN classification accuracies of d_{cos} and ℓ_2 were increased in Corel and Hba but that of $\ell_{0.5}$ was decreased in both datasets. 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 more consistent results across different datasets. The detailed results of this comparison are provided in Table 3.12 in Appendix 3.D.

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

Various data-dependent distance metric adaptation techniques to improve the task-specific performance of distance measures in a given dataset have been proposed. Weighted Minkowski distance (Deza and Deza, 2009) assigns weight to the distance in each dimension, based on the observed data. Standardised 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 optimisation. In transductive learning, Lundell and Ventura (2007) 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.

Distance metric learning (Yang, 2006; Wang and Sun, 2015) 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 a low-dimensional space by ignoring dimensions with smaller eigen values, regularised matrix relevance learning (Schneider et al., 2010) 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 datasets. They optimise distance metric specifically for the given task which may not be good for other tasks using the same dataset. They are not general-purpose measures like m_p, ℓ_p or d_{cos}.

All the adaptive distance 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. All these techniques still rely on geometric models and metric assumptions. Although metric-based measures have nice mathematical properties, their assumptions might be inappropriate to model some problems. Recently, Schleif and Tino (2015) discussed issues of metric based proximity learning and provided a comprehensive review of non-metric proximity learning.

In this paper, we focus only 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 future, it would be interesting to investigate how learning can be applied to data-dependent dissimilarity measures 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, it appears to have some similarity with Locality Sensitive Hashing (LSH) (Indyk and Motwani, 1998). The aims of binning are different in the two cases. In LSH, bins are used to quickly find a small set of candidate nearest neighbours of a test instance, of which the kNNs are searched using the Euclidean distance. In contrast, in m_p, probability data mass in bins is used as a direct measure of dissimilarity. It is an open question whether LSH can be used to generate candidate sets quickly for m_p. LSH has nice theoretical bounds for Euclidean distance but it is not clear if similar bounds can be derived for m_p.

3.7 Conclusions and future work

In this paper, we propose 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 kNN 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 datasets. By simply replacing the geometric distance in each dimension

with the probability mass, kNN using m_p significantly improves the performance of kNN using ℓ_p in many high-dimensional datasets.

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

Although ℓ_p can be made data-dependent through 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 are 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 datasets where the data lie in a low-dimensional subspace. In such high-dimensional datasets, m_p produces better task-specific performance than ℓ_p with the rank transformation.

Future work includes investigating learning for m_p and comparing 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 neighbour search in the case of large N.

Acknowledgements

A preliminary version of this paper was published in the Proceedings of the IEEE International Conference on Data Mining (ICDM) 2014 (Aryal et al., 2014b). 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 AFOSR AOARD award number 15IOA008-154005 and JSPS KAKENHI grant number 2524003.

Appendix 3.A: Probabilistic interpretation of m_p

The formulation of $m_p(\mathbf{x}, \mathbf{y})$ (Eqn 3.8) 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 $R(\mathbf{x}, \mathbf{y})$ have a 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:

$$P(\mathbf{t} \in R(\mathbf{x}, \mathbf{y})|\phi(\mathbf{x})) \approx \prod_{i=1}^{M} P(t_i \in R_i(\mathbf{x}, \mathbf{y})|\phi_i(\mathbf{x})) \tag{3.14}$$

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 Eqn 3.14 is sensitive to outliers. An approximation which is tolerant to outliers can be estimated by replacing the product with the summation (Minka, 2003). The sum-based approximation relates to the probability of $\mathbf{t}$ in Eqn 3.14 under the following *outlier model*. Consider a data generation process in which, to sample t_i, a coin with the probability of turning heads $(1-\epsilon)$ is flipped. If the coin turns heads, 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 3.1. *(Minka, 2003) 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

$$P'(t_i|\phi(\mathbf{x}), \epsilon) = \epsilon/A + (1-\epsilon)P(t_i|\phi_i(\mathbf{x})) \tag{3.15}$$

We assume that each dimension is generated independently, hence

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

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 (2003), 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 the coin turns up heads 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 normalisation constant to ensure the total probability sums up to 1 in the ith dimension.

Lemma 3.2. *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 3.1 by drawing t_i from $\phi_i(\mathbf{x})^p$ $p > 1$ when the coin turns up heads in the data generation process. □

As a result of Lemma 3.2 (by considering the outlier tolerant model), $P(\mathbf{t} \in R(\mathbf{x}, \mathbf{y}))$ can be approximated as:

$$P(\mathbf{t} \in R(\mathbf{x}, \mathbf{y})) \approx C_1 + C_2 \times \sum_{i=1}^{M} \frac{P_i(t_i \in R_i(\mathbf{x}, \mathbf{y}))^p}{Z_{i,p}} \tag{3.16}$$

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 Eqn 3.16 are independent of $\mathbf{x}$ and $\mathbf{y}$ and they are simply the scaling factors of the dissimilarity measure. In order to find the nearest neighbour 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 Eqn 3.16, $m_p(\mathbf{x}, \mathbf{y})$ can be expressed as its rescaled version as follows:

$$m_p(\mathbf{x}, \mathbf{y}) = \left(\frac{1}{M} \sum_{i=1}^{M} P_i \left(t_i \in R_i(\mathbf{x}, \mathbf{y})\right)^p \right)^{\frac{1}{p}} \tag{3.17}$$

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

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

$$\hat{P}_i \left(t_i \in R_i(\mathbf{x}, \mathbf{y})\right) = \frac{|R_i(\mathbf{x}, \mathbf{y})|}{N} \tag{3.18}$$

Hence, Eqn 3.17 and Eqn 3.18 lead to m_p defined in Eqn 3.8.

Appendix 3.B: Analysis of concentration and hubness

We examined 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 datasets 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 normalised to be in unit range in each dimension.

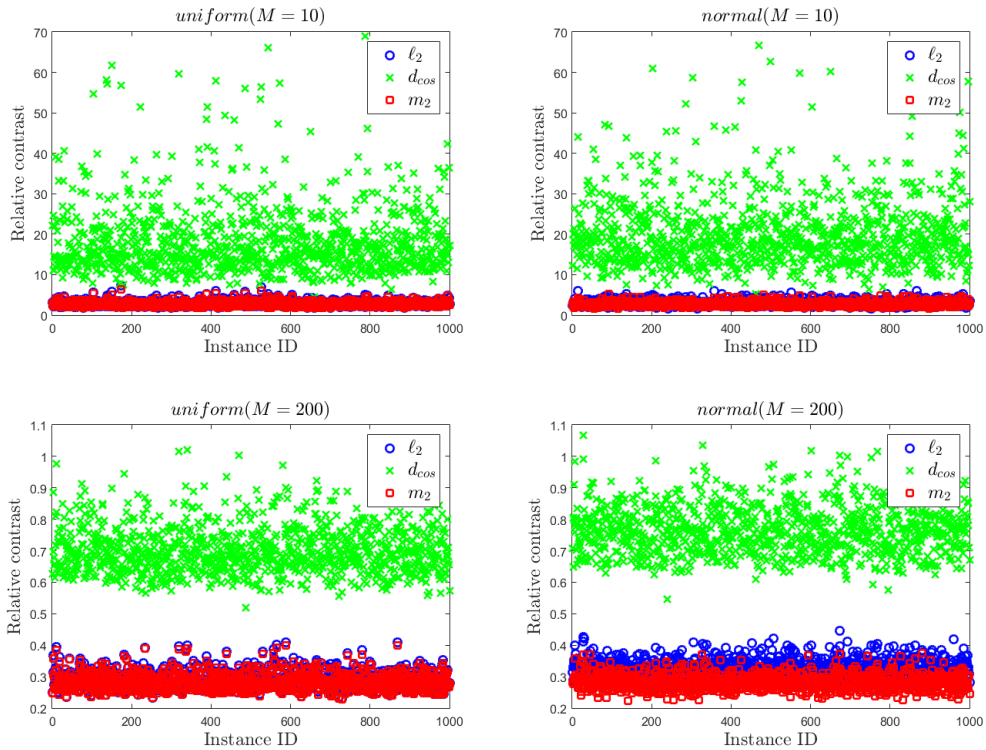


Figure 3.5: Relative contrast $\left(\frac{dmax(\mathbf{x},d)-dmin(\mathbf{x},d)}{dmin(\mathbf{x},d)}\right)$ of m_2, ℓ_2 and d_{cos}. Note that x-axis is instance id and corresponding y-axis value is the relative contrast of that instance.

Appendix 3.B.1: Concentration

The relative contrast between the nearest and farthest neighbour was computed for all $N = 1000$ instances in each dataset 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 Figure 3.5.

The relative contrast of all three measures decreased substantially (note that the y-axes have different scales in Figure 3.5) when the number of dimensions were increased from $M = 10$ 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 datasets, we observed that m_2 is more concentrated than ℓ_2 and d_{cos}.

Appendix 3.B.2: Hubness

In order to examine the hubness phenomenon, 5-Occurrences of each instance $\mathbf{x} \in D$ were 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 5NN of $\mathbf{y}$. Then, the O_5 distribution is plotted for each measure (m_2, ℓ_2 and d_{cos}) in all four synthetic datasets, and the results are shown in Figure 3.6.

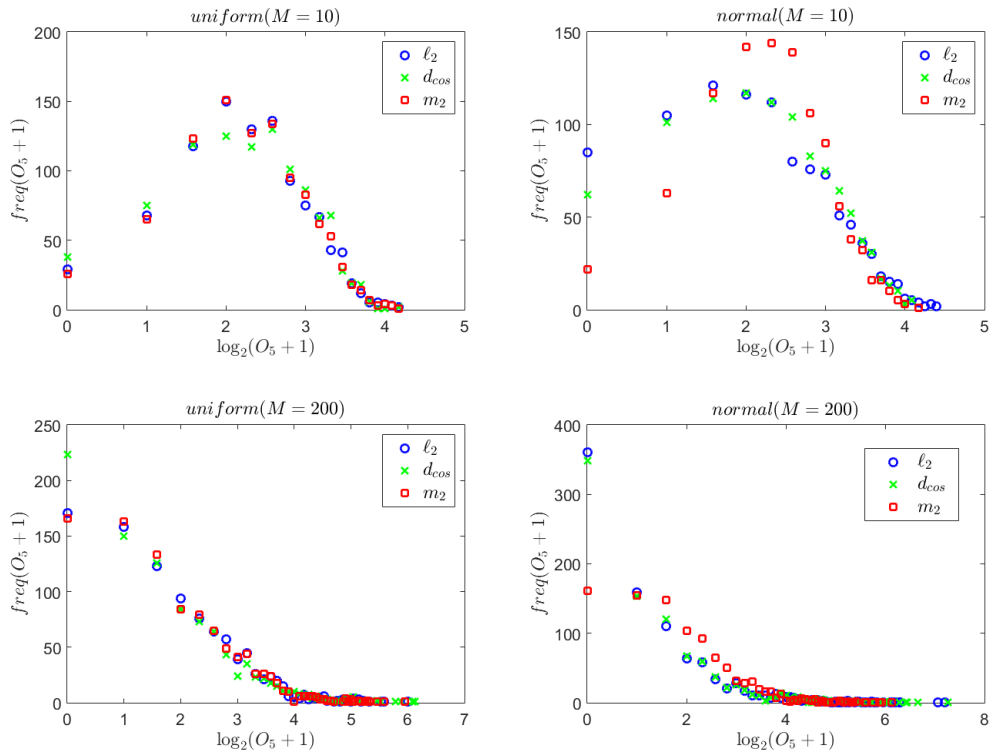


Figure 3.6: The O_5 distributions of m_2, ℓ_2 and d_{cos} in synthetic datasets. Note that x-axis is in the log scale hence x-axis value is $\log(O_5 + 1)$ to consider the case of $O_5 = 0$.

The O_5 distributions of all three measures become skewed when the number of dimensions were 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 m_2 is proportional to ℓ_2 under uniform distribution (also reflected in Figure 3.2(a)). 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 5NN 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 neighbours using ℓ_2 and d_{cos} have $O_5 = 146$ and 152, respectively, whereas the most popular nearest neighbour using m_2 has $O_5 = 69$.

We also observed similar behaviour in many real datasets where the O_5 distribution of m_2 is less skewed than that of ℓ_2 and d_{cos}.

Appendix 3.C: Standard error

Table 3.10 shows the standard error of classification accuracies (in %) of 5NN classification over a 10-fold cross-validation (average classification accuracy is presented in Table 3.4 in Section 3.4.1).

Table 3.10: Standard error of accuracies of 5NN classification over a 10-fold cross-validation. Average classification accuracy is presented in Table 3.4 in Section 3.4.1

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 3.11 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 3.6 in Section 3.4.2).

Appendix 3.D: Comparison with geometric distance measures after dimensionality reduction

Average 5NN classification accuracies over a 10-fold 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 the original space in 16 out of 22 datasets with continuous only attributes are provided in Table 3.12. 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

Table 3.11: Standard error of P@10 over N queries. Average P@10 is presented in Table 3.6 in Section 3.4.2

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

of the correlation matrix of the training data to ensure that the projection was robust to scale differences in the original dimensions. Note that PCA did not complete in 24 hours in the remaining six datasets with $M > 5000$: New3s (26832), Ohscal (11465), Arcene (10000), Wap (8460), R52 (7369) and NG20 (5489).

References

Aggarwal, C. C., Hinneburg, A. and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space, *In Proceedings of the International Conference on Database Theory*, Springer, Berlin Heidelberg, pp. 420–434.

Ariyaratne, H. B. and 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.

Aryal, S., Ting, K. M., Haffari, G. and Washio, T. (2014b). Mp-dissimilarity: A data dependent dissimilarity measure, *In Proceedings of the IEEE International Conference on Data Mining*, IEEE, pp. 707–712.

Bache, K. and Lichman, M. (2013). UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences.
URL: *http://archive.ics.uci.edu/ml*

Bellet, A., Habrard, A. and Sebban, M. (2013). A survey on metric learning for feature vectors and structured data, *CoRR* **abs/1306.6709**.
URL: *http://arxiv.org/abs/1306.6709*

Beyer, K. S., Goldstein, J., Ramakrishnan, R. and Shaft, U. (1999). When is "nearest neighbour" meaningful?, *In Proceedings of the 7th International Conference on Database Theory*, Springer-Verlag, London, UK, pp. 217–235.

Table 3.12: Average accuracy of 5NN classification over a 10-fold 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

Boriah, S., Chandola, V. and 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.

Cardoso-Cachopo, A. (2007). *Improving Methods for Single-label Text Categorization*, PhD thesis, Instituto Superior Tecnico, Technical University of Lisbon, Lisbon, Portugal.

Conover, W. J. and Iman, R. L. (1981). Rank transformations as a bridge between parametric and nonparametric statistics, *The American Statistician* **35**(3): 124–129.

Deza, M. M. and Deza, E. (2009). *Encyclopedia of Distances*, Springer, Berlin Heidelberg.

Fodor, I. (2002). A survey of dimension reduction techniques, *Technical Report UCRL-ID-148494*, Lawrence Livermore National Laboratory, University of California, USA.

François, D., Wertz, V. and Verleysen, M. (2007). The concentration of fractional distances, *IEEE Transactions on Knowledge and Data Engineering* **19**(7): 873–886.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. (2009). The WEKA Data Mining Software: An Update, *SIGKDD Exploration Newsletter* **11**(1): 10–18.

Han, E.-H. and 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-Verlag, London, UK, pp. 424–431.

Indyk, P. and Motwani, R. (1998). Approximate nearest neighbours: Towards removing the curse of dimensionality, *In Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, ACM, New York, USA, pp. 604–613.

Jolliffe, I. (2005). *Principal component analysis*, Wiley Online Library.

Krumhansl, C. L. (1978). Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density, *Psychological Review* **85**(5): 445–463.

Kulis, B. (2013). Metric learning: A survey, *Foundations and Trends in Machine Learning* **5**(4): 287–364.

Lan, M., Tan, C. L., Su, J. and Lu, Y. (2009). Supervised and traditional term weighting methods for automatic text categorization, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31**(4): 721–735.

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, USA, pp. 296–304.

Lundell, J. and 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.

Mahalanobis, P. C. (1936). On the generalized distance in statistics, *In Proceedings of the National Institute of Sciences of India*, Vol. 2, pp. 49–55.

Minka, T. P. (2003). The 'summation hack' as an outlier model. Microsoft Research. **URL:** *http://research.microsoft.com/en-us/um/people/minka/papers/minka-summation.pdf*

Radovanović, M., Nanopoulos, A. and Ivanović, M. (2010). Hubs in space: Popular nearest neighbours in high-dimensional data, *Journal of Machine Learning Research* **11**: 2487–2531.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval, *Information Processing and Management* **24**(5): 513–523.

Salton, G. and McGill, M. J. (1986). *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, USA.

Schleif, F.-M. and Tino, P. (2015). Indefinite proximity learning: A review, *Neural Computation* **27**(10): 2039–2096.

Schneider, P., Bunte, K., Stiekema, H., Hammer, B., Villmann, T. and Biehl, M. (2010). Regularization in matrix relevance learning, *IEEE Transactions on Neural Networks* **21**(5): 831–840.

Tanimoto, T. T. (1958). An elementary mathematical theory of classification and prediction, *Technical report*, International Business Machines Corporation, USA.

Tuytelaars, T., Lampert, C., Blaschko, M. B. and Buntine, W. (2010). Unsupervised object discovery: A comparison, *International Journal of Computer Vision* **88**(2): 284–302.

Tversky, A. (1977). Features of Similarity, *Psychological Review* **84**(2): 327–352.

Wang, F. and Sun, J. (2015). Survey on distance metric learning and dimensionality reduction in data mining, *Data Mining and Knowledge Discovery* **29**(2): 534–564.

Yang, L. (2006). Distance metric learning: A comprehensive survey, *Technical report*, Michigan State University, USA.

Zhou, G.-T., Ting, K. M., Liu, F. T. and Yin, Y. (2012). Relevance feature mapping for content-based multimedia information retrieval, *Pattern Recognition* **45**(4): 1707–1720.

Chapter 4

Generalised m_p-dissimilarity and its relation to other data-dependent measures

This chapter generalises m_p-dissimilarity where p is allowed to be 0 by introducing m_0-dissimilarity. It investigates the relationships and characteristics of different data-dependent measures. It shows that m_p is a generalised data-dependent dissimilarity measure, of which the existing data-dependent measures of rank difference and Lin's probabilistic measure are special cases with $p > 0$ and $p = 0$, respectively. It also analyses the behaviour of different data-dependent measures with the change in units and scales of measurements of feature values. Empirical evaluation reveals that the fully data-dependent measure of m_p-dissimilarity, which is robust to units and scales of measurement, is more effective than other data-dependent and data-independent similarity measures.

The work on generalised m_p-dissimilarity and a comparative study of its relationship, characteristics and relative performance with existing data-dependent measures has been reported in the following paper:

Aryal, S., Ting, K. M., Washio, T. and Haffari, G. (2017), A comparative study of data-dependent approaches in measuring similarities of data objects, *Knowledge Discovery and Data Mining* (under review).

This chapter is a copy of the paper submitted to the Knowledge Discovery and Data Mining journal. In order to generate a consistent presentation within the thesis, the format and some notations or symbols used have been changed, minor grammar and spelling mistakes have been corrected, and sections of the submitted paper have been renumbered.

A comparative study of data-dependent approaches in measuring similarities of data objects

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

[1]School of Engineering and Information Technology, Federation University, Australia
[2]Clayton School of Information Technology, Monash University, Australia
[3]The Institute of Scientific and Industrial Research, Osaka University, Japan

Abstract:

Conventional distance-based similarity measures are data-independent and sensitive to units or scales of measurement. There are existing data-dependent approaches, such as rank difference, Lin's probabilistic measure and m_p-dissimilarity, which are not sensitive to units or scales of measurement. Although they have been shown to be more effective than the traditional distance measures, their characteristics and relative performances have not been investigated.
In this paper, we study the characteristics and relationships of different data-dependent measures and find that m_p-dissimilarity is a generic data-dependent measure with data-dependent self-similarity, whereas rank difference and Lin's measure are special cases with data-independent self-similarity. We evaluate the effectiveness of a wide range of data-dependent and data-independent measures in content-based information retrieval and kNN classification tasks. Our findings show that the fully data-dependent measure of m_p is a more effective alternative to other data-dependent and commonly-used distance-based similarity measures as its task-specific performance is more consistent across a wide range of datasets.

Keywords: Distance measures, ℓ_p-norm, Lin's probabilistic similarity, rank transformation, data-dependent similarity measures, m_p-dissimilarity

4.1 Introduction

Measuring pairwise similarities of data instances is ubiquitous in many data-mining algorithms. The conventional approach to similarity measurement is primarily based on a geometric model where data are assumed to be embedded in a multi-dimensional space and the similarity of two instances is estimated as the inverse of their distance in the space (Deza and Deza, 2009). Minkowski distance (also known as ℓ_p-norm with $p > 0$) and cosine distance are the most widely-used distance measures.

In a geometric model, there is an implicit assumption that a unit distance implies the same degree of similarity everywhere in the space, which is referred to as *interval scale assumption* by Stevens (1946). This assumption can be problematic for two reasons:

1. *The need for data-dependence*: Psychologists (Tversky, 1977; Krumhansl, 1978) argue that the human-judged similarity between two instances is data-dependent as two instances in a dense region are less similar than two instances of equal distance in a less dense region. For example, many people earn in the range of 50 to 150 thousands, and significantly fewer persons earn more than one million a year. Two individuals earning $w = \$50\text{k}$ and $x = \$150\text{k}$ are judged to be less similar than two individuals earning $y = \$1100\text{k}$ and $z = \$1200\text{k}$, even though $z - y = x - w = \$100\text{k}$, because there are many more people earning in the range of \$50k to \$150k than those earning more than a million.
2. *Sensitivity to units or scales of measurement*: The commonly-used distance measures are sensitive to units or scales of measurement. For example, in the logarithmic scale of base 10, the annual incomes in the above example become $w' = 4.70$, $x' = 5.18$, $y' = 6.04$ and $z' = 6.08$. Although $x - w = z - y$ in the original scale, $x' - w' > z' - y'$ in the logarithmic scale. Unfortunately, the units or scales of measurement may not be known in many data-mining problems. Distance measures can produce poor task-specific performances if different units or scales of measurement are used in the same dataset (Fernando and Webb, 2017).

One simple solution is to assume that data are ordinal and use measures such as (1) rank difference - distance after rank transformation (Conover and Iman, 1981) and (2) Lin's information theoretic measure (Lin, 1998). They are scale-invariant and the similarity of two distinct values x and y (i.e., $x \neq y$) is data-dependent. However, their self-similarities are constant everywhere in the space, regardless of the data distribution. The similarity of two individuals earning \$50k is the same as two individuals earning \$1200k, even though the former is judged by humans to be less similar than the latter, because there are many more people earning \$50k than those earning \$1200k.

In order to understand the importance of data-dependent self-similarity, consider an example of a multidimensional dataset with 10 instances, the values of which in two dimensions i and j are distributed as shown in Table 4.1 (Aryal et al., 2017). In this example, $Inst1$ and $Inst2$ have the same values in both dimensions, but their value in dimension i is less common (has lower probability) than their value in dimension j. In measures with data-independent self-similarity, their similarities in two dimensions i and

Table 4.1: An example of data distribution in two dimensions of a multi-dimensional dataset (Aryal et al., 2017).

Dim.	$Inst1$	$Inst2$	$Inst3$	$Inst4$	$Inst5$	$Inst6$	$Inst7$	$Inst8$	$Inst9$	$Inst10$
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
i	2	2	1	1	1	1	1	1	1	1
j	2	2	2	2	2	2	2	2	1	1
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.

j have the same contribution to the overall similarity. Psychologists argue that having the same values in dimensions i (rare value) and j (common value) do not provide the same amount of information about the similarity of $Inst1$ and $Inst2$. The situation where many instances have the same value in many dimensions can be very common in high-dimensional spaces, as data often lie in a low-dimensional subspace.

Recently, Aryal et al. (2017, 2014b) introduced a fully data-dependent dissimilarity measure called m_p-dissimilarity (with $p > 0$) where even the self-similarity is data-dependent. In the example shown in Table 4.1, the similarity of $Inst1$ and $Inst2$ in dimension i is more than their similarity in dimension j using m_p as suggested by psychologists.

All these data-dependent measures have been shown to produce better task-specific performance than traditional distance measures, but their characteristics and relative performances have not been investigated. Furthermore, some data-dependent measures such as rank difference are inefficient to compute, particularly for testing instances which are new and not previously seen in the training data.

In this paper, we make the following contributions:

1. Analysis of the characteristics and relationships of different data-dependent measures. We generalise m_p from $p > 0$ to $p \geq 0$ by introducing m_0-dissimilarity and show that m_p is a generic data-dependent measure, where rank difference and Lin's measure are special cases of m_p with $p > 0$ and m_0, respectively, having *data-independent self-similarities*.

2. Evaluation of the task-specific performance and sensitivity to units or scales of measurement of a wide range of data-dependent and data-independent (distance-based) measures in content-based information retrieval (CBIR) and kNN classification tasks in a wide range of datasets. Our results show that (a) *data-dependent measures* produce more consistent results than commonly-used data-independent measures; (b) among data-dependent measures, those with *data-dependent self-similarity*, particularly m_0 (introduced in this paper), produce better results than those with *data-independent self-similarity* in bag-of-words (BoW) text datasets; and (c) most data-dependent measures are robust to units or scales of measurement.

In addition, based on the methodology used by m_p (Aryal et al., 2017), we improve the efficiency and effectiveness of rank difference and Lin's measure as follows:

i. The efficiency of rank difference is improved by converting continuous domain into ordinal intervals through discretisation, as in the case of m_p (Aryal et al., 2017). With this improvement, rank difference, Lin's measure and m_p have runtimes similar to those of the traditional distance measures.

ii. Adapting the formulations of data-dependent similarity measures of rank difference, Lin's measure and m_p so that they work well in datasets with bag-of-words (BoW) vector representations.

The rest of the paper is organised as follows. Section 4.2 reviews existing data-dependent (dis)similarity measures, along with some widely-used data-independent distance measures. In Section 4.3, we discuss the characteristics and relationships of different data-dependent measures and introduce a new variant of m_p with $p = 0$ (i.e., m_0). Section 4.4 discusses the adaptation of data-dependent measures for bag-of-words (BoW) vector representations. Converting real valued attributes into ordinal intervals in order to speed up data-dependent measures is discussed in Section 4.5. Section 4.6 presents empirical evaluation results, followed by discussion in Section 4.7. Finally, we conclude the paper in the last section.

4.2 Similarity or dissimilarity measures

Let D be a collection of N data instances where each instance $\mathbf{x}$ is represented by an M-dimensional vector of numerical values of its M selected features, i.e., $\mathbf{x} = \langle x_1, x_2, \cdots, x_M \rangle$. Let $d(\mathbf{x}, \mathbf{y})$ be a measure of dissimilarity[1] of $\mathbf{x}$ and $\mathbf{y}$. In the traditional approach, D is assumed to be embedded in an M-dimensional metric space and $d(\mathbf{x}, \mathbf{y})$ is computed as their geometric (spatial or angular) distance in that space.

Minkowski distance (also known as ℓ_p-norm) estimates $d(\mathbf{x}, \mathbf{y})$ as the power mean with $p > 0$ of distances in each dimension. Euclidean distance (ℓ_2-norm) is a popular choice, as it intuitively corresponds to the distance defined in the real three-dimensional world.

In high-dimensional sparse data distributions such as bag-of-words (BoW) text datasets (Salton and McGill, 1986), cosine distance (also known as angular distance) is a more sensible choice, because the direction of vectors is more important than their lengths. Note that the cosine distance of two vectors is proportional to their Euclidean distance if the vectors are normalised to unit lengths.

Minkowski distance becomes meaningless as the number of dimensions increases, because all pairs of points become almost equidistant in a high-dimensional space (Beyer et al., 1999; François et al., 2007). Recently, Mansouri and Khademi (2015) introduced multiplicative distance where $d(\mathbf{x}, \mathbf{y})$ is computed as the product of their distances in every dimension, and showed that it is more effective than the traditional Minkowski distance in high-dimensional spaces.

The formulations of Minkowski, cosine and multiplicative distances are provided in Table 4.2.

[1]Similarity is the inverse of dissimilarity. We use dissimilarity in this paper to be consistent with distance measures.

Table 4.2: Distance measures. $abs(\cdot)$ returns an absolute value and $p > 0$

Distance measure	Notation	$d(\mathbf{x}, \mathbf{y})$
Minkowski	$\ell_p(\mathbf{x}, \mathbf{y})$	$\left(\sum_{i=1}^{M} abs(x_i - y_i)^p\right)^{\frac{1}{p}}$
Cosine	$d_{cos}(\mathbf{x}, \mathbf{y})$	$1 - \frac{\sum_{i=1}^{M} x_i \times y_i}{\sqrt{\sum_{i=1}^{M} x_i^2} \times \sqrt{\sum_{i=1}^{M} y_i^2}}$
Multiplicative	$d_{MD}(\mathbf{x}, \mathbf{y})$	$\left(\prod_{i=1}^{M} (1 + abs(x_i - y_i))\right) - 1$

In all distance measures discussed above, $d(\mathbf{x}, \mathbf{y})$ is *data-independent* because it is estimated using the feature values of $\mathbf{x}$ and $\mathbf{y}$ only and the distribution of data has no influence on it. This has been suspected to be one of the key reasons why a distance measure that performs well in one data distribution can perform poorly in others (Aryal et al., 2014b). Several data-dependent measures are discussed in the literature which utilise information from the distribution of data, i.e., $d(\mathbf{x}, \mathbf{y})$ is *data-dependent*. We review some of them in this section.

4.2.1 Mahalanobis distance and metric learning

The Mahalanobis distance (Mahalanobis, 1936; Deza and Deza, 2009) of $\mathbf{x}$ and $\mathbf{y}$ is defined as follows:

$$d_{mah}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y})} \tag{4.1}$$

where $\Sigma \in \mathbb{R}^{M \times M}$ is the covariance matrix of D and T is a transpose operator.

Although it takes into account the differences in variance across different dimensions and captures covariance between them, it does not consider variation in local data distribution within a dimension. Covariance is not enough to capture the characteristics of non-normal distributions, and it can therefore perform poorly in many real-world problems where data distribution is often non-normal.

Instead of using the inverse of the covariance matrix, distance metric learning algorithms (Wang and Sun, 2015; Weinberger et al., 2006) learn a generalised Mahalanobis distance from D defined as follows:

$$d_{genMah}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \Omega (\mathbf{x} - \mathbf{y})} \tag{4.2}$$

where $\Omega \in \mathbb{R}^{M \times M}$ is a positive semi-definite matrix. It can be factorized as $\Omega = \Lambda^T \Lambda$ where $\Lambda \in \mathbb{R}^{\omega \times M}$ and ω is a positive integer. Hence, distance metric learning can be expressed as: $d_{genMah}(\mathbf{x}, \mathbf{y}) = \|\Lambda \mathbf{x} - \Lambda \mathbf{y}\|_2$ (Wang and Sun, 2015). The generalised Mahalanobis distance is the Euclidean distance of vectors projected by Λ.

The main learning task is to learn a projection matrix Λ to improve the task-specific performance of the Euclidean distance, subject to some constraints. For example, in a classification problem, the task is to learn Λ such that instances belonging to the same class

become closer to each other (similarity constraints) and instances belonging to different classes are separated further apart (dissimilarity constraints) (Weinberger et al., 2006).

4.2.2 Rank difference

In rank transformation (Conover and Iman, 1981), feature values of instances in each dimension are ranked in ascending order with the smallest value having rank 1, the second smallest value having rank 2, and so on. If n ($\leq N$) instances have the same value, and the value has rank r, then all these n instances are assigned the average rank $\frac{r+(r+1)+\cdots+(r+n)}{n}$ and the next available rank is $r+n$. The dissimilarity of $\mathbf{x}$ and $\mathbf{y}$ is estimated by aggregating their rank difference in each dimension using the same power mean formulation as in ℓ_p as follows:

$$d_{rank}(\mathbf{x}, \mathbf{y}, p) = \left(\frac{1}{M} \sum_{i=1}^{M} abs(\tilde{x}_i - \tilde{y}_i)^p \right)^{\frac{1}{p}} \tag{4.3}$$

where $\tilde{x}_i$ and $\tilde{y}_i$ are the ranks of x_i and y_i in dimension i.

Rank difference is data-dependent because for a given magnitude difference $abs(x_i - y_i) > 0$, $abs(\tilde{x}_i - \tilde{y}_i)$ is higher if x_i and y_i are located in a dense region than in a sparse region. However, the self-dissimilarity is data-independent, i.e., $abs(\tilde{x}_i - \tilde{x}_i) = 0$ everywhere in the space.

4.2.3 Lin's probabilistic measure

Assuming data are ordinal in each dimension, the dissimilarity[2] of $\mathbf{x}$ and $\mathbf{y}$ can be estimated using Lin's probabilistic measure (Lin, 1998) as follows:

$$d_{lin}(\mathbf{x}, \mathbf{y}) = 1 - \frac{1}{M} \sum_{i=1}^{M} \frac{2 \times \log \sum_{z_i=\min(x_i,y_i)}^{\max(x_i,y_i)} P(z_i)}{\log P(x_i) + \log P(y_i)} \tag{4.4}$$

where $P(x_i)$ is the probability of x_i which can be 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 x_i in D and u_i is the total number of distinct values in dimension i. Note that the default base of the logarithm in this paper is e (i.e., natural logarithm) unless specified otherwise.

Although the dissimilarity of x_i and y_i is data-dependent, the self-dissimilarity is constant, regardless of the data distribution.

4.2.4 Random forest-based measures

Shi and Horvath (2006) introduced a similarity measure based on unsupervised random forest (URF) (Breiman, 2001). Recently, Fernando and Webb (2017) used a different implementation of trees called unsupervised stochastic forest (USF), where each tree is built from a small random subsample of data ($\mathcal{D}_i \subset D, |\mathcal{D}_i| = \psi \ll N$). At each internal node in a tree, subsamples are partitioned into two equal subsets by splitting at the median

[2] Author(s) defined it as a similarity measure, but we define it as a dissimilarity measure to be consistent with other measures.

of values in a randomly-chosen attribute. It builds t balanced binary trees, each with ψ leaf nodes. The dissimilarity [3] of $\mathbf{x}$ and $\mathbf{y}$ is estimated as follows:

$$d_{USF}(\mathbf{x}, \mathbf{y}) = 1 - \frac{1}{t} \sum_{j=1}^{t} I\left(L_j(\mathbf{x}) = L_j(\mathbf{y})\right) \tag{4.5}$$

where $I(\cdot)$ is the indicator function, and $L_j(\mathbf{x})$ is the leaf node where $\mathbf{x}$ falls in j^{th} tree.

Fernando and Webb (2017) showed that d_{USF} produces competitive task-specific results in comparison to URF and rank difference but runs faster because (a) trees in USF are shallower than those in URF; and (b) rank difference is very expensive as it requires a range search to find the rank of a previously unseen value in each dimension. It is data-dependent for $\mathbf{x} \neq \mathbf{y}$ because they are more likely to fall in the same leaf if they are in a sparse region because leaves in sparse regions are larger than those in dense regions. However, the self-dissimilarity is zero everywhere in the space, regardless of the data distribution.

4.2.5 m_p-dissimilarity

Aryal et al. (2014b, 2017) introduced a fully data-dependent dissimilarity measure called m_p-dissimilarity with $p > 0$ where even the self-dissimilarity is data-dependent. In each dimension, rather than using the spatial distance $abs(x_i - y_i)$, the dissimilarity is estimated as the probability mass in a region covering x_i and y_i. The measure uses the similar power mean formulation with $p > 0$ as in ℓ_p to aggregate dissimilarities in each dimension.

$$m_p(\mathbf{x}, \mathbf{y}) = \left(\frac{1}{M} \sum_{i=1}^{M} \left(\frac{|R_i(\mathbf{x}, \mathbf{y})|}{N} \right)^p \right)^{\frac{1}{p}} \tag{4.6}$$

where $R_i(\mathbf{x}, \mathbf{y}) = [\min(x_i, y_i), \max(x_i, y_i)]$ and $|R_i(\mathbf{x}, \mathbf{y})| = |\{z_i : \min(x_i, y_i) \leq z_i \leq \max(x_i, y_i)\}|$.

It has been shown that m_p is more effective than ℓ_p and cosine distance, particularly in high-dimensional datasets (Aryal et al., 2014b, 2017).

4.2.6 Probability mass-based dissimilarity measure using trees

In a recent study, Ting et al. (2016) used a probability mass-based dissimilarity measure where the dissimilarity between $\mathbf{x}$ and $\mathbf{y}$ is estimated as the average probability data mass in the deepest node shared by them in a collection of t trees. They used the tree implementation of isolation forest (IF) (Liu et al., 2008), where each tree is built from a small subsample of data ($\mathcal{D}_i \subset D, |\mathcal{D}_i| = \psi \ll N$) with random non-empty partitioning (attribute and split point are selected randomly at each intermediate node) of the space until instances in $\mathcal{D}_i$ are isolated or the tree height reaches the maximum of $\log_2 \psi$. Once a tree is built, the data mass in each node is calculated from the entire data D. Using a

[3] Author(s) defined it as a similarity measure, but we define it as a dissimilarity measure to be consistent with other measures.

collection of t trees, the dissimilarity of $\mathbf{x}$ and $\mathbf{y}$ is estimated as follows:

$$m_{IF}(\mathbf{x}, \mathbf{y}) = \frac{1}{t} \sum_{j=1}^{t} \left(\frac{|R_j(\mathbf{x}, \mathbf{y})|}{N} \right) \tag{4.7}$$

where $R_j(\mathbf{x}, \mathbf{y})$ is the deepest node where $\mathbf{x}$ and $\mathbf{y}$ appear together in j^{th} tree.

4.3 Characteristics and relationships of data-dependent measures

On the basis of data-dependence, the data-dependent dissimilarity measures discussed in Section 4.2 can be categorised into two groups:

1. ***data-dependent projections***: Data are projected into a new space through a data-dependent projection and the dissimilarity of two instances is estimated using distance measures (usually ℓ_2) in the projected space. Mahalanobis distance and distance metric learning-based measures are examples of such measures. Even though the projection into a new space maintains the geometric interpretation, it is difficult to interpret the meaning of the new dimensions.
2. ***data-dependent measures***: Dissimilarities of instances are estimated using data-dependent measures in the original space. They can be further categorised into two subgroups:
   2.1. *One-dimensional data-dependent measures*: The dissimilarity of two instances is estimated by aggregating data-dependent dissimilarities in each dimension in the original space. Examples are d_{rank}, d_{lin} and m_p.
   2.2. *Tree-based data-dependent measures*: The dissimilarity of two instances is estimated by aggregating data-dependent dissimilarities w.r.t. subsets of dimensions in the original space using tree structures. Examples are d_{USF} and m_{IF}.

The data-dependent characteristic in some of these measures is applicable only for $\mathbf{x} \neq \mathbf{y}$ and self-dissimilarity is always a constant. The characteristics of all the data-dependent measures reviewed in the last section are summarised in Table 4.3.

Distance metric learning (d_{genMah}) learns the best projection matrix Λ by optimising task-specific constraints in a given dataset. Metric learned for one task may not be good for other tasks in the same dataset. It is not a general-purpose data-dependent measure like others discussed in Table 4.3. It is computationally expensive in high-dimensional and/or large datasets. Different distance metric learning algorithms for different tasks are discussed in the literature. We refer interested readers to the latest survey papers by Kulis (2013) and Wang and Sun (2015). In this paper, our primary focus is on general-purpose data-dependent measures which do not require learning and optimisation.

Note that the formulation of m_{IF} is similar to m_1 except that $R_i(\mathbf{x}, \mathbf{y})$ is considered and implemented differently. Unlike in m_1, where regions are defined in each dimension separately only after $\mathbf{x}$ and $\mathbf{y}$ are given, multi-dimensional regions are defined by random partitioning of the space using multiple trees in m_{IF}.

Table 4.3: Characteristics of data-dependent measures

	Measure	Basis of data-dependence	data-dependent? $\mathbf{x} \neq \mathbf{y}$	$\mathbf{x} = \mathbf{y}$	Learning required?
Projection based	d_{mah}	Projection based on data covariance	✓	×	×
	d_{genMah}	Projection based on task-specific constraints in the given dataset	✓	×	✓
1-dimensional	d_{rank}	Measure based on rank difference	✓	×	×
	d_{lin}	Measure based on data mass	✓	×	×
	m_p	Measure based on data mass	✓	✓	×
Tree-based	d_{USF}	Definition or size of regions	✓	×	×
	m_{IF}	Definition or size of regions and measure based on data mass	✓	✓	×

All one-dimensional data-dependent dissimilarity measures (m_p, d_{rank} and d_{lin}) assume that data are ordinal and estimate $d(\mathbf{x}, \mathbf{y})$ based on the probability mass distribution of values in each dimension.

The difference and similarity between d_{rank} and m_p are as follows:

- Similar under uniform data distribution. Note that $|R_i(\mathbf{x}, \mathbf{y})| = abs(\tilde{x}_i - \tilde{y}_i) + \frac{f(x_i)+f(y_i)}{2}$. As a result, d_{rank} and m_p are equivalent if the probability mass distribution over u_i possible values in dimension i is uniform (i.e. $\forall_{x_i,y_i}\ f(x_i) = f(y_i) = b_i$ where $b_i = \frac{N}{u_i}$ is a constant) because $|R_i(\mathbf{x}, \mathbf{y})| = abs(\tilde{x}_i - \tilde{y}_i) + b_i$. They are different if the probability mass distribution is not uniform.
- Difference in self-dissimilarity. For example, in the case of annual income where there are significantly more people earning x_i = \$50k than those earning y_i = \$1m (i.e., $f(x_i) > f(y_i)$), the dissimilarity of two individuals earning \$50k is higher than that of two individuals earning \$1m under m_p because $|R_i(\mathbf{x}, \mathbf{x})| > |R_i(\mathbf{y}, \mathbf{y})|$. In contrast, they both are zero under d_{rank} because $abs(\tilde{x}_i - \tilde{x}_i) = abs(\tilde{y}_i - \tilde{y}_i) = 0$. In other words, the self-dissimilarity is data-independent in d_{rank}, whereas it is data-dependent in m_p.

The relationship of m_p with d_{lin} is not straightforward. In the next subsection, we generalise m_p where p is allowed to be zero by introducing a new variant of m_p with $p = 0$ (m_0) and then discuss its relationship with d_{lin}.

4.3.1 m_0-dissimilarity

m_0-dissimilarity estimates the dissimilarity of $\mathbf{x}$ and $\mathbf{y}$ as the geometric mean of their probability mass-based dissimilarities in each dimension.

$$m_0(\mathbf{x}, \mathbf{y}) = \left(\prod_{i=1}^{M} \frac{|R_i(\mathbf{x}, \mathbf{y})|}{N} \right)^{\frac{1}{M}} \tag{4.8}$$

It can be shown that $m_p \to m_0$ when $p \to 0$. Let $A = \frac{1}{M}$ and $\alpha_i = \frac{|R_i(\mathbf{x},\mathbf{y})|}{N}$. Eqn 4.6 can then be rewritten as:

$$m_p(\mathbf{x},\mathbf{y}) = \exp\left(\log\left(\left(A\sum_{i=1}^{M}\alpha_i^p\right)^{\frac{1}{p}}\right)\right) = \exp\left(\frac{\log\left(A\sum_{i=1}^{M}\alpha_i^p\right)}{p}\right)$$

At the limit when $p \to 0$, the exponential component can be estimated using L'Hopital rule as:

$$\lim_{p\to 0}\frac{\log\left(A\sum_{i=1}^{M}\alpha_i^p\right)}{p} = \lim_{p\to 0}\frac{A\sum_{i=1}^{M}\alpha_i^p\log\alpha_i}{A\sum_{i=1}^{M}\alpha_i^p} = \frac{\sum_{i=1}^{M}\log\alpha_i}{M} = \log\left(\prod_{i=1}^{M}\alpha_i\right)^{\frac{1}{M}}$$

Using the above two equations and substituting α_i, we obtain Eqn 4.8.

It has a nice 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 $\mathbf{y}$ in the space which has the length of $R_i(\mathbf{x},\mathbf{y})$ in dimension i, and estimate the probability data mass in the region. In other words, it estimates the probability of a randomly-selected point $\mathbf{z}$ falling in the region, i.e., $P(\mathbf{z} \in R(\mathbf{x},\mathbf{y}))$. In order to have a reasonable estimate of $P(\mathbf{z} \in R(\mathbf{x},\mathbf{y}))$, a large amount of data is required in high-dimensional spaces. Assuming that the dimensions are independent, $P(\mathbf{z} \in R(\mathbf{x},\mathbf{y}))$ can be approximated from the observed data as follows:

$$\hat{P}(\mathbf{z} \in R(\mathbf{x},\mathbf{y})) \approx \prod_{i=1}^{M} P(z_i \in R_i(\mathbf{x},\mathbf{y})) \tag{4.9}$$

where $P(z_i \in R_i(\mathbf{x},\mathbf{y}))$ is the probability of z_i falling in $R_i(\mathbf{x},\mathbf{y})$ in dimension i which can be estimated from the observed data D as $\hat{P}(z_i \in R_i(\mathbf{x},\mathbf{y})) = \frac{|R_i(\mathbf{x},\mathbf{y})|}{N}$. The outer power of $\frac{1}{M}$ in Eqn 4.8 is simply a scaling factor of the dissimilarity, and does not change the similarity rankings of instances.

In order to avoid floating point overflow in the case of large M, $\log m_0(\mathbf{x},\mathbf{y})$ is used as the degree of dissimilarity of $\mathbf{x}$ and $\mathbf{y}$, effectively using the summation of the logarithm of dissimilarities in each dimension.

$$\log m_0(\mathbf{x},\mathbf{y}) = \frac{1}{M}\sum_{i=1}^{M}\log\left(\frac{|R_i(\mathbf{x},\mathbf{y})|}{N}\right) \tag{4.10}$$

Now, it is interesting to note the relationship of m_0 (as defined in Eqn 4.10) with d_{lin} (defined in Eqn 4.4) since $\log\left(\frac{|R_i(\mathbf{x},\mathbf{y})|}{N}\right) = \log\sum_{z_i=\min(x_i,y_i)}^{\max(x_i,y_i)} P(z_i)$. Unlike in d_{lin}, the dissimilarity in each dimension in m_0 is not normalised by $(\log P(x_i) + \log P(y_i))$ resulting in data-dependent self-dissimilarity when probability mass distribution in dimension i is non-uniform, which is the only difference between them.

In other words, we can consider m_p $(p \geq 0)$ as a generic, fully data-dependent measure of which d_{rank} and d_{lin} are special cases of m_p $(p > 0)$ and m_0, respectively, where the special cases have *data-independent self-dissimilarity* and the generic measure has *data-dependent self-dissimilarity.*

Note that m_p ($p > 0$) and m_0 are data-dependent counterparts of the Minkowski distance (ℓ_p) and the multiplicative distance (d_{MD}), respectively, where the dissimilarity of two instances in each dimension is estimated using probability data mass between them instead of using spatial distance.

4.4 (Dis)similarity measures in bag-of-words vector representation

In the bag-of-words (BoW) (Salton and McGill, 1986) vector representation, each component of a vector represents the frequency of a feature (i.e., a term in a document). Many components of a vector representing a document have zero value because a document contains only a small proportion of words in the dictionary, resulting in a sparse vector representation. Euclidean distance is not a good choice for such problems (Salton and McGill, 1986; Salton and Buckley, 1988). The direction of a vector is more important than its length. Hence, cosine distance is a more sensible choice to measure dissimilarity between two documents. It has been shown that the cosine distance with inverse document frequency (IDF) (Salton and Buckley, 1988) weighted vectors produces better results than the cosine distance with unweighted vectors. The assumption of IDF weighting is that rare terms are more important than frequent terms. The IDF weighted vector component of ith term in a document $\mathbf{x}$ is estimated as $x'_i = x_i \times \log \frac{N}{df_i}$ where df_i is the number of documents in a corpus in which the ith term occurs. The dissimilarity of two documents is estimated using cosine distance as:

$$d_{cosIdf}(\mathbf{x}, \mathbf{y}) = d_{cos}(\mathbf{x}', \mathbf{y}') \tag{4.11}$$

In data-dependent measures such as rank difference, Lin and m_p, explicit IDF weighting is not required as they use a similar statistic based on the number of documents in the measure itself. However, they require a simple adjustment in their formulations. Since the absence of a term in both $\mathbf{x}$ and $\mathbf{y}$ (i.e., $x_i = y_i = 0$) does not provide any information about their (dis)similarity, these terms should be ignored. Those terms where $x_i = y_i = 0$ are ignored implicitly in the cosine distance as they do not affect any terms in its formulation. Aryal et al. (2015, 2017) re-defined m_p ($p > 0$) (Eqn 4.6) in the BoW vector representation as follows:

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

where $F_{\mathbf{x}}$ is the set of indices of terms 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 terms used for measuring dissimilarity of any two documents.

(a) Two clusters (b) Outlier

Figure 4.1: Situations where equal-width discretisation (EWD) can be problematic for dissimilarity measurement.

Using the same idea, the other three one-dimensional data-dependent measures - rank difference (Eqn 4.3), Lin's measure (Eqn 4.4) and m_0-dissimilarity (Eqn 4.10) are re-defined in the BoW vector representation as follows:

$$d^{bow}_{rank,p}(\mathbf{x},\mathbf{y}) = \left(\frac{1}{|F_{\mathbf{x},\mathbf{y}}|} \sum_{i \in F_{\mathbf{x},\mathbf{y}}} (\tilde{x}_i - \tilde{y}_i)^p \right)^{\frac{1}{p}}$$

$$d^{bow}_{lin}(\mathbf{x},\mathbf{y}) = 1 - \frac{1}{|F_{\mathbf{x},\mathbf{y}}|} \sum_{i \in F_{\mathbf{x},\mathbf{y}}} \frac{2 \times \log \sum_{z_i=\min(x_i,y_i)}^{\max(x_i,y_i)} P(z_i)}{\log P(x_i) + \log P(y_i)}$$

$$\log m_0^{bow}(\mathbf{x},\mathbf{y}) = \frac{1}{|F_{\mathbf{x},\mathbf{y}}|} \sum_{i \in F_{\mathbf{x},\mathbf{y}}} \log \frac{|R_i(\mathbf{x},\mathbf{y})|}{N}$$

4.5 Numeric to ordinal conversion to speed up one-dimensional data-dependent measures

The one-dimensional data-dependent measure of d_{rank} is computationally expensive because computing the rank of an unseen feature value in a set of N seen values in each dimension is in the order of $O(\log N)$ using a binary search (Fernando and Webb, 2017). It is infeasible to use in datasets with large N and/or M. The time complexity can be reduced by converting a continuous valued domain in each dimension into an ordinal discrete domain by discretising the range of data values into $\eta \ll N$ intervals, as done by Aryal et al. (2017) in the case of m_p. By computing and storing the frequency and rank of each interval from the seen data in the pre-processing step, the rank of an unseen value can be estimated by finding the interval in which it falls in $O(\log \eta)$. Unlike the ranking of the original values which is strictly monotonic, i.e., $a < b \implies \tilde{a} < \tilde{b}$, ranking intervals is weakly monotonic, i.e., $a < b \implies \tilde{A} \leq \tilde{B}$ where A and B are the intervals in which a and b fall.

If the number of unique values in the seen data $u \leq \eta$, sorted one-dimensional data can be discretised easily by splitting at the mid-point of each pair of consecutive unique values. In the case of $u > \eta$, discretisation can be done in main two ways: (i) equal-width discretisation (EWD), where intervals are of the same width, and (ii) equal-frequency discretisation (EFD), where intervals have the same frequency. In EFD, it may not be possible to have the same frequency in each interval because of duplicate values; hence intervals are created to have approximately the same frequency.

EWD may not be effective to measure dissimilarities of data in some distributions, such as those shown in Figures 4.1(a) and 4.1(b), where the data fall in intervals at the

Table 4.4: Time and space complexities. Note that the time complexity in the last column is to compute dissimilarity of a pair of instances in program execution. N: Number of instances, M: Number of dimensions, t: Number of trees in forest-based methods, ψ: Subsample size to build trees in forest-based methods, η: Average number of intervals over all dimensions in the EFD.

Measures	Pre-processing		$d(\mathbf{x}, \mathbf{y})$ in execution
	Time	Space	Time
ℓ_p, d_{MD} and d_{cos}	$O(NM)$	$O(M)$	$O(M)$
d_{mah}	$O(NM^2)$	$O(M^2)$	$O(M)$
d_{USF}	$O(t\psi \log_2 \psi)$	$O(t\psi)$	$O(t \log_2 \psi)$
d_{IF}	$O(tN \log_2 \psi + t\psi^2)$	$O(t\psi^2)$	$O(t \log_2 \psi)$
d_{rank}, d_{lin} and m_p	$O(NM\eta + M\eta^2)$	$O(M\eta^2)$	$O(M \log_2 \eta)$

Complexities of distance metric learning (d_{genMah}) are not included in the table as they depend on the constraints and optimisation techniques used. Different metric learning algorithms are discussed in Kulis (2013) and Wang and Sun (2015).

two ends and many intervals in the middle are empty. Many instances falling in the same interval in dense regions become equally similar to each other and cannot be differentiated. This issue is less severe in EFD, where only $\varphi = \lceil \frac{N}{\eta} \rceil$ instances are allowed to be in the same interval. Hence, EFD is used to convert numeric values in continuous domain into ordinal intervals, as done by Aryal et al. (2017). Note that intervals can have different number of instances if more than φ instances have the same value (i.e., there are many duplicate values), resulting in data-dependent self-dissimilarity even though EFD is used. Many instances having the same value in many dimensions is a common characteristic of many high-dimensional datasets as data often lie in a low-dimensional subspace.

4.5.1 Time and space complexities

In each dimension, pairwise dissimilarities of intervals can be pre-computed from the observed data and stored as a matrix. Similar pre-processing can be done in m_{IF} where pairwise dissimilarity of each pair of leaf nodes in each tree is pre-computed. After pre-processing, the dissimilarity between two instances in each dimension can be computed as a table look-up by finding intervals or leaves where they fall. The time and space complexities of pre-processing along with the time complexities to estimate $d(\mathbf{x}, \mathbf{y})$ using dissimilarity measures discussed in Section 4.2 are provided in Table 4.4.

Note that distance measures such as ℓ_p, d_{cos} and d_{MD} require pre-processing to normalise values in each dimension to be in the unit range. The constants in time complexities to compute $d(\mathbf{x}, \mathbf{y})$ in the program runtime are higher in distance measures than in data-dependent measures, because of the floating-point operations to compute distance in each dimension. Floating-point operations are not required in data-dependent measures to compute dissimilarity in each dimension as it is done as a table look-up.

Table 4.5: Characteristics of datasets in terms of the number of instances (N), number of dimensions (M) and number of classes (C). The last six datasets are bag-of-words (BoW) text datasets

Name	N	M	C	Application area
Gas	13790	128	6	Chemistry (gases)
Ismis	12495	191	6	Music collection
Corel	10000	67	100	Image collection
SatImg	6435	36	6	Satellite images
Blocks	5473	10	5	Web design (webpage blocks)
Mfeat	2000	649	10	Handwritten digits
Steel	1941	25	7	Steel plates manufacturing
ImgSeg	1500	19	7	Image segmentation
Hba	1500	187	15	Music collection
Gtzan	1000	230	10	Music collection
NG20	18821	5489	20	20 Newsgroups text collection
Ohscal	11162	11465	10	Ohsumed patients' document collection
R52	9100	7369	52	Reuters (52 classes) collection
R8	7674	3497	8	Reuters (8 classes) collection
Fbis	2463	2000	17	TREC document collection
Wap	1560	8460	20	Yahoo web pages collection

4.6 Empirical evaluation

In order to evaluate the relative performance of data-dependent and data-independent (distance) measures discussed in Section 4.2, we used them in two data-mining tasks (a) Content-based information retrieval (CBIR), where the task is to retrieve similar (relevant) instances to a given query instance from a database (i.e., query-by-example); and (b) kNN classification, where the task is to predict the class label of a test instance based on its k most similar (nearest neighbour) instances in a training set. We conducted a series of experiments to evaluate contending dissimilarity measures in terms of (i) task-specific performance and runtime; and (ii) sensitivity to units or scales of feature values.

4.6.1 Datasets

We used 16 datasets from different application areas with varying numbers of instances (N), numbers of dimensions (M) and numbers of classes (C). The properties of these datasets are provided in Table 4.5, of which the last six are bag-of-words (BoW) text datasets and the first 10 are non-BoW datasets.

Of the BoW text datasets, NG20, R52 and R8 were from Cardoso-Cachopo (2007)[4]; and Ohscal, Wap and Fbis were from Han and Karypis (2000)[5]. Of the non-BoW datasets, Corel and Gtzan were from Zhou et al. (2012); HBA was from Ariyaratne and Zhang (2012); Ismis was the dataset used in the International Symposium on Methodologies for

[4]http://web.ist.utl.pt/acardoso/datasets/
[5]http://www.cs.waikato.ac.nz/ml/weka/datasets.html

Intelligent Systems (ISMIS) 2011 music information retrieval contest[6]; and the remaining six non-BoW datasets were from the UCI Machine Learning repository (Bache and Lichman, 2013)[7].

4.6.2 Experimental set-up

Each dataset was divided into two subsets $\mathcal{D}$ and $\mathcal{Q}$ using 10-fold cross-validation, where 9 folds (90% of instances) were included in $\mathcal{D}$ and the remaining one fold (10% of instances) was included in $\mathcal{Q}$. In the CBIR task, $\mathcal{D}$ was used as a database from which relevant instances were extracted for each query in $\mathcal{Q}$. In the kNN classification task, $\mathcal{D}$ was used as the training set and $\mathcal{Q}$ was used as the testing set. We repeated the experiment 10 times using each of the 10 folds as $\mathcal{Q}$ and the remaining 9 folds as $\mathcal{D}$. The average task-specific performance and standard error over 10 runs were reported. The task-specific performances of two measures were considered to be significantly different if their confidence intervals based on two standard errors did not overlap.

In the CBIR task, for each query $\mathbf{q} \in \mathcal{Q}$, instances in $\mathcal{D}$ were ranked in ascending order of their dissimilarity to $\mathbf{q}$ using different dissimilarity measures. The top k instances were presented as the relevant instances to $\mathbf{q}$. For performance evaluation, an instance was considered to be relevant to $\mathbf{q}$ if they had the same category label. In order to demonstrate the consistency of measures at different top k retrieved results, we evaluated the precision at the top k retrieved results ($P@k$) with $k = 1, 2, \cdots, 25$ and used the mean average precision up to $k = 25$, $MAP@25 = \frac{\sum_{k=1}^{25} P@k}{25}$ as a performance evaluation criterion. The average $MAP@25$ and standard error over 10 runs were reported.

In the kNN classification task, a class label for a test instance $\mathbf{q} \in \mathcal{Q}$ was predicted using the class labels of its k least dissimilar (or nearest neighbours) instances in the training set $\mathcal{D}$ using different dissimilarity measures. We used $k = 5$, i.e., 5NN classification. The classification error in the test set $\mathcal{Q}$ was used as a performance evaluation criterion. The average classification error and standard error over 10 runs were reported.

For distance-based measures, min-max normalisation was done using the data ranges in $\mathcal{D}$ in each dimension and the same range was used to normalise the instances in $\mathcal{Q}$. Similarly, in the BoW text datasets, IDF term-weighting factors were estimated from $\mathcal{D}$ and the same weights were used for documents in both $\mathcal{D}$ and $\mathcal{Q}$.

In the tree-based measures (d_{USF} and m_{IF}), the subsample size (ψ) was set to the default settings suggested by their respective authors - 32 in d_{USF} (Fernando and Webb, 2017) and 256 in m_{IF} (Ting et al., 2016). The ensemble size was set to $t = \max(100, M)$ to ensure that the ensemble size was sufficiently large in high-dimensional datasets. The number of intervals (η) in EFD (Section 4.5) was set as default to $\eta = \lfloor \log_2 |\mathcal{D}| \rfloor + 1$, as suggested by Sturges (1926) for the number of histograms.

As d_{USF} and m_{IF} are random methods (they build trees from a small random subsample of $\mathcal{D}$), d_{USF} and m_{IF} experiments with each pair of $\mathcal{D}$ and $\mathcal{Q}$ sets were repeated 10 times and the average result was considered.

[6] http://tunedit.org/challenge/music-retrieval
[7] https://archive.ics.uci.edu/ml/datasets.html

Table 4.6: Average MAP@25 and standard error (within the parentheses in the second row in small font) over 10 runs in non-BoW datasets. The best result is underlined and the results equivalent (insignificant difference based on two standard errors) to the best result are bold-faced.

	Gas	Ismis	Corel	SatImg	Blocks	Mfeat	Steel	SegImg	Hba	Gtzan
d_{cos}	**<u>0.982</u>**	**0.898**	0.155	0.804	0.948	**0.954**	0.591	0.897	0.357	**0.514**
	(0.001)	(0.001)	(0.001)	(0.002)	(0.001)	(0.003)	(0.002)	(0.005)	(0.003)	(0.009)
ℓ_2	**0.981**	**0.897**	0.151	**0.869**	0.938	**0.954**	0.590	0.899	0.370	**0.512**
	(0.001)	(0.002)	(0.002)	(0.002)	(0.001)	(0.003)	(0.003)	(0.004)	(0.004)	(0.010)
ℓ_1	**<u>0.982</u>**	**<u>0.899</u>**	0.202	**<u>0.872</u>**	0.944	**<u>0.955</u>**	0.610	**<u>0.908</u>**	0.421	0.498
	(0.001)	(0.001)	(0.001)	(0.002)	(0.001)	(0.003)	(0.003)	(0.004)	(0.004)	(0.009)
d_{MD}	**<u>0.982</u>**	**0.898**	0.207	**0.871**	0.945	**<u>0.955</u>**	0.613	**0.908**	0.425	0.495
	(0.001)	(0.001)	(0.001)	(0.002)	(0.001)	(0.003)	(0.003)	(0.004)	(0.005)	(0.008)
d_{mah}	0.959	0.768	0.133	0.603	0.949	0.422	0.597	0.838	0.192	0.229
	(0.001)	(0.002)	(0.001)	(0.002)	(0.001)	(0.003)	(0.005)	(0.007)	(0.004)	(0.004)
m_{IF}	0.977	0.883	0.196	0.865	0.949	**0.950**	0.609	0.899	0.443	0.507
	(0.001)	(0.002)	(0.001)	(0.002)	(0.001)	(0.003)	(0.004)	(0.004)	(0.005)	(0.007)
d_{USF}	0.979	0.869	0.213	0.864	**0.951**	0.945	0.602	0.898	0.443	0.495
	(0.001)	(0.002)	(0.001)	(0.002)	(0.002)	(0.003)	(0.004)	(0.004)	(0.005)	(0.007)
d_{rank}	**0.980**	**0.898**	**<u>0.256</u>**	0.866	**0.953**	**0.953**	**<u>0.627</u>**	0.899	**0.489**	**0.520**
	(0.001)	(0.002)	(0.001)	(0.002)	(0.002)	(0.003)	(0.005)	(0.003)	(0.007)	(0.008)
d_{lin}	**0.980**	**0.898**	0.252	0.864	**0.953**	**0.949**	**0.618**	0.888	**0.497**	**<u>0.523</u>**
	(0.001)	(0.002)	(0.001)	(0.002)	(0.001)	(0.003)	(0.005)	(0.004)	(0.007)	(0.007)
m_1	**0.980**	**0.898**	**0.255**	0.865	**<u>0.954</u>**	**0.951**	**<u>0.627</u>**	0.897	**0.490**	**0.520**
	(0.001)	(0.002)	(0.001)	(0.002)	(0.002)	(0.003)	(0.005)	(0.003)	(0.007)	(0.008)
m_0	**0.980**	**0.898**	**0.255**	0.864	**0.953**	0.947	0.615	0.892	**<u>0.498</u>**	**<u>0.523</u>**
	(0.001)	(0.002)	(0.001)	(0.002)	(0.002)	(0.003)	(0.005)	(0.004)	(0.007)	(0.007)

All the experimental set-ups and dissimilarity measures were implemented in Python using Scikit-Learn Machine Learning Library (Pedregosa et al., 2011). All the experiments were conducted on a Linux machine with a 2.27 GHz processor and 16 GB memory.

4.6.3 Content-based information retrieval (CBIR) task

We present the CBIR results in non-BoW and BoW text datasets separately in the next two subsections.

CBIR in non-BoW datasets

In non-BoW datasets, we compared the effectiveness of data-dependent measures, Rank distance with $p = 1$ (d_{rank})[8], Lin's probabilistic measure (d_{lin}), measure based on unsupervised stochastic forest (d_{USF}), m_p-dissimilarity with $p = 0$ and 1 (m_0 and m_1) and mass-based dissimilarity measure using isolation forest (m_{IF}), against data-independent measures, Euclidean and Manhattan distances (ℓ_2 and ℓ_1), cosine distance (d_{cos}), multiplicative distance (d_{MD}) and Mahalanobis distance (d_{mah}). The average MAP@25 and standard error over 10 runs of all contending measures in the 10 non-BoW datasets are presented in Table 4.6.

The results in Table 4.6 show that even though data-independent measures (ℓ_2, ℓ_1, d_{cos} and d_{MD}) are good in some datasets, they perform poorly in others. Each of ℓ_2,

[8]Fernando and Webb (2017) have shown that it is a better alternative than any other p settings. Hereafter, to simplify notation, we refer to $d_{rank}(\mathbf{x}, \mathbf{y}, 1)$ as $d_{rank}(\mathbf{x}, \mathbf{y})$.

Table 4.7: Win:loss:draw counts of one-dimensional data-dependent measures against the other contenders based on two standard errors in the CBIR task in non-BoW datasets.

	d_{cos}	ℓ_2	ℓ_1	d_{MD}	m_{IF}	d_{USF}
m_0	5:1:4	4:2:4	4:3:3	4:3:3	6:0:4	5:0:5
m_1	5:0:5	4:0:6	5:2:3	5:2:3	6:0:4	5:0:5
d_{lin}	5:0:5	4:2:4	4:2:4	4:2:4	6:1:3	5:1:4
d_{rank}	5:0:5	4:0:6	5:2:3	5:2:3	6:0:4	6:0:4

ℓ_1 and d_{MD} produced the best or competitive results to the best performing measure in five datasets, followed by d_{cos} in four datasets. In contrast, data-dependent measures of d_{rank} and m_1 produced the best or competitive results to the best performing measure in eight datasets, followed by d_{lin} and m_0 in seven and six datasets, respectively. The tree-based data-dependent measures of d_{USF} and d_{IF} produced competitive result to the best performing measure in only one dataset each. The projection-based measure of d_{mah} did not produce competitive results to the best performing measure in any dataset. This could be due to the normality assumption of Mahalanobis distance.

The summarised CBIR results in 10 non-BoW datasets in terms of win:loss:draw counts of one-dimensional data-dependent measures (d_{rank}, d_{lin}, m_1 and m_0) against the other key contenders based on the two standard errors significance test are provided in Table 4.7. The table shows that they had more wins than losses over data-independent (distance) and tree-based data-dependent measures.

It is interesting to note that m_1 and m_0 produced similar results to d_{rank} and d_{lin}, respectively. As discussed in Section 4.3, m_1 is equivalent to d_{rank} and m_0 is equivalent to d_{lin} if probability mass in each dimension is uniformly distributed. Since there are not many x_i where $f(x_i) > \varphi$ (expected interval frequency) in each dimension, each interval has almost the same mass of φ because of the equal-frequency discretisation.

Another interesting result to note is the superior performance of m_1 over m_{IF}. The only difference between them is the implementation to define regions: multi-dimensional regions are defined through hierarchical partitions using trees in m_{IF}, whereas one-dimensional regions are defined through equal-frequency discretisation in m_1. The reason could be the hierarchical partitioning of the space (detailed discussion is provided in Section 4.7).

The relative average precision ($P@k$) of different measures at each of $k = 1, 2, \cdots, 25$ is consistent with the average $MAP@25$ results presented in Table 4.6. The $P@k$ for $k = 1, 2, \cdots, 25$ of ℓ_2, d_{cos}, m_{IF}, m_{USF}, d_{rank} and m_0 in the Corel and Hba datasets are presented in Figure 4.2.

In terms of runtime, all dissimilarity measures had comparable runtimes in all datasets. The average total runtime of pre-processing and retrieval of all queries in $\mathcal{Q}$ over 10 runs is presented in Table 4.8.

One-dimensional data-dependent measures d_{rank}, m_1, d_{lin} and m_0 run faster than distance-based measures such as ℓ_2 and d_{cos} because they do not require floating-point operations to compute dissimilarity in each dimension, which is done as a table look-up. They ran faster than tree-based data-dependent measures of m_{IF} because $\eta = \lfloor \log_2 N \rfloor + 1$ is generally smaller than $\psi = 256$. They ran slower than the tree-based measure of d_{USF} in

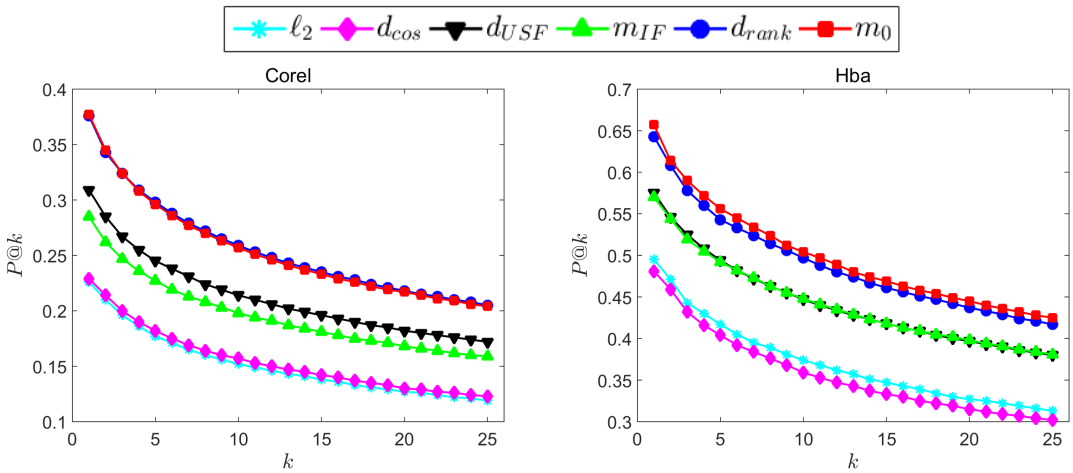


Figure 4.2: Average $P@k$ at $k = 1, 2, \cdots, 25$ in the Corel and Hba datasets.

Table 4.8: Average CBIR runtime (seconds) for a query set over 10 runs in non-BoW datasets. The presented runtime is the total runtime including pre-processing and retrieval time for all queries in the query set.

	Gas	Ismis	Corel	SatImg	Blocks	Mfeat	Steel	SegImg	Hba	Gtzan
d_{cos}	3043	2517	1534	585	353	66	50	35	31	14
ℓ_2	1949	1752	940	374	137	102	31	26	23	8
ℓ_1	1193	1151	601	245	128	54	16	15	12	7
d_{MD}	993	785	371	174	124	22	7	8	10	5
d_{mah}	13577	8027	697	307	104	1687	10	11	107	49
m_{IF}	1131	1467	635	247	180	231	23	20	31	22
d_{USF}	506	463	241	77	61	95	7	6	6	5
d_{rank}	576	369	175	111	30	29	5	7	11	7
d_{lin}	571	509	301	118	41	24	4	7	12	7
m_1	557	365	268	114	64	30	4	7	10	6
m_0	571	533	281	116	31	28	4	7	12	6

some datasets because of the pre-processing to compute the pairwise dissimilarity matrix of intervals in each dimension, which is not required in d_{USF}. The Mahlanobis distance (d_{mah}) was up to two orders of magnitude slower than other contending measures because computing the covariance matrix and its inverse can be expensive in datasets with large N and/or M.

CBIR in BoW text datasets

In the document retrieval task, we evaluated the retrieval results of the BoW versions of data-dependent measures d_{rank}, m_1, d_{lin} and m_0 (discussed in Section 4.4) against the cosine distance with and without IDF term weighting (d_{cosIdf} and d_{cos}). We did not consider the other distance-based measures such as ℓ_2 and ℓ_1 as contenders in the text datasets as they have been shown to produce significantly worse results than cosine distance (Salton and Buckley, 1988). We did not consider tree-based measures (d_{USF} and m_{IF}) as contenders in text datasets because they did not produce competitive results in

Table 4.9: Average MAP@25 and standard error (within the parentheses in small font) over 10 runs in BoW text datasets. The best result is underlined and the results equivalent (insignificant difference based on two standard errors) to the best result are bold-faced.

	NG20	Ohscal	R52	R8	Fbis	Wap
d_{cos}	0.542(0.002)	0.521(0.003)	**<u>0.843</u>**(0.003)	**0.909**(0.003)	**<u>0.687</u>**(0.005)	0.621(0.008)
d_{cosIdf}	0.701(0.002)	0.475(0.003)	0.803(0.003)	0.854(0.002)	0.674(0.006)	0.626(0.006)
d_{rank}	0.628(0.002)	0.588(0.002)	0.832(0.002)	**0.908**(0.002)	0.654(0.005)	0.660(0.006)
d_{lin}	0.638(0.002)	0.587(0.002)	0.832(0.002)	**0.907**(0.002)	0.653(0.005)	0.657(0.006)
m_1	0.653(0.002)	**0.589**(0.002)	0.833(0.002)	**0.907**(0.002)	0.662(0.005)	**<u>0.710</u>**(0.005)
m_0	**<u>0.715</u>**(0.001)	**<u>0.593</u>**(0.002)	**0.838**(0.002)	**<u>0.910</u>**(0.001)	0.672(0.005)	**0.708**(0.005)

Table 4.10: Win:loss:draw counts of m_0 and m_1 against the other contenders based on two standard errors in the CBIR task in BoW datasets.

	d_{cos}	d_{cosIdf}	d_{rank}	d_{lin}
m_0	3:1:2	5:0:1	5:0:1	5:0:1
m_1	3:2:1	4:2:0	2:0:4	2:0:4

non-BoW datasets where the number of dimensions is lower than in text datasets (we discuss their limitations in high-dimensional datasets in Section 4.7). MAP@25 of all contending dissimilarity measures in the six BoW text datasets are presented in Table 4.9.

Table 4.9 shows that m_0 produced better or competitive retrieval results to the best performing measure in five datasets, with the exception of Fbis where it produced a worse retrieval result than the best performing measure d_{cos}. The summarised CBIR results in six BoW datasets in terms of win:loss:draw counts of one-dimensional data-dependent measures with data-dependent self-dissimilarity (m_0 and m_1) against the other key contenders based on the two standard errors significance test provided in Table 4.10 show that they had more wins than losses over the cosine measures (d_{cos} and d_{cosIdf}) and one-dimensional data-dependent measures with data-independent self-dissimilarity (d_{lin} and d_{rank}).

Of the two variants of m_p, m_0 produced results better than or similar to m_1. m_1 produced significantly worse results than m_0 in NG20 and R52, while producing competitive results in others. Of the union of terms in documents $\mathbf{x}$ and $\mathbf{y}$ ($F_{\mathbf{x},\mathbf{y}}$), only a small proportion of terms occur in both $\mathbf{x}$ and $\mathbf{y}$. They have small dissimilarities w.r.t those few common terms because only a few documents have non-zero frequency of a term and large dissimilarities w.r.t many other terms that occur in either one of them only because many documents have zero frequency of a term. As those few small dissimilarities have a greater influence in the geometric mean than in the arithmetic mean, m_0 produced better similarity results than m_1[9].

It is interesting to note that m_0 and m_1 produced either similar or better retrieval results than d_{rank} and d_{lin} (although d_{rank} produced better retrieval results than m_1 in R8, the difference is not significant). The only difference between them is the data-dependent

[9]We also examined whether the geometric mean of rank differences produced better results than the arithmetic mean (d_{rank}); but we observed that it produced worse results than d_{rank} in all six datasets.

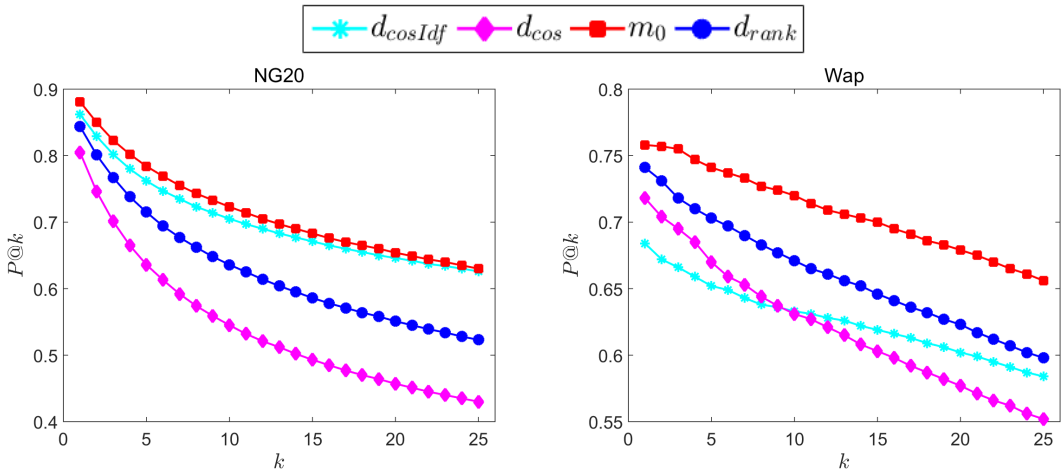


Figure 4.3: Average $P@k$ at $k = 1, 2, \cdots, 25$ in the NG20 and Wap datasets.

self-dissimilarity. As $\eta = \lfloor \log_2 N \rfloor + 1$ is generally larger than the number of distinct frequency values of each term, u_i (the average number of frequency values $\bar{u}_i$ per term in the six text datasets is from 3 to 12), the equal-frequency discetisation creates $u_i < \eta$ frequency intervals for each term. Because a term occurs zero times or only once in many documents and occurs multiple times in a few documents (i.e., zero or small integers are more common frequency values than larger integers), frequency intervals for each term have varying probability mass—higher at small integers (with the highest at zero) and very small in larger integers. For a term, the same frequency value which is rare (i.e, low probability mass) provides more information about the similarity of two documents than the same frequency value which is very common (i.e., high probability mass). Therefore, assigning dissimilarity between documents w.r.t terms with matching frequency values based on the probabilities of the matching frequency values (data-dependent self-dissimilarities) can differentiate documents well and produce better results.

It is interesting to note that d_{cosIdf} produced better retrieval results than d_{cos} only in two (NG20 and Wap) out of six datasets. This result is consistent with the study by Aryal et al. (2015), which shows that IDF term weighting may not always improve task-specific performance, and may be detrimental in some datasets.

The relative average precision ($P@k$) of different measures at each of $k = 1, 2, \cdots, 25$ is generally consistent with the average $MAP@25$ results presented in Table 4.9, except in one dataset where a measure produced better $P@k$ than another measure at some k and worse at others. The $P@k$ for $k = 1, 2, \cdots, 25$ of d_{cos}, d_{cosIdf}, d_{rank} and m_0 in the NG20 and Wap datasets are presented in Figure 4.3. In Wap, d_{cosIdf} was worse than d_{cos} until $k = 9$ and produced better results after $k = 9$. This result shows that it is important to evaluate the average precision of measures over a wide range of k to generalise the effectiveness of measures.

In terms of runtime, all dissimilarity measures had comparable runtimes (they all ran in the same order of magnitude) in all six datasets. The average total runtime for pre-processing and retrieval of all queries in $\mathcal{Q}$ over 10 runs in the six BoW text datasets is provided in Table 4.11. The cosine distance with IDF term weighting (d_{cosIdf}) ran faster

Table 4.11: Average CBIR runtime (seconds) for a query set over 10 runs in BoW text datasets. The presented runtime is the total runtime including pre-processing and retrieval times for all queries in the query set.

	NG20	Ohscal	R52	R8	Fbis	Wap
d_{cos}	34263	21915	5341	1453	92	1070
d_{cosIdf}	25367	4952	3484	781	87	515
d_{rank}	14717	12446	5057	1286	97	113
d_{lin}	17192	15049	3662	1283	100	201
m_1	18955	13396	4006	1251	101	150
m_0	16211	18475	4133	1475	111	130

than the cosine distance without IDF term weighting (d_{cos}) because the IDF-based weights of terms occurring in all documents are zero and these terms can be ignored, which reduces the number of floating-point operations required in the dissimilarity measurement of any two documents.

4.6.4 kNN classification task

In the kNN classification task, we compared the performance of all the contending data-dependent measures with data-independent (distance-based) measures and a supervised distance metric learning method designed specifically for kNN classification called Large Margin Nearest Neighbour (d_{lmnn}) (Weinberger et al., 2006). The method learns a space to project data where instances belonging to the same class become closer to each other (similarity constraints) and instances belonging to different classes are separated further apart (dissimilarity constraints) in the training set. In other words, it projects data into a new space where the kNN classification accuracy of ℓ_2 can be maximised.

We did kNN classification experiments only in the non-BoW datasets where the number of dimensions is low to moderate, because d_{lmnn} is very expensive to learn in high-dimensional BoW text datasets. We used the python implementation of d_{lmnn} by Stewart (2015) available in the GitHub repository[10]. The parameter of the maximum number of iterations in d_{lmnn} was set to 1000.

The average classification errors and standard error of the 12 contending dissimilarity measures over a 10-fold cross-validation are provided in Table 4.12.

Among dissimilarity measures that do not require learning, as in the CBIR results discussed in Section 4.6.3, one-dimensional data-dependent measures (d_{rank}, m_1, d_{lin} and m_0) produced better or competitive classification results than data-independent and tree-based data-dependent measures for eight datasets, except the SatImg and ImgSeg datasets where they produced worse results than the best performing distance measure. In comparison with the distance metric learning-based measure, they produced worse classification results than d_{lmnn} in four out of six datasets where d_{lmnn} could run and produced better classification results in the other two datasets (Blocks and Steel). d_{lmnn} did not complete in datasets with large N and/or M - Gas, Corel, Ismis and Mfeat because of insufficient

[10]https://github.com/michaelstewart/metric-learn

Table 4.12: Average 5NN classification error and standard error (within the parentheses in the second row in small font) over a 10-fold cross-validation in non-BoW datasets. The best result is underlined and the results equivalent (insignificant difference based on two standard errors) to the best result are bold-faced.

	Gas	Ismis	Corel	SatImg	Blocks	Mfeat	Steel	SegImg	Hba	Gtzan
d_{cos}	**0.007**	0.052	0.760	0.149	**0.033**	**0.020**	0.308	0.054	0.498	0.289
	(0.001)	(0.002)	(0.003)	(0.004)	(0.002)	(0.003)	(0.011)	(0.008)	(0.012)	(0.013)
ℓ_2	**0.007**	0.054	0.765	0.093	0.043	**0.019**	0.303	0.053	0.484	0.301
	(0.001)	(0.002)	(0.002)	(0.003)	(0.002)	(0.003)	(0.010)	(0.006)	(0.006)	(0.018)
ℓ_1	**0.005**	0.052	0.686	**0.087**	0.039	**0.020**	0.289	**0.041**	0.423	0.323
	(0.001)	(0.002)	(0.004)	(0.003)	(0.002)	(0.003)	(0.007)	(0.005)	(0.010)	(0.016)
d_{MD}	**0.005**	0.052	0.681	**0.087**	0.039	**0.019**	0.286	**0.041**	0.417	0.326
	(0.001)	(0.002)	(0.003)	(0.003)	(0.002)	(0.003)	(0.006)	(0.005)	(0.014)	(0.016)
d_{mah}	0.014	0.088	0.790	0.273	**0.036**	0.362	0.311	0.083	0.730	0.725
	(0.001)	(0.002)	(0.003)	(0.003)	(0.002)	(0.011)	(0.009)	(0.010)	(0.008)	(0.009)
d_{lmnn}	n/a	n/a	n/a	**0.086**	0.038	n/a	0.307	**0.037**	**0.300**	**0.206**
				(0.003)	(0.003)		(0.006)	(0.004)	(0.009)	(0.014)
m_{IF}	**0.007**	0.055	0.696	0.096	**0.032**	**0.021**	0.295	0.051	0.404	0.324
	(0.001)	(0.002)	(0.002)	(0.003)	(0.002)	(0.002)	(0.004)	(0.004)	(0.009)	(0.010)
d_{USF}	**0.007**	0.059	0.673	**0.091**	**0.033**	**0.021**	0.297	0.048	0.394	0.324
	(0.001)	(0.002)	(0.002)	(0.002)	(0.003)	(0.003)	(0.007)	(0.004)	(0.010)	(0.011)
d_{rank}	**0.006**	**0.046**	**0.613**	0.096	**0.031**	**0.019**	**0.273**	**0.043**	0.354	0.289
	(0.001)	(0.002)	(0.001)	(0.002)	(0.003)	(0.003)	(0.008)	(0.003)	(0.007)	(0.008)
d_{lin}	**0.006**	**0.045**	0.617	0.099	**0.034**	**0.020**	**0.271**	0.060	0.349	0.300
	(0.001)	(0.002)	(0.001)	(0.002)	(0.003)	(0.003)	(0.007)	(0.004)	(0.007)	(0.007)
m_1	**0.006**	**0.046**	**0.612**	0.098	**0.031**	**0.021**	**0.271**	0.049	0.351	0.289
	(0.001)	(0.002)	(0.001)	(0.002)	(0.002)	(0.003)	(0.009)	(0.003)	(0.007)	(0.008)
m_0	**0.006**	**0.046**	**0.613**	0.099	**0.034**	**0.020**	**0.280**	0.056	0.347	0.300
	(0.001)	(0.002)	(0.001)	(0.002)	(0.003)	(0.003)	(0.009)	(0.004)	(0.007)	(0.007)

n/a: result not available because of out of memory error

memory in a machine with 16 GB RAM. This result shows that simple data-dependent measures such as d_{rank}, m_1, d_{lin} and m_0 which do not use any supervised information in the training or pre-processing steps and do not require any learning, can be quite competitive with complex method such as d_{lmnn} which is specifically designed to maximise kNN classification results in some datasets. d_{lmnn} has high time and space complexities because it requires learning and optimisation to find the best projection of data, making it inapplicable to datasets with large N and/or M. It was at least one order of magnitude slower than the simple data-dependent measures, as shown in Table 4.14.

The summarised kNN classification results for 10 non-BoW datasets in terms of win:loss: draw counts of one-dimensional data-dependent measures (d_{rank}, d_{lin}, m_1 and m_0) against the other key contenders based on the two standard errors significance test provided in Table 4.13 show that they had more wins than losses over data-independent (distance) and tree-based data-dependent measures. Although they had more losses than wins over the data-dependent measure that require learning (d_{lmnn}), they were better than or competitive with d_{lmnn} in some datasets.

Table 4.13: Win:loss:draw counts of one-dimensional data-dependent measures against the contenders based on two standard errors in the kNN classification task in non-BoW datasets.

	d_{cos}	ℓ_2	ℓ_1	d_{MD}	m_{IF}	d_{USF}	d_{lmnn}
m_0	5:0:5	5:1:4	3:2:5	4:2:4	5:0:5	5:1:4	1:4:1
m_1	5:0:5	5:0:5	6:1:3	5:1:4	5:0:5	5:1:4	2:4:0
d_{lin}	5:0:5	5:1:4	4:2:4	5:2:3	5:1:4	5:2:3	1:4:1
d_{rank}	5:0:5	6:0:4	6:1:3	5:1:4	6:0:4	5:1:4	2:4:0

Table 4.14: Average 5NN classification runtime (seconds) over a 10-fold cross-validation. The presented runtime is the average total runtime including pre-processing, training and testing time.

	Gas	Ismis	Corel	SatImg	Blocks	Mfeat	Steel	SegImg	Hba	Gtzan
d_{cos}	2782	2526	1514	663	399	36	39	31	38	18
ℓ_2	2013	1761	1059	386	142	36	25	22	24	12
ℓ_1	1066	1151	622	252	125	18	15	14	16	7
d_{MD}	933	829	417	159	79	20	6	10	12	5
d_{mah}	13522	6385	738	247	170	696	10	13	202	43
d_{lmnn}	n/a	n/a	n/a	4088	1368	n/a	440	350	2764	2592
m_{IF}	1116	1338	627	251	118	224	23	19	26	23
d_{USF}	504	468	241	83	56	77	7	5	8	5
d_{rank}	656	508	280	90	35	26	4	6	10	7
d_{lin}	617	537	295	83	74	26	4	6	10	7
m_1	574	540	286	85	31	22	4	7	8	6
m_0	603	551	289	89	69	23	4	6	12	7

n/a: result not available because of out of memory error

4.6.5 Robustness to units and scales of measurement

In order to investigate the robustness of dissimilarity measures to scales and units of measurement, we evaluated their performances after some monotonic transformation of feature values, as done by Fernando and Webb (2017). We used the non-BoW datasets only in this experiment[11].

We employed six different linear and non-linear order-preserving and order-reversing monotonic transformations, as discussed in Fernando and Webb (2017), where each feature value x was transformed using: e^x, e^{-x}, $\frac{1}{x}$, $\log x$, x^2 and $\sqrt{x}$. Because $\frac{1}{x}$ and $\log x$ are not defined for $x = 0$, all transformations were applied on $x'' = b(x + a)$ where $a = 0.0001$ and $b = 100$. A positive value b was used to transform values into a wide range which changes the inter-point distance significantly. Note that the feature values in all dimensions were normalised to a unit range of [0,1] before applying the transformations, in order to ensure the same effect of a and b in all dimensions. Once the feature values were transformed, they were renormalised to be in the unit range. We used exactly the same procedure of monotonic transformation as employed by Fernando and Webb (2017).

[11]The BoW text datasets were not used because there is no issue of scales and units of measurement as feature values are frequency counts.

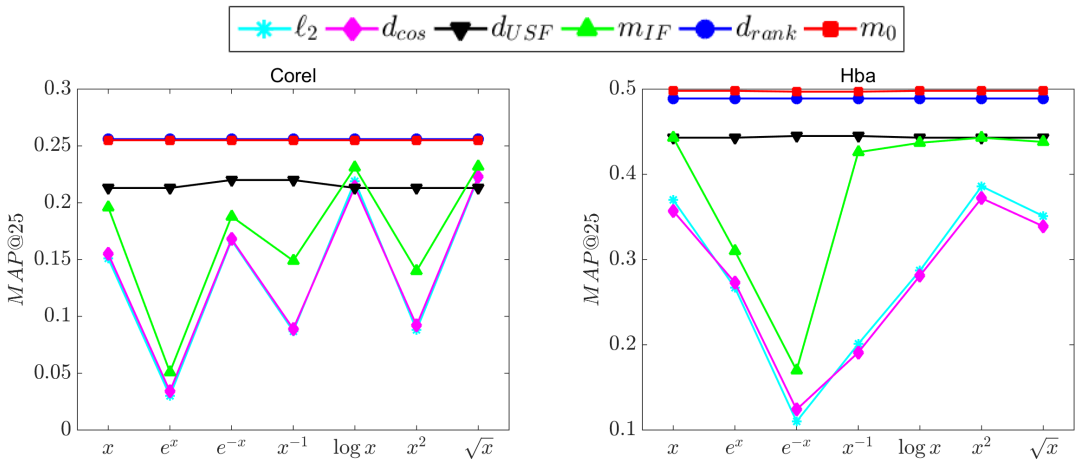


Figure 4.4: Average $MAP@25$ over 10 runs in the Corel and Hba datasets with different monotonic transformation of feature values.

In the CBIR task, we observed that the retrieval results of all four one-dimensional data-dependent measures and d_{USF} remained almost the same with or without monotonic transformations, whereas those of distance-based measures and m_{IF} varied significantly when different monotonic transformations were applied and produced significantly worse results, particularly with e^x and e^{-x}. The information retrieval results in terms of the average $MAP@25$ over 10 runs of ℓ_2, d_{cos}, m_{IF}, d_{USF}, d_{rank} and m_0 in the Corel and Hba datasets are provided in Figure 4.4. The trend was similar in the other datasets. It is interesting to note that ℓ_2, d_{cos} and m_{IF} produced their best retrieval results with the $\sqrt{x}$ transformation in the Corel dataset. This demonstrates that it is important to use the right scale to achieve optimal task-specific performance using distance measures and m_{IF}.

Even the data-dependent measure of m_{IF} was sensitive to monotonic transformations because of the random split on a randomly-selected attribute at each node to build trees. The probability of selecting a split point between any two points is proportional to their distance. When the distribution is skewed, many cut points will be in sparse regions, resulting in many instances in dense regions being in the same leaf, which cannot be differentiated (as in the case of equal-width discretisation discussed in Section 4.5). This is not a problem in d_{rank} or m_0, as the equal frequency discretisation creates intervals such that each interval will have no more than $\varphi = \lceil \frac{N}{\eta} \rceil$ instances even in dense region, unless there are more than φ instances with the same value.

Similar behaviour was observed in the kNN classification task when different transformations were applied. The average 5NN classification errors of ℓ_2, d_{lmnn}, m_{IF}, d_{USF}, d_{rank} and m_0 over a 10-fold cross-validation in the SatImg and Hba datasets are provided in Figure 4.5. Note that d_{lmnn} and ℓ_2 had similar behaviour. They were both sensitive to monotonic transformations and produced worse classification results than data-dependent measures like d_{USF}, d_{rank} and m_0 with some transformations such as e^x and e^{-x}.

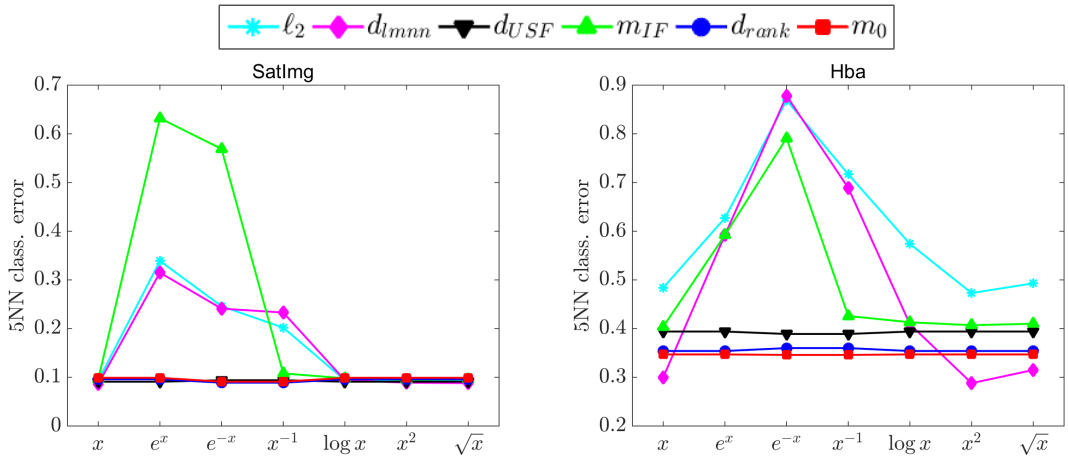


Figure 4.5: Average 5NN classification error over a 10-fold cross-validation in SatImg and Hba datasets with different monotonic transformations of feature values.

4.6.6 Summary of experimental results

Our empirical results in the above three subsections are summarised as follows:

a. In datasets where the dimensionality of data is low to moderate and the distribution is not sparse (i.e., not many instances have the same values) in many dimensions, as in the case of non-BoW datasets, one-dimensional data-dependent measures (d_{rank}, d_{lin}, m_1 and m_0) produce better or equivalent task-specific performances than traditional distance-based (data-independent) and tree-based data-dependent measures.

b. In datasets where the dimensionality of data is high and data distribution is sparse (i.e., many instances have the same value) in many dimensions as in the case of BoW text datasets, one-dimensional data-dependent measures with data-dependent self-dissimilarity (m_1 and m_0, particularly m_0) produced better results than one-dimensional data-dependent measures with data-independent self-dissimilarity (d_{rank} and d_{lin}), and the commonly-used cosine distance with or without IDF term weighting.

c. Simple one-dimensional data-dependent measures (d_{rank}, m_1, d_{lin} and m_0) can produce task-specific results competitive with the complex supervised distance metric learning method of d_{lmnn} in some datasets. As they do not require any optimisation and learning, they run significantly faster than metric learning methods which have high space and time complexities.

d. Even though tree-based data-dependent measures (d_{USF} and m_{IF}) produced better or competitive results than traditional distance-based methods, they produced worse results than one-dimensional data-dependent measures.

e. All one-dimensional data-dependent measures and d_{USF} are robust to units and scales of measurement of feature values, whereas all distance-based measures including d_{lmnn} and the tree-based data-dependent measure of m_{IF} are sensitive to units and scales of measurement.

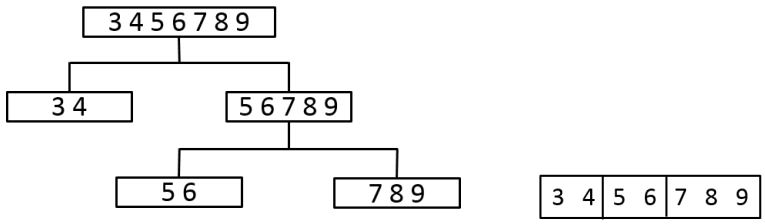
4.7 Discussion

As the magnitudes of feature values are used directly in dissimilarity measurement, distance measures are sensitive to units and scales of measurement. In order to address this issue to some extent, data pre-processing techniques, such as min-max normalisation to ensure feature values in the unit range, and standardisation to ensure unit variance in all dimensions are used to adjust the positions of the data objects in the space. In the case of BoW text datasets, the positions of document vectors are adjusted through IDF term weighting. Distance measures require some form of transformation to produce good results and finding the right transformation is not easy. However, one-dimensional data-dependent measures such as d_{rank}, d_{lin} and m_p are robust to units and scales of measurement.

Although scale-invariant and data-dependent measures of rank difference and Lin's measure were introduced decades ago, they are not used as alternatives to distance measures mainly due to their high computational complexities when one or both instances given for similarity measurement are unseen. This is often the case in data mining, where similar instances of a query/test instance are to be searched in the seen database (Fernando and Webb, 2017). In this paper, we improved the runtimes of data-dependent measures such as d_{rank} to be of the same order as that of distance measures by converting numeric data in continuous domain in each dimension to ordinal domain through discretisation.

Some studies have used an ensemble of random trees to measure similarity between data objects. Torkkola and Tuv (2005) used random forest (Breiman, 2001) and measured similarity between $\mathbf{x}$ and $\mathbf{y}$ as the average shared pathlength in random trees. Shi and Horvath (2006) used the number of shared leaves over a collection of random trees. Aryal et al. (2014a) used isolation forest (Liu et al., 2008) and measured similarity using relative mass. Note that the similarity measure based on relative mass is asymmetric (i.e., the similarity of $\mathbf{x}$ to $\mathbf{y}$ can be different from the similarity of $\mathbf{y}$ to $\mathbf{x}$). Fernando and Webb (2017) used a different implementation of random trees and measured similarity using the number of shared leaves, and Ting et al. (2016) used data mass in isolation trees as the measure of dissimilarity of $\mathbf{x}$ and $\mathbf{y}$. Of all random tree-based measures, m_{IF} is the only fully data-dependent measure where even the self-dissimilarity is data-dependent.

Note that both tree-based (m_{IF} and d_{USF}) and one-dimensional data-dependent measures (d_{rank}, d_{lin} and m_p) estimate the dissimilarity of $\mathbf{x}$ and $\mathbf{y}$ using the regions (leaves or intervals) into which they fall. The only difference is that the former defines regions using a subset of attributes, whereas the latter defines one-dimensional regions through discretisation. There are two limitations of these region-based approaches. Firstly, they lose differences in the feature values of instances in a region. This may not be an issue in most datasets, because these differences are small. Secondly, instances with similar magnitudes or rank differences may appear to be more dissimilar if they happen to fall in two regions. For example, in the case of one-dimensional data as shown in Figure 4.6, 4 is more dissimilar to 5 than 6, simply because the partitioning happens to be between 4 and 5. This problem is even worse in m_{IF} because of the hierarchical partitioning of regions — 4 is more dissimilar to 5 than 9. The expectation in tree-based implementations is that



(a) Tree (b) Discretisation

Figure 4.6: Partition of one-dimensional data to define regions

4 and 5 will occur together in a node much deeper than 5 and 9 would in many trees when multiple trees are created. Hence, tree-based measures require a fairly large number of trees to produce good results, but the runtime increases linearly with the number of trees.

In m_{IF} and d_{USF}, the dissimilarity of $\mathbf{x}$ and $\mathbf{y}$ is estimated using a small subset of attributes in each tree. The number of possible combinations of such attributes subsets increases exponentially with the increase in the number of dimensions. In m_{IF}, even the order in which attributes are selected at each intermediate node to build a tree is important. For a given subset of features, the dissimilarity of $\mathbf{x}$ and $\mathbf{y}$ in a tree can be different if attributes are selected in different orders. Therefore it is necessary to build a large number of trees (more than M) to cover as many attributes subsets as possible. With a large number of trees ($t \geq M$), they become computationally more expensive than one-dimensional data-dependent measures (d_{rank}, d_{lin} and m_p). The CBIR results of d_{USF} and m_{IF} with a varying number of trees in the Corel and Hba datasets are discussed in Appendix 4.A.

The characteristics and effectiveness of tree-based and one-dimensional data-dependent measures depend on the definition of regions. Fernando and Webb (2017) discussed that some implementations of random trees are sensitive to units and scales of measurement. Our empirical results in Section 4.6.5 also show that m_{IF} is sensitive to units, whereas d_{USF} is not. Similarly, in one-dimensional definition of regions, equal-width discretisation (EWD) is sensitive to units, whereas equal-frequency discretisation (EFD) is invariant. Our empirical evaluation reveals that EFD always produced either better or competitive results with EWD. The CBIR results of d_{rank}, d_{lin}, m_1 and m_0 with EFD and EWD in the Corel and Hba datasets are provided in Figure 4.8 in Appendix 4.B.

In many existing dissimilarity measures, the self-dissimilarity of data objects is zero. It is assumed that two data objects are identical if they have the same feature values. Black (1952) argued against this with counter examples and claimed that it is possible to have two distinct objects with the same properties. This can easily happen in data mining, because real-world entities are represented by a fixed number of selected features. Two distinct objects can appear to be identical if they happen to have the same values for all selected features, despite their differences in other unselected features. Unlike existing measures, probability mass-based dissimilarity measures (e.g., m_p and m_{IF}) do not consider two objects with the same values for all selected features to be identical (i.e., zero dissimilarity), they assign the dissimilarity based on the number of objects with the same value in each selected feature.

4.8 Conclusions

In this paper, we have studied the characteristics and relationships of different data-dependent measures. The study has deepened our understanding of these measures in two aspects. First, we extend the one-dimensional data-dependent measure of m_p from $p > 0$ to $p \geq 0$ by introducing m_0-dissimilarity. We show that m_p ($p \geq 0$) is a generic data-dependent measure, where rank distance (d_{rank}) and Lin's measure (d_{lin}) are special cases of m_p with $p > 0$ and $p = 0$, respectively, with a unique difference: m_p has *data-dependent self-dissimilarity* whereas the other two measures have *data-independent self-dissimilarity.*

Second, the empirical evaluations revealed that (a) one-dimensional data-dependent measures (d_{rank}, d_{lin}, m_p) produce more consistent results than commonly-used distance-based measures and tree-based data-dependent measures across different datasets; (b) among one-dimensional data-dependent measures, those with data-dependent self-dissimilarities (i.e., m_p) produce better results than those with data-independent self-dissimilarities (i.e., rank difference and Lin's measure) in datasets where many instances have the same value (i.e., the probability mass is concentrated at a few values) as in the case of BoW text datasets; and (c) unlike traditional distance-based methods, data-dependent measures such as rank difference, Lin's measure and m_p are robust to units and scales of measurement.

To summarise, fully data-dependent similarity (including data-dependent self-similarity) and robustness to units and scales of measurement are two important characteristics of a similarity measure in order to produce good task-specific performance across a wide range of datasets. The fully data-dependent measure of m_p-dissimilarity, which has both of these characteristics, is a more effective similarity measure for data objects than other measures, among all the measures without learning investigated in this study.

Appendix 4.A: Effect of ensemble size in tree-based data-dependent measures

In order to investigate the effect of ensemble size (t) in tree-based data-dependent measures (d_{USF} and m_{IF}), we evaluated their task-specific performance by varying the number of trees. The CBIR performances of d_{USF} and m_{IF} with a number of trees up to $t = 1000$ in the Corel and Hba datasets are shown in Figure 4.7.

As expected, MAP@25 of both measures increased with the increase of t in both datasets. However, they did not produce competitive retrieval results with one-dimension data-dependent measure of m_0 even with $t = 1000$ in both datasets where the number of dimensions (M) is much less than 1000: Corel ($M = 67$) and Hba ($M = 187$). This result shows that tree-based methods require a large ensemble size ($t < M$) to produce a good result, but using a large t makes them expensive to run. For example, average total runtime (building trees, pre-processing and retrieval) of one run in the Corel dataset with $t = 1000$ took 809 seconds in d_{USF} and 2216 seconds in m_{IF}, whereas m_0 took 281 seconds only.

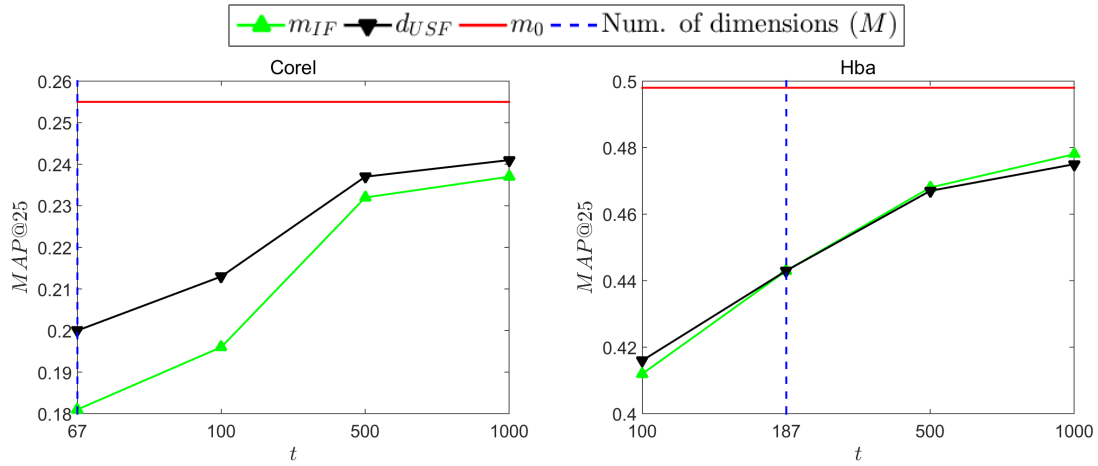


Figure 4.7: Average $MAP@25$ in the Corel ($M = 67$) and Hba ($M = 187$) datasets with different ensemble size.

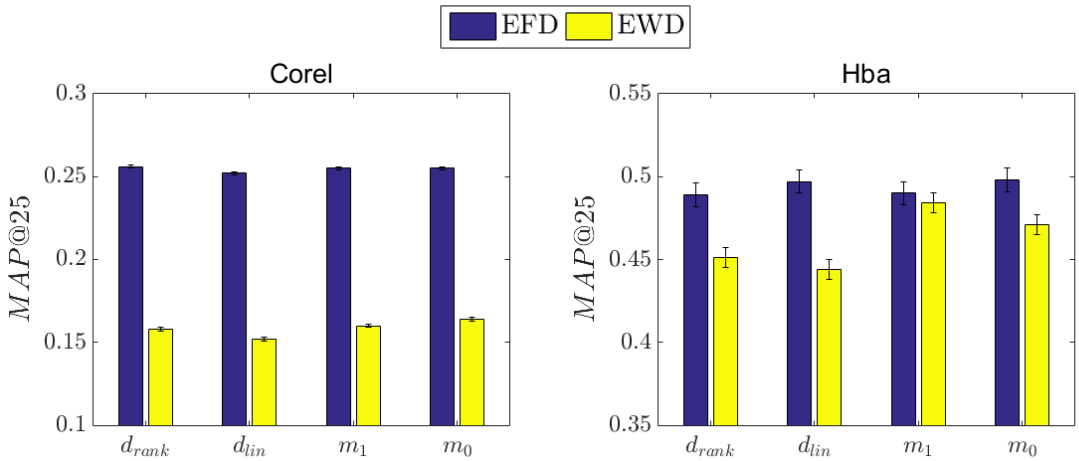


Figure 4.8: Average $MAP@25$ of d_{rank}, d_{lin}, m_1 and m_0 over 10 runs in the Corel and Hba datasets with equal-frequency discretisation (EFD) and equal-width discretisation (EWD).

Appendix 4.B: Effectiveness of equal-frequency and equal-width discretisation approaches to speed up one-dimensional data-dependent measures

We evaluated the performances of one-dimensional data-dependent measures with equal-width discretisation (EWD), and equal-frequency discretisation (EFD) in the CBIR task using the Corel and Hba datasets. We used the same number of intervals $\eta = \lfloor \log_2 N \rfloor + 1$ with both discretisation approaches; therefore, the only difference between them was the discretisation approach. The average $MAP@k$ of d_{rank}, d_{lin}, m_1 and m_0 over 10 runs in the Corel and Hba datasets with EFD and EWD are provided in Figure 4.8.

The CBIR results in Figure 4.8 show that EFD produced either better or at least competitive results with EWD. It did not produce worse retrieval results than EWD in any case.

References

Ariyaratne, H. B. and 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.

Aryal, S., Ting, K. M., Haffari, G. and Washio, T. (2014b). Mp-dissimilarity: A data dependent dissimilarity measure, *In Proceedings of the IEEE International Conference on Data Mining*, IEEE, pp. 707–712.

Aryal, S., Ting, K. M., Haffari, G. and Washio, T. (2015). Beyond tf-idf and cosine distance in document dissimilarity measures, *In Proceedings of the 11th Asia Information Retrieval Societies Conference*, Springer, Cham, pp. 400–406.

Aryal, S., Ting, K. M., Washio, T. and Haffari, G. (2017). Data-dependent dissimilarity measure: an effective alternative to geometric distance measures, *Knowledge and Information Systems* pp. 1–28, doi:10.1007/s10115–017–1046–0.

Aryal, S., Ting, K. M., Wells, J. R. and Washio, T. (2014a). Improving iForest with Relative Mass, *In Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, Cham, pp. 510–521.

Bache, K. and Lichman, M. (2013). UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences.
URL: *http://archive.ics.uci.edu/ml*

Beyer, K. S., Goldstein, J., Ramakrishnan, R. and Shaft, U. (1999). When is "nearest neighbour" meaningful?, *In Proceedings of the 7th International Conference on Database Theory*, Springer-Verlag, London, UK, pp. 217–235.

Black, M. (1952). The identity of indiscernibles, *MIND: A Quarterly Review of Psychology and Philosophy* **61**(242): 153–164.

Breiman, L. (2001). Random forests, *Machine Learning* **45**(1): 5–32.

Cardoso-Cachopo, A. (2007). *Improving Methods for Single-label Text Categorization*, PhD thesis, Instituto Superior Tecnico, Technical University of Lisbon, Lisbon, Portugal.

Conover, W. J. and Iman, R. L. (1981). Rank transformations as a bridge between parametric and nonparametric statistics, *The American Statistician* **35**(3): 124–129.

Deza, M. M. and Deza, E. (2009). *Encyclopedia of Distances*, Springer, Berlin Heidelberg.

Fernando, T. L. and Webb, G. I. (2017). SimUSF: an efficient and effective similarity measure that is invariant to violations of the interval scale assumption, *Data Mining and Knowledge Discovery* **31**(1): 264–286.

François, D., Wertz, V. and Verleysen, M. (2007). The concentration of fractional distances, *IEEE Transactions on Knowledge and Data Engineering* **19**(7): 873–886.

Han, E.-H. and 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-Verlag, London, UK, pp. 424–431.

Krumhansl, C. L. (1978). Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density, *Psychological Review* **85**(5): 445–463.

Kulis, B. (2013). Metric learning: A survey, *Foundations and Trends in Machine Learning* **5**(4): 287–364.

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, USA, pp. 296–304.

Liu, F., Ting, K. M. and Zhou, Z.-H. (2008). Isolation forest, *In Proceedings of the Eighth IEEE International Conference on Data Mining*, pp. 413–422.

Mahalanobis, P. C. (1936). On the generalized distance in statistics, *In Proceedings of the National Institute of Sciences of India*, Vol. 2, pp. 49–55.

Mansouri, J. and Khademi, M. (2015). Multiplicative distance: a method to alleviate distance instability for high-dimensional data, *Knowledge and Information Systems* **45**(3): 783–805.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* **12**: 2825–2830.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval, *Information Processing and Management* **24**(5): 513–523.

Salton, G. and McGill, M. J. (1986). *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, USA.

Shi, T. and Horvath, S. (2006). Unsupervised learning with random forest predictors, *Journal of Computational and Graphical Statistics* **15**(1): 118–138.

Stevens, S. S. (1946). On the theory of scales of measurement, *Science* **103**(2684): 677–680.

Stewart, M. (2015). Metric learning algorithms in Python. GitHub repository.
URL: *https://github.com/michaelstewart/metric-learn*

Sturges, H. A. (1926). The choice of a class interval, *Journal of the American Statistical Association* **21**(153): 65–66.

Ting, K. M., Zhu, Y., Carman, M., Zhu, Y. and Zhou, Z.-H. (2016). Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure, *In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1205–1214.

Torkkola, K. and Tuv, E. (2005). Ensemble learning with supervised kernels, *In Proceedings of the 16th European Conference on Machine Learning*, Springer-Verlag, Berlin, Heidelberg, pp. 400–411.

Tversky, A. (1977). Features of Similarity, *Psychological Review* **84**(2): 327–352.

Wang, F. and Sun, J. (2015). Survey on distance metric learning and dimensionality reduction in data mining, *Data Mining and Knowledge Discovery* **29**(2): 534–564.

Weinberger, K., Blitzer, J. and Saul, L. (2006). Distance metric learning for large margin nearest neighbour classification, *In Proceedings of the 18th International Conference on Neural Information Processing Systems*, MIT Press, pp. 1473–1480.

Zhou, G.-T., Ting, K. M., Liu, F. T. and Yin, Y. (2012). Relevance feature mapping for content-based multimedia information retrieval, *Pattern Recognition* **45**(4): 1707–1720.

Chapter 5

Inter-document similarity measurement in the bag-of-words vector space model

To explain the superior performance of m_0-dissimilarity over the most widely-used cosine measure in BoW text datasets in the last chapter, this chapter investigates the issues of existing BoW document similarity measures more closely. It discusses the shortcomings of the underlying assumptions of term-weighting schemes employed in existing BoW document similarity measures and provides an alternative assumption, which is more congruous with the requirements of inter-document similarity measurement. Based on the new assumption, it introduces a new simple but effective BoW inter-document similarity measure called Sp, where the explicit adjustment of document vectors through term weighting is not required and evaluates the performance of Sp with that of existing BoW document similarity measures using different term-weighting schemes. Sp is a simplified version of m_0-dissimilarity in BoW document similarity measurement.

This work on inter-document similarity measurement in the BoW vector space model has been reported in the following papers:

Aryal, S., Ting, K. M., Washio, T. and Haffari, G. (2015), Beyond tf-idf and cosine distance in document dissimilarity measures, In *Proceedings of the 11th Asia Information Retrieval Societies Conference (AIRS) 2015*, Springer Cham, pp. 400-406.

Aryal, S., Ting, K. M., Washio, T. and Haffari, G. (2017), A new simple and effective measure for inter-document similarity measurement, *Computational Intelligence* (under review).

The journal paper is an extended version of the conference paper. This chapter is a copy of the paper submitted to the journal and a copy of the conference paper is attached in Appendix B. In order to generate consistent presentation within the thesis, the format and some notations or symbols used have been changed, minor grammar and spelling mistakes have been corrected, and sections of the submitted paper have been renumbered.

A new simple and effective measure for inter-document similarity measurement

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

[1]Federation University, Australia
[2]Monash University, Australia
[3]Osaka University, Japan

Abstract:

To measure the similarity of two documents in the bag-of-words (BoW) vector representation, different term-weighting schemes are used to improve the performance of cosine similarity—the most widely-used inter-document similarity measure in text mining. In this paper, we identify the shortcomings of the underlying assumptions of term weighting in the inter-document similarity measurement task, and provide a more fit-for-purpose assumption. Based on this new assumption, we introduce a new simple but effective similarity measure which does not require explicit term weighting. The proposed measure employs a more nuanced probabilistic approach than those used in term weighting to measure the similarity of two documents w.r.t each term occurring in the two documents. Our empirical comparison with the existing similarity measures using different term-weighting schemes shows that the new measure produces (i) better results in the binary BoW representation; and (ii) competitive and more consistent results in the term-frequency-based BoW representation.

Keywords: Inter-document similarity, tf-idf term weighting, cosine similarity, BM25, weighted Jaccard, Sp

5.1 Introduction

Pairwise similarity measurements of documents is a fundamental task in many text-mining problems such as query-by-example, document classification and clustering.

In the bag-of-words (BoW) (Salton and McGill, 1986; Manning et al., 2008) vector space model, a document $\mathbf{x}$ is represented by an M-dimensional vector where M is the number of terms in a given dictionary, i.e., $\mathbf{x} = \langle x_1, x_2, \cdots, x_M \rangle$; and it has the following two representations:

1. Term-frequency-based representation: each $x_i \in \mathbb{Z}_+$ ($\mathbb{Z}_+$ is a set of non-negative integers) is the occurrence frequency of term t_i in document $\mathbf{x}$.
2. Binary representation: each $x_i \in \{0, 1\}$ where 0 represents the absence of term t_i in document $\mathbf{x}$ and 1 represents the presence of t_i in $\mathbf{x}$.

Because the number of terms in a document is significantly less than that in the dictionary, every document is represented as a sparse BoW vector, where many entries are zero. Because of sparsity, Euclidean distance is not a good similarity measure and the angular distance, also known as cosine distance, is the preferred choice of inter-document similarity measures (Salton and McGill, 1986; Salton and Buckley, 1988).

Because all terms in a document are not equally important to represent its subject, different term-weighting schemes (Manning et al., 2008; Salton and Buckley, 1988) are used to adjust vector components based on the importance of their terms.

The idea of term weighting was first introduced in the field of information retrieval (IR), where the task is to measure the relevance of documents in a given collection D for a given query phrase consisting of a few terms. It is based on the following two assumptions (Manning et al., 2008; Salton and Buckley, 1988; Zobel and Moffat, 1998):

i. A term is important in a document if it occurs multiple times in the document.
ii. A rare term that occurs in a few documents in the collection is more important than frequent terms that occur in many documents in the collection.

The importance of terms in a document is estimated independent of the query. Because a query in the IR task is short and each term generally occurs only once, it is not an issue that the weights are determined independent of the query.

However, it can be counter-productive in the query-by-example task, where the query itself is a document, and terms often occur more than once in the query document. For example, to a query document $\mathbf{q}$, a document $\mathbf{x}$ having more occurrences of the terms in $\mathbf{q}$ may not be more similar than $\mathbf{y}$ which has exactly the same occurrences of terms in $\mathbf{q}$.

Previous research in the BoW inter-document similarity measurement task has focused on developing effective term-weighting schemes to improve the task-specific performance of existing measures, such as cosine and Best Match 25 (BM25) (Salton and Buckley, 1988; Robertson et al., 1994; Joachims, 1997; Singhal, 1997; Roberston and Zaragoza, 2009; Paltoglou and Thelwall, 2010; Han et al., 2012; Wang and Zhang, 2013). In contrast, we investigate an alternative similarity measure where an adjustment of vector components using term weighting is not required.

This paper makes the following contributions:

1. It identifies the shortcomings of the underlying assumptions of term-weighting schemes employed in existing measures, and provides an alternative assumption which is more congruous with the requirements of inter-document similarity measurement.

Table 5.1: Key notations

D	A collection of N documents (i.e., $\|D\| = N$)
$\mathbf{x}$	BoW vector of a document $\langle x_1, x_2, \cdots, x_M \rangle$
t_i	The ith term in the dictionary
n_i	The number of documents in D having t_i
$T_{\mathbf{x}}$	The set of terms in $\mathbf{x}$
$w_i(\mathbf{x})$	The importance or weight of t_i in $\mathbf{x}$
$tf_i(\mathbf{x})$	Term frequency factor of t_i in $\mathbf{x}$
$idf(t_i)$	Inverse document frequency factor of t_i
$s(\mathbf{x}, \mathbf{y})$	The similarity of two documents $\mathbf{x}$ and $\mathbf{y}$
$dl(\mathbf{x})$	The length of document $\mathbf{x}$ (i.e., $\sum_{i=1}^{M} x_i$)
$avgdl$	The average length of documents in D
$\mathbf{x} \underset{\mathbf{q}\{i\}}{\succ} \mathbf{y}$	$\mathbf{x}$ is more similar to $\mathbf{q}$ than $\mathbf{y}$ w.r.t $t_i \in T_{\mathbf{q}}$
$\mathbf{x} \underset{\mathbf{q}\{i\}}{=} \mathbf{y}$	$\mathbf{x}$ is equally similar to $\mathbf{q}$ as $\mathbf{y}$ w.r.t $t_i \in T_{\mathbf{q}}$

2. It introduces a new simple but effective inter-document similarity measure which is based on the new assumption and does not require explicit term weighting. It uses a more nuanced probabilistic approach than those used in term weighting to measure the similarity of two documents w.r.t each term occurring in the two documents under measurement.
3. It compares the performance of the new measure with existing measures (which use different term-weighting schemes) in the query-by-example task. Our results reveal that the new measure produces (i) better results than existing measures in the binary BoW representation; and (ii) results competitive with and more consistent than existing measures in term-frequency-based BoW representation.

The rest of the paper is organised as follows. Related work in the areas of term weighting and inter-document similarity measures is discussed in Section 5.2. Issues of term weighting in inter-document similarity measurement are discussed in Section 5.3. The proposed new inter-document similarity measure is presented in Section 5.4, followed by empirical results in Section 5.5, related discussion in Section 5.6, and the last section presents the conclusions.

The key notations used in this paper are defined in Table 5.1.

5.2 Related work

In this section, we discuss term weighting and some widely-used existing BoW inter-document similarity measures.

5.2.1 Term weighting

In the field of IR, there has been considerable research on effective term-weighting schemes. The importance of a term t_i in document $\mathbf{x}$, $w_i(\mathbf{x})$, is estimated using different variants and combinations of two factors (Manning et al., 2008; Salton and Buckley, 1988; Joachims,

1997; Robertson et al., 1994; Singhal, 1997; Roberston and Zaragoza, 2009; Paltoglou and Thelwall, 2010; Han et al., 2012; Wang and Zhang, 2013): (i) a document-based factor based on the frequency of t_i in $\mathbf{x}$, x_i; and (ii) a collection-based factor based on the number of documents where t_i occurs, n_i.

The most widely-used term-weighting scheme is term frequency - inverse document frequency (tf-idf) where $w_i(\mathbf{x}) = tf_i(\mathbf{x}) \times idf(t_i)$ (Manning et al., 2008; Salton and Buckley, 1988); and it includes:

i. Document-based factor: $tf_i(\mathbf{x}) = 1 + \log(x_i)$ if $x_i > 0$, and 0 otherwise;

ii. Collection-based factor: $idf(t_i) = \log\left(\frac{N}{n_i}\right)$.

In the IR task, the idea of tf-idf term weighting is based on the following assumptions (Zobel and Moffat, 1998):

i. Documents with multiple occurrences of query terms are more relevant than documents with a single occurrence of query terms [the tf assumption].

ii. Documents with rare query terms occurring in a few documents in the collection are more relevant to the query than documents with frequent query terms occurring in many documents in the collection [the idf assumption].

The tf factor considers the importance of t_i in a document. Even though a document with multiple occurrences of a query term is more likely to be relevant to the given query, a document with greater occurrences of one query term is not necessarily more relevant than a document with fewer occurrences of two query terms. Therefore, the logarithmic scaling of raw term frequencies is used to reduce the over-influence of high frequencies of query terms (Manning et al., 2008; Salton and Buckley, 1988).

The idf factor considers the importance of t_i in the given collection. Basically, it ranks the importance of terms in the given dictionary based on the number of documents where they occur. Terms occurring in only a few documents (i.e., rare terms) are considered to be more important in documents, and they are given more weight than the terms occurring in many documents (i.e., frequent terms) (Manning et al., 2008; Salton and Buckley, 1988).

5.2.2 Inter-document similarity measures

Here, we discuss three commonly-used measures to estimate the similarity of two document vectors $\mathbf{x}$ and $\mathbf{y}$, $s(\mathbf{x}, \mathbf{y}) \rightarrow \mathbb{R}$ where $\mathbb{R}$ is a real domain.

Cosine similarity

The cosine similarity measure with tf-idf term weighting is the most commonly-used inter-document similarity measure. Using term-weighted vectors, the cosine similarity of two documents $\mathbf{x}$ and $\mathbf{y}$ is estimated as:

$$s_{cos}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{M} w_i(\mathbf{x}) \times w_i(\mathbf{y})}{\sqrt{\sum_{i=1}^{M} w_i(\mathbf{x})^2} \times \sqrt{\sum_{i=1}^{M} w_i(\mathbf{y})^2}} \tag{5.1}$$

Note that the two terms in the denominator of Eqn 5.1 are the Euclidean lengths (ℓ_2-norms) of the term-weighted vectors.

It is important to normalise the similarity of documents by their lengths, otherwise cosine similarity favours longer documents which have higher probability of having more terms in common with the query document over shorter documents (Salton and McGill, 1986; Manning et al., 2008; Salton and Buckley, 1988; Singhal et al., 1996).

Best Match 25 (BM25)

BM25 (Roberston and Zaragoza, 2009; Jones et al., 2000) is a state-of-the-art document ranking measure in IR. It is based on the probabilistic framework of term weighting by Robertson et al. (1994). Han et al. (2012) used BM25 to measure the similarity of two documents $\mathbf{x}$ and $\mathbf{y}$ as follows:

$$s_{bm25}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{M} idf_{bm25}(t_i) \times \frac{x_i \cdot (a+1)}{x_i + a \cdot \left(1 - b + b \cdot \frac{dl(\mathbf{x})}{avgdl}\right)} \times \frac{y_i \cdot (a+1)}{y_i + a \cdot \left(1 - b + b \cdot \frac{dl(\mathbf{y})}{avgdl}\right)} \tag{5.2}$$

where $dl(\mathbf{x}) = \sum_{i=1}^{M} x_i$ is the normal length of document $\mathbf{x}$ (i.e., ℓ_1-norm of the unweighted vector), $avgdl = \frac{1}{N}\sum_{\mathbf{x}\in D} dl(\mathbf{x})$ is the average normal document length, a and b are free parameters that control the influence of the term frequencies and document lengths, and $idf_{bm25}(t_i)$ is the idf factor of term t_i defined as follows:

$$idf_{bm25}(t_i) = \log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right) \tag{5.3}$$

BM25 uses different variants of tf and idf factors in the similarity measure. The pivoted normal document length (Singhal et al., 1996) is used in the tf factor so that longer documents which have higher probability of having more terms in common with the query document are not favoured over shorter documents.

Jaccard similarity

The Jaccard similarity (Jaccard, 1901) of two documents $\mathbf{x}$ and $\mathbf{y}$ is estimated as follows:

$$s_{jac}(\mathbf{x},\mathbf{y}) = \frac{|T_{\mathbf{x}} \cap T_{\mathbf{y}}|}{|T_{\mathbf{x}} \cup T_{\mathbf{y}}|} \tag{5.4}$$

where $T_{\mathbf{x}} = \{t_i : x_i > 0\}$ is the set of terms in document $\mathbf{x}$ and $|\cdot|$ is the cardinality of a set.

Jaccard similarity only considers the number of terms occurring in both $\mathbf{x}$ and $\mathbf{y}$ and does not take into account the importance of terms in documents. The similarity is normalised by the number of distinct terms occurring in either $\mathbf{x}$ or $\mathbf{y}$ to take into account that $\mathbf{x}$ and $\mathbf{y}$ have higher chance of having terms in common if they have more terms.

The weighted or generalised version of Jaccard similarity (Chierichetti et al., 2010) of two documents using term-weighted vectors is defined as follows:

$$s_{wjac}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{M} \min\{w_i(\mathbf{x}), w_i(\mathbf{y})\}}{\sum_{i=1}^{M} \max\{w_i(\mathbf{x}), w_i(\mathbf{y})\}} \tag{5.5}$$

The similarity of $\mathbf{x}$ and $\mathbf{y}$ w.r.t $t_i \in T_{\mathbf{x}} \cap T_{\mathbf{y}}$ depends on the importance of t_i in the two documents. The similarity is normalised by the sum of maximum weights of all $t_i \in T_{\mathbf{x}} \cup T_{\mathbf{y}}$.

Note that the weighted Jaccard similarity of $\mathbf{x}$ and $\mathbf{y}$ (Eqn 5.5) in the binary BoW vector representation without any term weighting is equivalent to the traditional Jaccard similarity (Eqn 5.4).

5.3 Issues of the tf-idf assumptions in inter-document similarity measurement

Even though the tf and idf assumptions discussed in Section 5.2.1 are intuitive in the IR task to rank documents for a given query phrase of a few terms, they can be counter-intuitive in the query-by-example task, which requires inter-document similarity measurements to rank documents in D w.r.t a given query document.

In the literature, the query-by-example task is treated as an IR task, where query is a document, and the same idea of the tf-idf term weighting is used. However, there is a fundamental difference between the two tasks. Unlike in the typical IR task where the query comprises a few distinct terms (i.e., each term generally occurs only once in the query phrase), the query in the query-by-example task is a long document which often has multiple occurrences of terms.

5.3.1 Issue of the tf assumption

For a query document $\mathbf{q}$ with terms $T_{\mathbf{q}}$, a document $\mathbf{x}$ having more occurrences of terms in $T_{\mathbf{q}}$ than in $\mathbf{q}$, may not be more similar to $\mathbf{q}$ than another document $\mathbf{y}$, which has similar occurrences of terms in $T_{\mathbf{q}}$ as in $\mathbf{q}$. For example, assume $\mathbf{x}$ and $\mathbf{y}$ have frequencies of $t_r \in T_{\mathbf{q}}$ as $x_r = 10$ and $y_r = 1$, respectively. If $\mathbf{q}$ has $q_r = 1$, it is difficult to say that $\mathbf{x}$ is more similar to $\mathbf{q}$ than $\mathbf{y}$ w.r.t t_r $\left(i.e., \mathbf{x} \underset{\mathbf{q}\{\mathrm{r}\}}{\succ} \mathbf{y}\right)$, simply because of $x_r > q_r$ (and $q_r = y_r$). It might be the case that $\mathbf{y}$ is exactly the same document as $\mathbf{q}$.

Because of the tf-based term-weighting factor, $\mathbf{x} \neq \mathbf{q}$ can be more similar to $\mathbf{q}$ than $\mathbf{q}$ itself using some existing measure such as BM25[1]. Therefore, the tf assumption can be counter-intuitive in inter-document similarity measurement.

5.3.2 Issue of the idf assumption

Similarly, $\mathbf{x}$ having rare terms of $T_{\mathbf{q}}$ may not be more similar to $\mathbf{q}$ than $\mathbf{y}$ having frequent terms of $T_{\mathbf{q}}$. For example, consider the scenario presented in Table 5.2:

Because $idf(t_h) = 0$, the term t_h will be completely ignored. However, $q_h = y_h = 10$ is more useful than $q_g = x_g = y_g = 1$ because $\mathbf{y}$ is the only document in D which has as many occurrences of t_h as $\mathbf{q}$. Even though there is no discrimination between documents

[1] It depends on the lengths of documents and parameters a and b.

Table 5.2: A scenario to demonstrate the issue of the idf assumption. Note that all $\frac{N}{2}$ documents having t_g have a frequency of 1; and all N documents having t_h have a frequency of 1 except $\mathbf{y}$ where $y_h = 10$

	n	$idf(t)$	$\mathbf{x}$	$\mathbf{y}$	$\mathbf{q}$
t_g	$\frac{N}{2}$	$\log(2)$	1	1	1
t_h	N	0	1	10	10

w.r.t t_g (all $\frac{N}{2}$ documents with t_g have a frequency of 1), t_g is assigned more weight with $idf(t_g) = \log 2$ than t_h with $idf(t_h) = 0$. As a result, $\mathbf{x}$ and $\mathbf{y}$ become equally similar to $\mathbf{q}$ w.r.t t_g and t_h $\left(i.e., \mathbf{x} \underset{\mathbf{q}\{g,h\}}{=} \mathbf{y}\right)$ even though $\mathbf{y}$ has exactly the same occurrences of t_g and t_h as $\mathbf{q}$. This example shows that the idf assumption can be counter-intuitive in document similarity measurements.

5.4 Our proposal to overcome the issues of tf-idf based term weighting in inter-document similarity measurement

The main problem of the tf-idf term weighting in inter-document similarity measurement is that the importance of t_i in $\mathbf{x}$, $w_i(\mathbf{x})$, is estimated without considering the frequency of t_i in $\mathbf{q}$, q_i. This is not an issue in the IR task because q_i is almost always 1 if t_i occurs in the given query phrase $\mathbf{q}$. In a query document, q_i can be larger than 1. Therefore, judging the importance of t_i in $\mathbf{x}$, without considering q_i can be counter-productive in inter-document similarity measurement.

A more fit-for-purpose approach would be to evaluate the importance of t_i in $\mathbf{x}$ by examining the similarity of x_i w.r.t. q_i. However, as discussed in Section 5.3.2, simply having similar occurrences of t_i (i.e., $x_i = q_i$) is not sufficient to consider them to be similar. The similarity measure should also consider how rare the frequency of t_i is in the collection.

Putting the above requirements together, the similarity of $\mathbf{x}$ and $\mathbf{q}$ w.r.t t_i should be based on the number of documents in D which have similar occurrence frequencies of t_i as in both $\mathbf{x}$ and $\mathbf{q}$. More formally, $\mathbf{x}$ and $\mathbf{q}$ are more likely to be similar w.r.t t_i if $|\{\mathbf{z} \in D : \min(x_i, q_i) \leq z_i \leq \max(x_i, q_i)\}|$ is small. The first part in Table 5.3 compares the underlying assumptions of the tf-idf term weighting (used in existing measures) and the proposed approach called Sp, to be introduced in the next subsection.

This approach addresses the limitations of both the tf and idf assumptions discussed in Section 5.3. The results of the new approach using the same examples discussed in Sections 5.3.1 and 5.3.2 are provided in the second part of Table 5.3. The comparisons demonstrate that the new approach provides more intuitive outcomes than the tf-idf term weighting.

Table 5.3: The tf-idf weighting (in existing measures) versus Sp: (i) Underlying assumptions for documents to be relevant/similar to a query document $\mathbf{q}$; and the relation of similarities of $\mathbf{x}$ and $\mathbf{y}$ to $\mathbf{q}$ (ii) in the same example discussed in Section 5.3.1 and (iii) in the same example used in Section 5.3.2.

tf-idf term weighting	Sp (the proposed approach)
(i) Underlying Assumptions	
tf: $\mathbf{y} \underset{\mathbf{q}\{\mathrm{i}\}}{\succ} \mathbf{x}$ if $y_i > x_i$ idf: $\mathbf{y} \underset{\mathbf{q}\{\mathrm{i,j}\}}{\succ} \mathbf{x}$ if $n_i < n_j$ (for $q_i > 0, q_j > 0$; $x_i = 0, x_j = q_j$; $y_i = q_i, y_j = 0$)	$\mathbf{y} \underset{\mathbf{q}\{\mathrm{i}\}}{\succ} \mathbf{x}$ if $\lvert\{\mathbf{z} \in D : \alpha(y_i, q_i) \leq z_i \leq \beta(y_i, q_i)\}\rvert < \lvert\{\mathbf{w} \in D : \alpha(x_i, q_i) \leq w_i \leq \beta(x_i, q_i)\}\rvert$ where $\alpha(\cdot,\cdot) = \min(\cdot,\cdot); \beta(\cdot,\cdot) = \max(\cdot,\cdot)$
(ii) Example discussed in Section 5.3.1 ($x_r = 10, y_r = 1, q_r = 1$)	
$\mathbf{x} \underset{\mathbf{q}\{\mathrm{r}\}}{\succ} \mathbf{y}$ because $x_r > y_r$ (even though $y_r = q_r$)	$\mathbf{y} \underset{\mathbf{q}\{\mathrm{r}\}}{\succ} \mathbf{x}$ because $\lvert\{\mathbf{z} \in D : q_r = z_r = y_r\}\rvert < \lvert\{\mathbf{w} \in D : q_r \leq w_r \leq x_r\}\rvert$
(iii) Example discussed in Section 5.3.2 (Table 5.2)	
$\mathbf{x} \underset{\mathbf{q}\{\mathrm{g,h}\}}{=} \mathbf{y}$ because (i) $q_g = x_g = y_g = 1$; and (ii) $idf(t_h) = 0$ (even though $q_h = y_h = 10$ and $x_h = 1$)	$\mathbf{y} \underset{\mathbf{q}\{\mathrm{g,h}\}}{\succ} \mathbf{x}$ because (i) $\lvert\{\mathbf{z} \in D : q_g = z_g = y_g\}\rvert = \lvert\{\mathbf{w} \in D : q_g = w_g = x_g\}\rvert$ (ii) $\lvert\{\mathbf{z} \in D : q_h = z_h = y_h\}\rvert < \lvert\{\mathbf{w} \in D : x_h \leq w_h \leq q_h\}\rvert$

5.4.1 Sp: A new document similarity measure

Recently, Aryal et al. (2014b, 2017) introduced a data-dependent measure where the similarity of two data objects u and v depends on the distribution of data between u and v (Aryal et al., 2014b, 2017). The intuition is that u and v are more likely to be similar if there are less data between them. For example, two individuals earning 800k and 900k are judged to be more similar by humans than two individuals earning 50k and 150k, because many more people earn in [50k, 150k] than [800k, 900k].

Using the similar idea, the similarity of two documents $\mathbf{x}$ and $\mathbf{y}$ can be estimated as:

$$s_{sp}(\mathbf{x}, \mathbf{y}) = \frac{1}{|T_{\mathbf{x}} \cup T_{\mathbf{y}}|} \sum_{t_i \in T_{\mathbf{x}} \cap T_{\mathbf{y}}} \log \frac{N}{|\{\mathbf{z} \in D : \min(x_i, y_i) \leq z_i \leq \max(x_i, y_i)\}|} \tag{5.6}$$

where $\frac{1}{|T_{\mathbf{x}} \cup T_{\mathbf{y}}|}$ is a normalisation term to account for the probability of a term occurring in both $\mathbf{x}$ and $\mathbf{y}$. The normalisation term reduces the bias towards documents having more terms because they have a higher probability of having terms in a query document than documents with fewer terms.

The number of distinct terms is used as a normalisation factor (as in the traditional Jaccard similarity) because it is not sensitive to multiple occurrences of the terms in a document which do not occur in the query document. In the IR task, Singhal et al. (1996) have shown that it is more effective than the cosine or normal length normalisation which penalise documents with multiple occurrences of the terms which are not in the query phrase.

Sp can be interpreted as a simple probabilistic measure where the similarity of two documents w.r.t $t_i \in T_{\mathbf{x}} \cap T_{\mathbf{y}}$ is assigned based on the probability of the frequency of t_i to be in $[\min(x_i, y_i), \max(x_i, y_i)]$, $P(\min(x_i, y_i) \leq \chi_i \leq \max(x_i, y_i))$ (where χ_i is a random variable representing the occurrence frequency of term t_i in a document). In practice,

$P(\min(x_i, y_i) \leq \chi_i \leq \max(x_i, y_i)) = \frac{|\{\mathbf{z} \in D \,:\, \min(x_i,y_i) \leq z_i \leq \max(x_i,y_i)\}|}{N}$, which is the inverse of the term used in Eqn 5.6.

5.4.2 Characteristics of Sp

The proposed measure has the following characteristics:

i) *Term weighting is not required*:

Unlike in existing measures such as cosine and BM25, x_i and y_i are not used directly in the similarity measure. They are used simply to define $\min(x_i, y_i)$ and $\max(x_i, y_i)$, and the similarity is based on $|\{\mathbf{z} \in D : \min(x_i, y_i) \leq z_i \leq \max(x_i, y_i)\}|$, which is invariant to the monotonic scaling of frequency values. Hence, Sp does not require additional term weighting to adjust frequency values.

ii) *Self-similarity is data-dependent and the upper bound of similarity*:

Unlike cosine and both variants of Jaccard similarity, where the self-similarity of documents is fixed with the maximum of 1, Sp has data-dependent self-similarity because $s_{sp}(\mathbf{x}, \mathbf{x})$ depends on the $P(x_i)$ for all $t_i \in T_{\mathbf{x}}$. Therefore, $s_{sp}(\mathbf{x}, \mathbf{x})$ and $s_{sp}(\mathbf{y}, \mathbf{y})$ can be different.

The similarity in Sp is bounded by its self-similarity i.e., $\forall_{\mathbf{y} \neq \mathbf{x}}\ s_{sp}(\mathbf{x}, \mathbf{x}) > s_{sp}(\mathbf{x}, \mathbf{y})$. Although BM25 also has data-dependent self-similarity, it is possible to have similarity of different documents larger than the self-similarity, i.e., there may be $\mathbf{y} \neq \mathbf{x}$ with $s_{bm25}(\mathbf{x}, \mathbf{y}) > s_{bm25}(\mathbf{x}, \mathbf{x})$[2].

iii) *Relationship with the traditional Jaccard similarity and idf term weighting*:

The formulation of Sp (Eqn 5.6) looks similar to the formulation of the traditional Jaccard similarity (Eqn 5.4), except that the similarity of $\mathbf{x}$ and $\mathbf{y}$ w.r.t $t_i \in T_{\mathbf{x}} \cap T_{\mathbf{y}}$ is based on $|\{\mathbf{z} \in D : \min(x_i, y_i) \leq z_i \leq \max(x_i, y_i)\}|$ in Sp, whereas it is the fixed constant of 1 in the traditional Jaccard similarity.

In the binary BoW vector representation, when $t_i \in T_{\mathbf{x}} \cap T_{\mathbf{y}}$ and $|\{\mathbf{z} \in D : z_i = 1\}| = n_i$, Sp assigns the similarity of $\mathbf{x}$ and $\mathbf{y}$ w.r.t t_i based on $idf(t_i)$, whereas in the traditional Jaccard similarity, it is 1, irrespective of whether t_i is rare or frequent in D .

In the term-frequency-based BoW representation, Sp is different from the idf weighting because $idf(t_i)$ is based on $|\{\mathbf{z} \in D : z_i > 0\}|$, whereas Sp is based on $|\{\mathbf{z} \in D : \min(x_i, y_i) \leq z_i \leq \max(x_i, y_i)\}|$, where $x_i > 0$ and $y_i > 0$ when $t_i \in T_{\mathbf{x}} \cap T_{\mathbf{y}}$.

5.4.3 Computational complexity

In the term-frequency-based BoW vector representation, it appears that computing $|\{\mathbf{z} \in D : \min(x_i, y_i) \leq z_i \leq \max(x_i, y_i)\}|$ naively can be expensive, as it requires a range search to find the number of documents with the frequencies of t_i between x_i and y_i. Since all x_i

[2] It depends on the lengths of $\mathbf{x}$ and $\mathbf{y}$ and parameters a and b.

Table 5.4: Characteristics of datasets (N: Number of documents, M: Number of terms, C: Number of classes).

Name	N	M	C	Collection
Fbis	2,463	2,000	17	TREC collection
La1s	3,204	13,195	6	TREC collection
La2s	3,075	12,432	6	TREC collection
New3s	9,558	26,832	44	TREC collection
Ng20	18,821	5,489	20	20 Newsgroup collection
Ohscal	11,162	11,465	10	Ohsumed patients records
R8	7,674	3,497	8	Reuters collection
R52	9,100	7,379	52	Reuters collection
Wap	1,560	8,460	20	Yahoo web pages
Webkb	4,199	1,817	4	University web pages

are integers (term occurrence frequency counts), $|\{\mathbf{z} \in D : \min(x_i, y_i) \leq z_i \leq \max(x_i, y_i)\}|$ can be computed in constant time by the following simple pre-processing.

Let m_i be the maximum frequency of term t_i in any document in the given collection D. We can maintain a cumulative frequency count array F_i of size $m_i + 1$ where $F_i[j]$ contains the number of documents with occurrences of t_i fewer than or equal to j.

Using F_i, $|\{\mathbf{z} \in D : \min(x_i, y_i) \leq z_i \leq \max(x_i, y_i)\}|$ can be estimated in constant time as $F_i[\max(x_i, y_i)] - F_i[\min(x_i, y_i) - 1]$. Note that $\min(x_i, y_i)$ cannot be 0 because $|\{\mathbf{z} \in D : \min(x_i, y_i) \leq z_i \leq \max(x_i, y_i)\}|$ is computed only if $t_i \in T_{\mathbf{x}} \cap T_{\mathbf{y}}$ (i.e., $x_i > 0$ and $y_i > 0$) and thus $\min(x_i, y_i) > 0$.

The above pre-processing requires $O(MN)$ time and $O(Mm)$ space, where m is the average maximum frequency of terms.

With the above pre-processing, the runtime for computing the similarity of $\mathbf{x}$ and $\mathbf{y}$ using Sp is the same as that of the existing similarity measures, which is $O(M)$.

5.5 Empirical evaluation

In this section, we present the results of experiments conducted to evaluate the task-specific performances of Sp, BM25, weighted Jaccard and cosine similarity in the query-by-example task to retrieve documents similar to a given query document. We conducted experiments with both the term-frequency-based and binary BoW vector representations. We used different combinations of tf and idf-based term-weighting factors with the weighted Jaccard and cosine similarity measures.

5.5.1 Datasets and experimental set-up

We used 10 datasets from 6 benchmark document collections. The characteristics of the datasets are provided in Table 5.4. NG20, R8, R52 and Webkb are from Cardoso-Cachopo (2007)[3]; and the others are from Han and Karypis (2000)[4].

[3]BoW vectors available at: http://web.ist.utl.pt/acardoso/datasets/
[4]BoW vectors available at: http://www.cs.waikato.ac.nz/ml/weka/datasets.html

Each dataset was divided into two subsets $\mathcal{D}$ and $\mathcal{Q}$ using a 10-fold cross-validation such that $\mathcal{D}$ and $\mathcal{Q}$ have 90% and 10% of the documents, respectively. $\mathcal{D}$ was used as a given collection from which similar documents were extracted for each query document in $\mathcal{Q}$. For each $\mathbf{q} \in \mathcal{Q}$, documents in $\mathcal{D}$ were ranked in descending order of their similarities to $\mathbf{q}$ using different contending similarity measures. The top k documents were presented as similar documents to $\mathbf{q}$.

For performance evaluation, a document was considered to be similar to $\mathbf{q}$ if they have the same class label. In order to demonstrate the consistency of a measure at different top k retrieved results, we evaluated the precision at the top k retrieved results ($P@k$ in terms of percentage) with $k = 1, 2, \cdots, 25$ and used the mean average precision up to $k = 25$. The performance evaluation criterion was: $MAP@25 = \frac{\sum_{k=1}^{25} P@k}{25}$.

We repeated the experiment 10 times using each of the 10 folds as $\mathcal{Q}$ and the remaining 9 folds as $\mathcal{D}$. The average $MAP@25$ and standard error over the 10 runs were reported. The average $MAP@25$ of two measures were considered to be significantly different if their confidence intervals based on two standard errors did not overlap.

The free parameters a and b in BM25 were set to 1.2 and 0.95, respectively, as recommended by Paltoglou and Thelwall (2010) and Jones et al. (2000).

All the experimental set-ups and similarity measures were implemented in Java using the WEKA platform (Hall et al., 2009). All the experiments were conducted on a Linux machine with a 2.27 GHz processor and 16 GB memory. We discuss the experimental results with the term-frequency-based and binary BoW vector representations separately in the following two subsections.

5.5.2 Results in the term-frequency-based BoW vector representation

Here we used two term-weighting schemes: tf factor only and tf-idf factors, with weighted Jaccard and cosine. The six contending measures were: *Sp*, *BM25*, *Cos.tf-idf* (cosine with tf-idf), *Cos.tf* (cosine with tf only), *WJac.tf-idf* (weighted Jaccard with tf-idf) and *WJac.tf* (weighted Jaccard with tf only).

The average $MAP@25$ and standard error over 10 runs of six contending measures are provided in Table 5.5 and the summarised results in terms of pairwise win-loss-draw counts of contending measures based on the two standard error significance test over the 10 datasets used in the experiment are provided in Table 5.6.

Table 5.5 shows that *Sp* and *Cos.tf* produced the best or competitive to the best result in five datasets each, followed by *WJac.tf-idf* in four, whereas *Cos.tf-idf*, *BM25* and *WJac.tf* were best or competitive to the best measure in only one dataset each.

The first column in Table 5.6 shows that *Sp* had more wins than losses over all contending measures. It had one more wins than losses against the closest contenders *Cos.tf* and *WJac.tf-idf*.

Of the two cosine measures, *Cos.tf* had more wins than losses to *Cos.tf-idf*. This shows that the idf term weighting can be counter-productive with cosine in inter-document similarity measurement. This is mainly due to the cosine normalisation which penalises documents with rare terms (with high idf weights) which are not in $\mathbf{q}$. In comparison

Table 5.5: Term-frequency-based BoW representation: Average MAP@25 and standard error over 10 runs. The best result is underlined and the results equivalent (insignificant difference based on two standard errors) to the best result are bold-faced.

	BM25	*Cos.tf-idf*	*Cos.tf*	*WJac.tf-idf*	*WJac.tf*	*Sp*
Fbis	65.12±0.62	**68.42±0.61**	**68.28±0.58**	**68.48±0.49**	66.75±0.54	**67.77±0.51**
La1s	74.41±0.32	75.97±0.42	73.08±0.49	**79.18±0.33**	77.54±0.47	**79.36±0.32**
La2s	76.42±0.49	78.11±0.42	75.24±0.44	**81.06±0.42**	79.45±0.37	**80.89±0.40**
New3s	67.01±0.18	68.31±0.19	**70.19±0.19**	69.36±0.16	68.45±0.15	68.98±0.16
Ng20	**76.47±0.19**	74.81±0.24	67.80±0.28	73.67±0.23	64.28±0.24	72.30±0.20
Ohscal	59.72±0.22	53.59±0.21	**61.06±0.26**	59.68±0.21	60.81±0.20	60.14±0.19
R52	85.50±0.20	80.80±0.27	**86.57±0.15**	84.55±0.21	84.72±0.19	84.39±0.22
R8	91.05±0.14	86.14±0.22	**92.93±0.19**	91.03±0.18	91.94±0.21	91.40±0.17
Wap	19.67±0.42	65.33±0.34	61.97±0.41	**70.54±0.46**	65.10±0.48	**70.92±0.50**
Webkb	70.28±0.23	68.55±0.24	73.04±0.27	73.90±0.31	**75.25±0.25**	**74.91±0.33**

Table 5.6: Term-frequency-based BoW representation: Win-loss-draw counts of measures in columns against those in rows based on the two standard error significance test over 10 runs.

	Sp	*WJac.tf*	*WJac.tf-idf*	*Cos.tf*	*Cos.tf-idf*
BM25	8-2-0	8-2-0	6-2-2	7-0-3	5-5-0
Cos.tf-idf	8-1-1	6-2-2	8-1-1	5-4-1	
Cos.tf	5-4-1	4-5-1	5-4-1		
WJac.tf-idf	3-2-5	3-6-1			
WJac.tf	5-2-3				

to *BM25*, *Cos.tf* produced better results with seven wins and no loss, and *Cos.tf-idf* was competitive with five wins versus five losses.

In the Wap dataset, *BM25* produced significantly worse results than the other contenders, due to the idf factor used in *BM25*. If a term t_i occurs in more than half of the documents in $\mathcal{D}$ (i.e., $n_i > \frac{N}{2}$), $idf_{bm25}(t_i)$ is negative and t_i has negative contribution to the similarity of two documents. When $idf_{bm25}(t_i)$ was replaced by the traditional $idf(t_i)$ in the formulation of *BM25* (Eqn 5.2), it produced MAP@25 = 67.04%, which was still worse than those of *Sp* and *WJac.tf-idf*.

In weighted Jaccard similarity, *WJac.tf-idf* produced better retrieval results than *WJac.tf*. It is interesting to note that *WJac.tf-idf* produced better retrieval results than *Cos.tf-idf*, *Cos.tf* and *BM25*. This could be mainly due to the vector length normalisations used in *BM25* and *cosine* that penalise documents with higher frequencies of terms which are not in $\mathbf{q}$.

Sp and *WJac.tf-idf* produced more consistent results than the other contending measures. They did not produce the worst result in any dataset, whereas *WJac.tf* produced the worst result in one dataset (NG20) followed by *Cos.tf* in two datasets (La1s and La2s), *BM25* in three datasets (Fbis, New3s and Wap), and *Cos.tf-idf* in four datasets (Ohscal, R8, R52 and Webkb).

In terms of runtime, all measures had runtimes of the same order of magnitude. For example, in the NG20 dataset, the average total runtime of one run (including pre-processing)

Table 5.7: Binary BoW representation: Average MAP@25 and standard error over 10 runs. The best result is underlined and the results equivalent (insignificant difference based on two standard errors) to the best result are bold-faced.

	BM25	*Cos.idf*	*Cos*	*WJac.idf*	*WJac*	*Sp*
Fbis	**67.90±0.50**	66.46±0.50	63.24±0.56	**67.17±0.46**	64.58±0.52	**66.94±0.47**
La1s	74.78±0.25	76.78±0.34	75.96±0.38	**78.54±0.34**	77.55±0.39	**79.04±0.30**
La2s	76.71±0.48	78.48±0.42	77.55±0.38	**80.02±0.39**	79.12±0.35	**80.54±0.40**
New3s	**69.61±0.20**	66.73±0.16	64.88±0.15	67.76±0.15	65.66±0.16	68.13±0.16
Ng20	**74.37±0.16**	73.80±0.17	64.12±0.20	72.26±0.19	63.07±0.21	72.61±0.20
Ohscal	**58.95±0.19**	55.06±0.17	58.56±0.18	58.66±0.21	58.45±0.17	**59.23±0.19**
R52	**83.87±0.24**	79.01±0.28	**84.19±0.20**	83.23±0.22	83.36±0.21	**83.80±0.22**
R8	90.54±0.16	86.03±0.19	**91.60±0.17**	90.24±0.17	91.10±0.20	90.92±0.18
Wap	16.47±0.34	66.97±0.47	59.16±0.44	**70.18±0.54**	65.09±0.48	**70.02±0.53**
Webkb	73.29±0.39	70.86±0.23	**75.61±0.27**	74.19±0.37	**75.59±0.29**	74.97±0.35

Table 5.8: Binary BoW representation: Win-loss-draw counts of measures in columns against those in rows based on the two standard error significance test over 10 runs.

	Sp	*WJac*	*WJac.idf*	*Cos*	*Cos.idf*
BM25	5-2-3	5-5-0	4-3-2	4-4-2	3-7-0
Cos.idf	8-1-1	5-4-1	8-1-1	4-6-0	
Cos	7-2-1	5-3-2	6-3-1		
WJac.idf	5-0-5	2-6-2			
WJac	8-0-2				

using *Sp* took 15935 seconds, whereas *BM25*, *Cos.tf-idf* and *WJac.tf-idf* took 27432, 16089 and 14875 seconds, respectively.

5.5.3 Results in the binary BoW vector representation

Here, the six contending measures were: *Sp*, *BM25*, *Cos.idf* (cosine with idf), *Cos* (cosine without idf), *WJac.idf* (weighted Jaccard with idf) and *WJac* (weighted Jaccard without idf). Note that *WJac* which does not use any term weighting is equivalent to the traditional Jaccard similarity defined in Eqn 5.4.

The average MAP@25 and standard error over 10 runs of the six contending measures are provided in Table 5.7, and the summarised results in terms of pairwise win-loss-draw counts of contending measures based on the two standard error significance test over the 10 datasets used in the experiment are provided in Table 5.8.

Table 5.7 shows that *Sp* produced the best or competitive to the best result in six datasets, followed by *BM25* in five, *WJac.idf* in four, *Cos* in two, and *WJac* in one dataset only. *Cos.idf* did not produce a competitive result to the best performing measure in any dataset.

In terms of pairwise win-loss-draw counts as shown in the first column in Table 5.8, *Sp* had many more wins than losses against all other contending measures.

It is interesting to note that *BM25*, *Cos.idf* and *Cos* using the binary BoW representation produced better retrieval results than their respective counterparts using the term-frequency-based BoW representation in some datasets. For example, (i) *BM25* in

Fbis, New3s and Webkb; (ii) *Cos.idf* in La1s, Ohscal, Wap and Webkb; and (iii) *Cos* in La1s, La2s and Webkb. In contrast, *WJac.idf*, *WJac* and *Sp* using binary BoW vectors did not produce better retrieval results than their respective counterparts using term-frequency-based BoW vectors.

As in the term-frequency-based BoW representation, all measures had runtimes of the same order of magnitude.

5.6 Discussion

Although some studies have used different variants of tf and idf term-weighting factors with the most widely-used cosine similarity, the tf and idf factors discussed in Section 5.2.1 have been shown to be the most consistent in the IR task (Singhal, 1997).

For the tf factor, instead of using the logarithmic scaling of x_i, some researchers have used other scaling approaches, such as augmented $\left(0.5 + 0.5 \times \frac{x_i}{\max(x_1, x_2, \cdots, x_M)}\right)$ (Salton and Buckley, 1988) and Okapi $\left(\frac{x_i}{2+x_i}\right)$ (Robertson et al., 1994). Similarly, for the idf factor, instead of using $\frac{N}{n_i}$, some researchers have used the probabilistic idf factor based on $\frac{N-n_i}{n_i}$ (Robertson et al., 1994; Singhal, 1997). Note that BM25 (Eqn 5.2) uses a tf factor similar to Okapi and an idf factor similar to the probabilistic idf factor (Roberston and Zaragoza, 2009).

In the supervised text-mining task of document classification, different approaches utilising class information have been proposed to estimate the collection-based term-weighting factors (Wang and Zhang, 2013; Debole and Sebastiani, 2003; Lan et al., 2009). Inverse category frequency (icf) (Wang and Zhang, 2013) has been shown to produce better classification results than the traditional idf factor with the cosine similarity measure. Icf considers the distribution of a term among classes rather than among documents in the given collection. The intuition behind icf is that the fewer classes in which a term t_i occurs, the more discriminating power the term t_i contributes to classification (Wang and Zhang, 2013). If C and c_i are the total number of classes and the number of classes in which t_i occurs at least once in at least one document, then the icf factor is estimated as $icf(t_i) = \log\left(1 + \frac{C}{c_i}\right)$.

We evaluated the performance of Sp in the kNN document classification task with existing measures using the supervised term-weighting scheme of icf (Wang and Zhang, 2013). Sp produced either better or competitive classification results with existing measures using supervised or unsupervised term weighting in the 5NN classification task. The classification results are provided in the Appendix 5.A.

Although the weighted Jaccard similarity has been used in other application domains (Chierichetti et al., 2010), it is not widely used to measure similarities of BoW documents. Our experimental results in Section 5.5 show that the weighted Jaccard similarity with the tf-idf term-weighting scheme may be an effective alternative to cosine and BM25 in inter-document similarity measurement.

Sp has superior performance over all contenders in the binary BoW vector representation. It can be very useful in application domains such as legal and medical where the

exact term frequency information may not be available due to privacy issues, because it is possible to infer information in a document from its term frequencies (Zhu et al., 2008).

5.7 Concluding remarks

For the purpose of inter-document similarity measurement tasks, we identify the limitations of the underlying assumptions of the most widely-used tf-idf term-weighting scheme employed in existing measures such as cosine and BM25, and provide an alternative assumption which is more intuitive in this task.

Based on the new assumption, we introduce a new simple but effective inter-document similarity measure called Sp.

Our empirical evaluation in the query-by-example task shows that:

1. Sp produces better or at least competitive results to the existing similarity measures with the state-of-the-art term-weighting schemes in term-frequency-based BoW representations. Sp produces more consistent results than the existing measures across different datasets.
2. Sp produces better results than the existing similarity measures with or without idf term weighting in the case of binary BoW representation.

When cosine and BM25 are employed, our results show that it is important to use an appropriate BoW vector representation (binary or term-frequency-based) and an appropriate term-weighting scheme. Using inappropriate representation and term weighting can result in poor performance.

In contrast, using Sp, users are not required to apply any additional term weighting to measure the similarity of two documents and still obtain better or competitive results in comparison to the best results obtained by cosine or BM25.

Acknowledgement

A preliminary version of this paper was published in the Proceedings of the 11th Asia Information Retrieval Societies Conference 2015 (Aryal et al., 2015).

References

Aryal, S., Ting, K. M., Haffari, G. and Washio, T. (2014b). Mp-dissimilarity: A data dependent dissimilarity measure, *In Proceedings of the IEEE International Conference on Data Mining*, IEEE, pp. 707–712.

Aryal, S., Ting, K. M., Haffari, G. and Washio, T. (2015). Beyond tf-idf and cosine distance in document dissimilarity measures, *In Proceedings of the 11th Asia Information Retrieval Societies Conference*, Springer, Cham, pp. 400–406.

Aryal, S., Ting, K. M., Washio, T. and Haffari, G. (2017). Data-dependent dissimilarity measure: an effective alternative to geometric distance measures, *Knowledge and Information Systems* pp. 1–28, doi:10.1007/s10115–017–1046–0.

Cardoso-Cachopo, A. (2007). *Improving Methods for Single-label Text Categorization*, PhD thesis, Instituto Superior Tecnico, Technical University of Lisbon, Lisbon, Portugal.

Chierichetti, F., Kumar, R., Pandey, S. and Vassilvitskii, S. (2010). Finding the Jaccard Median, *In Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, Society for Industrial and Applied Mathematics, pp. 293–311.

Debole, F. and Sebastiani, F. (2003). Supervised term weighting for automated text categorization, *In Proceedings of the 2003 ACM Symposium on Applied Computing*, ACM, New York, USA, pp. 784–788.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. (2009). The WEKA Data Mining Software: An Update, *SIGKDD Exploration Newsletter* **11**(1): 10–18.

Han, E.-H. and 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-Verlag, London, UK, pp. 424–431.

Han, X., Li, S. and Shen, Z. (2012). A k-NN Method for Large Scale Hierarchical Text Classification at LSHTC3, *In Proceedings of the Workshop on Large Scale Hierarchical Classification at European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pp. 1–12.

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura, *Bulletin de la Socit Vaudoise des Sciences Naturelles* **37**: 547–579.

Joachims, T. (1997). A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization, *In Proceedings of the Fourteenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, USA, pp. 143–151.

Jones, K. S., Walker, S. and Robertson, S. E. (2000). A probabilistic model of information retrieval: Development and comparative experiments, *Information Processing and Management* **36**(6): 779–808.

Lan, M., Tan, C. L., Su, J. and Lu, Y. (2009). Supervised and traditional term weighting methods for automatic text categorization, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31**(4): 721–735.

Manning, C. D., Raghavan, P. and Schtze, H. (2008). *Introduction to Information Retrieval*, Cambridge University Press, New York, USA.

Paltoglou, G. and Thelwall, M. (2010). A study of information retrieval weighting schemes for sentiment analysis, *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 1386–1395.

Roberston, S. and Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond, *Foundations and Trends in Information Retrieval* **3**(4): 333–389.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. and Gatford, M. (1994). Okapi at trec-3, *In Proceedings of the Third Text Retrieval Conference (TREC)*, pp. 109–126.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval, *Information Processing and Management* **24**(5): 513–523.

Salton, G. and McGill, M. J. (1986). *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, USA.

Singhal, A., Buckley, C. and Mitra, M. (1996). Pivoted document length normalization, *In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New York, USA, pp. 21–29.

Singhal, A. K. (1997). *Term Weighting Revisited*, PhD thesis, The Faculty of the Graduate School, Cornell University.

Wang, D. and Zhang, H. (2013). Inverse-category-frequency based supervised term weighting schemes for text categorization, *Journal of Information Science and Engineering* **29**(2): 209–225.

Zhu, X., Goldberg, A. B., Rabbat, M. and Nowak, R. (2008). Learning Bigrams from Unigrams, *In Proceedings of ACL-08: HLT*, Association for Computational Linguistics, pp. 656–664.

Zobel, J. and Moffat, A. (1998). Exploring the similarity space, *SIGIR Forum* **32**(1): 18–34.

Appendix 5.A: kNN classification results

In order to predict a class label for a test document $\mathbf{q}$, its k nearest neighbour (or most similar) documents were searched in the given labelled training set of documents using a contending similarity measure, and the majority class among the kNNs was predicted as the class label for $\mathbf{q}$.

All classification experiments were conducted using a 10-fold cross-validation (10 runs with each one out of the 10 folds as the test set and the remaining 9 folds as the training set). The average classification accuracy and standard error over a 10-fold cross-validation were reported. All collection-based term-weighting factors (idf and icf) were computed from the training set and used in both the training and test documents. The parameter k was set to the commonly-used value of 5 (i.e., 5NN classification was used).

Table 5.9: Term-frequency-based BoW representation: Win-loss-draw counts of measures in columns against those in rows based on the two standard errors significance test over a 10-fold cross-validation of 5NN classification.

	Sp	*WJac.tf*	*WJac.tf-idf*	*WJac.tf-icf*	*Cos .tf*	*Cos.tf-idf*	*Cos.tf-icf*
BM25	6-2-2	7-2-1	6-2-2	7-2-1	5-2-3	4-5-1	2-3-5
Cos.tf-icf	6-1-3	5-2-3	5-0-5	6-1-2	4-2-4	2-5-3	
Cos.tf-idf	8-0-2	5-1-4	9-0-1	6-1-3	5-4-1		
Cos.tf	5-4-1	4-3-3	5-3-2	5-1-4			
WJac.tf-icf	2-2-6	0-3-7	3-2-5				
WJac.tf-idf	1-1-8	2-5-3					
WJac.tf	4-1-5						

We discuss the 5NN classification results with the term-frequency-based and binary BoW vector representations separately in the following two subsections.

Term-frequency-based BoW vector representation

We used term weighting based on tf only, tf-idf and tf-icf with weighted Jaccard and cosine, resulting in eight contending measures: *Sp*, *BM25*, *Cos.tf-icf*, *Cos.tf-idf*, *Cos.tf*, *WJac.tf-icf*, *WJac.tf-idf* and *WJac.tf*.

The average classification accuracies and standard errors over a 10-fold cross-validation of the eight contending measures are provided in Table 5.10 and the summarised results in terms of pairwise win-loss-draw counts of contending measures based on the two standard error significance test in the 10 datasets used in the experiment are provided in Table 5.9.

The 5NN classification accuracies in Table 5.10 show that *Sp*, *WJac.tf-idf*, *WJac.tf-icf* and *Cos.tf* produced the best or competitive to the best result in five datasets each, followed by *WJac.tf* in four, *Cos.tf-icf* and *BM25* in two datasets each, and *Cos.tf-idf* in one dataset only.

The pairwise win-loss-draw counts of *Sp* in the first column of Table 5.9 show that it had more wins than losses over all contending measures except *Wjac.tf-idf* and *Wjac.tf-icf*, where it had competitive results with the same number of wins and losses.

Sp and all three variants of weighted Jaccard similarity produced better classification results than all three variants of cosine and BM25. As in the similar document retrieval task discussed in Section 5.5, BM25 produced the worst classification accuracy in the Wap dataset because of $idf_{bm25}(t_i)$. The classification accuracy was increased to 79.42% when $idf_{bm25}(t_i)$ was replaced by the traditional idf $idf(t_i)$.

The supervised term weighting using icf (tf-icf) did not always produce better classification results than the traditional tf-idf-based term weighting with both cosine and weighted Jaccard. It had five wins and two losses with cosine, whereas it had two wins and three losses with weighted Jaccard.

Table 5.10: Term-frequency-based BoW representation: Average 5NN classification accuracy and standard error over a 10-fold cross-validation. The best result is underlined and the results equivalent (insignificant difference based on two standard errors) to the best result are bold-faced.

	BM25.tf	*Cos.tf.icf*	*Cos.tf.idf*	*Cos.tf*	*WJac.tf.icf*	*WJac.tf.idf*	*WJac.tf*	*Sp*
Fbis	76.98 ± 1.04	**80.83 ± 0.84**	**79.33 ± 1.05**	**80.23 ± 0.78**	**79.29 ± 0.95**	**79.54 ± 0.91**	**79.21 ± 0.84**	**79.21 ± 0.80**
La1s	83.68 ± 0.58	83.43 ± 0.84	86.70 ± 0.66	82.30 ± 0.80	87.30 ± 0.80	**88.89 ± 0.61**	87.05 ± 0.70	**88.48 ± 0.46**
La2s	86.28 ± 0.50	84.52 ± 0.47	87.93 ± 0.77	84.23 ± 0.61	88.46 ± 0.63	**90.11 ± 0.41**	88.03 ± 0.56	**89.59 ± 0.48**
New3s	79.31 ± 0.30	79.54 ± 0.34	78.99 ± 0.34	**81.47 ± 0.29**	**80.90 ± 0.30**	**80.86 ± 0.33**	80.60 ± 0.36	80.55 ± 0.36
Ng20	**88.55 ± 0.15**	87.57 ± 0.22	86.92 ± 0.22	84.74 ± 0.34	86.28 ± 0.27	87.41 ± 0.25	83.05 ± 0.28	86.62 ± 0.17
Ohscal	72.63 ± 0.29	72.04 ± 0.43	66.95 ± 0.45	**74.25 ± 0.46**	**74.50 ± 0.29**	72.36 ± 0.21	**74.22 ± 0.33**	73.19 ± 0.34
R52	**92.30 ± 0.20**	91.20 ± 0.28	87.72 ± 0.54	**92.18 ± 0.18**	91.69 ± 0.22	91.17 ± 0.22	90.63 ± 0.21	90.94 ± 0.25
R8	95.19 ± 0.21	**95.34 ± 0.17**	90.80 ± 0.25	**95.81 ± 0.23**	**95.39 ± 0.20**	94.98 ± 0.23	**95.27 ± 0.31**	95.28 ± 0.27
Wap	17.76 ± 0.79	75.90 ± 0.46	76.92 ± 0.76	72.44 ± 0.58	80.70 ± 0.80	**82.31 ± 0.92**	76.22 ± 0.58	**82.50 ± 0.79**
Webkb	81.16 ± 0.38	81.86 ± 0.48	77.92 ± 0.43	81.58 ± 0.57	**84.14 ± 0.42**	83.33 ± 0.61	**84.40 ± 0.38**	**84.33 ± 0.53**

Table 5.11: Binary BoW representation: Average 5NN classification accuracy and standard error over a 10-fold cross-validation. The best result is underlined and the results equivalent (insignificant difference based on two standard errors) to the best result are bold-faced.

	BM25	*Cos.icf*	*Cos.idf*	*Cos*	*WJac.icf*	*WJac.idf*	*WJac*	*Sp*
Fbis	**79.29 ± 0.65**	**79.62 ± 0.99**	77.75 ± 0.95	**78.20 ± 0.88**	**79.82 ± 1.08**	**79.86 ± 0.98**	**78.28 ± 0.81**	**79.05 ± 0.82**
La1s	84.27 ± 0.63	**87.45 ± 0.79**	**87.70 ± 0.63**	85.89 ± 0.72	**88.05 ± 0.73**	**88.70 ± 0.54**	87.55 ± 0.62	**88.67 ± 0.49**
La2s	86.08 ± 0.58	87.48 ± 0.49	**89.43 ± 0.42**	86.67 ± 0.50	88.52 ± 0.68	**89.99 ± 0.43**	88.00 ± 0.60	**90.15 ± 0.48**
New3s	**80.72 ± 0.42**	79.10 ± 0.36	78.31 ± 0.41	78.19 ± 0.35	79.79 ± 0.36	**80.03 ± 0.41**	78.74 ± 0.29	**80.15 ± 0.40**
Ng20	**87.59 ± 0.13**	87.19 ± 0.20	87.25 ± 0.19	82.80 ± 0.20	85.64 ± 0.12	86.61 ± 0.18	82.16 ± 0.20	86.84 ± 0.24
Ohscal	72.02 ± 0.32	72.36 ± 0.31	68.54 ± 0.32	72.89 ± 0.24	**73.65 ± 0.28**	72.36 ± 0.32	72.89 ± 0.29	72.79 ± 0.33
R52	**91.20 ± 0.24**	89.74 ± 0.38	86.14 ± 0.43	90.21 ± 0.27	90.51 ± 0.25	89.81 ± 0.24	89.75 ± 0.17	**90.80 ± 0.19**
R8	**94.80 ± 0.21**	**95.05 ± 0.13**	90.98 ± 0.44	**94.99 ± 0.23**	**95.10 ± 0.22**	94.54 ± 0.30	**94.86 ± 0.17**	**95.05 ± 0.31**
Wap	15.51 ± 0.69	76.86 ± 0.65	78.27 ± 1.01	69.68 ± 0.82	80.00 ± 0.56	**81.92 ± 0.96**	76.28 ± 0.62	**81.60 ± 0.81**
Webkb	**83.97 ± 0.49**	**84.68 ± 0.54**	81.45 ± 0.60	**84.26 ± 0.49**	**84.88 ± 0.44**	**84.16 ± 0.47**	**84.73 ± 0.42**	**84.71 ± 0.53**

Table 5.12: Binary BoW representation: Win-loss-draw counts of measures in columns against those in rows based on the two standard errors significance test over a 10-fold cross-validation of 5NN classification.

	Sp	*WJac*	*WJac.idf*	*WJac.icf*	*Cos*	*Cos.idf*	*Cos.icf*
BM25	4-1-5	4-3-3	3-2-5	4-3-3	3-3-4	3-6-1	3-3-4
Cos.icf	4-0-6	0-1-8	3-2-5	3-1-6	0-4-6	1-5-4	
Cos.idf	6-0-4	4-3-3	7-1-2	7-1-2	4-4-2		
Cos	6-0-4	3-2-5	5-0-5	6-0-4			
WJac.icf	3-1-6	0-5-5	3-3-4				
WJac.idf	1-0-9	0-4-6					
WJac	6-0-4						

Binary BoW vector representation

We used weighted Jaccard and cosine similarities with and without idf and icf weighting, resulting in eight contending measures: *Sp*, *BM25*, *Cos.idf*, *Cos.icf*, *Cos*, *WJac.idf*, *WJac.icf* and *WJac*.

The average classification accuracies and standard errors over a 10-fold cross-validation of the eight contending measures are provided in Table 5.11, and the summarised results in terms of pairwise win-loss-draw counts of contending measures, based on the two standard error significance test in the 10 datasets used in the experiment, are provided in Table 5.12.

The 5NN classification accuracies in Table 5.11 show that *Sp* produced the best or competitive to the best result in eight datasets. The closest contenders *BM25* and *WJac.idf* produced the best or competitive to the best result in six datasets each, followed by *WJac.icf* in five, *Cos.icf* in four, *WJac* and *Cos* in three datasets each, and *Cos.idf* in two datasets only.

In terms of pairwise win-loss-draw counts, as shown in the first column in Table 5.12, *Sp* had more wins than losses against all other contending measures. It had one win and no loss against *WJac.idf* and three wins and one loss against *WJac.icf*.

As in the term-frequency-based BoW representation, the supervised term-weighting scheme based on icf did not always produce better classification results than the traditional idf-based term-weighting scheme with both cosine and weighted Jaccard in the binary BoW vector presentation. It had five wins and one loss with cosine, whereas it had three wins and three losses with the weighted Jaccard.

It is interesting to note that *BM25*, *Cos.icf*, *Cos.idf* and *Cos* which use the binary BoW vector representation produced better classification accuracies than their respective counterparts using the term-frequency-based BoW representation in some datasets; e.g., *BM25* was better in three datasets (Fbis, New3s, WebKb), *Cos.icf* and *Cos* in three datasets (La1s, La2s, Webkb), and *Cos.idf* in two datasets (La2s, Webkb). However, all three variants of weighted Jaccard and *Sp* with the term-frequency-based BoW representation produced either better or competitive results with the binary BoW representation.

Chapter 6

Thesis conclusions and future work

This chapter concludes the thesis in Section 6.1 and provides potential avenues for future research in Section 6.2.

6.1 Thesis conclusions

To overcome the limitations of conventional distance-based (dis)similarity measures such as ℓ_p-norm and cosine stated in Section 1.3 of Chapter 1, this thesis has investigated an alternative approach of measuring similarities of data instances. The conclusions of this thesis are summarised as follows:

6.1.1 Traits of an effective similarity measure

This thesis has shown that fully data-dependent similarity and robustness to units and scales of measurement are important characteristics of a similarity measure to produce consistent task-specific performance across a wide range of datasets. The task-specific performances of distance-based similarity measures vary significantly in different datasets because they do not possess these characteristics.

In traditional distance-based similarity measures, the similarity of two instances is based solely on their geometric positions in the feature space, which is independent of the underlying data distribution. This thesis has shown that exploiting the local data distribution between the two instances is more beneficial than using spatial distance alone in measuring their similarity.

Simply having the same value (i.e., zero distance) in a dimension does not necessarily mean that the two instances are similar in that dimension. The probability mass around zero distance contributes useful information towards the similarity measurement in discriminating instances in this and other dimensions. For example, having zero distance in a dimension where every instance has the same value provides less information about the similarity of two instances than having zero distance in a dimension where only those two instances have the same value and other instances are significantly different from them.

This becomes more prominent in high-dimensional problems where data often lie in low-dimensional subspaces and many instances have the same value in many dimensions in the original space.

Distance-based similarity measures are sensitive to units and scales which are often unknown in data mining where only feature values are given. In order to produce good similarity results using distance measures, it is important to transform data by pre-processing to ensure all dimensions are on the same scale. Using inappropriate scales may result in poor task-specific performance. However, finding the appropriate scale is difficult. Therefore, measures which are robust to units and scales of measurement provide more consistent results than similarity measures which are sensitive.

6.1.2 m_p-dissimilarity: An effective alternative to distance measures

This thesis has introduced a new data-dependent measure called "m_p-dissimilarity" which has both of the above-mentioned characteristics. It has the same formulation as the traditional ℓ_p-norm, but the geometric distance of two instances in each dimension is replaced with the probability data mass between them.

The dissimilarity of two instances in each dimension is data-dependent, including self-dissimilarity. Under m_p-dissimilarity, having the same value in a dimension where every instance has the same value contributes less in the overall similarity of two instances than having the same value in a dimension where only those two instances have that value.

As m_p-dissimilarity does not use the actual feature values in the dissimilarity measure and the dissimilarity of two instances is based on the number of instances between them, the dissimilarity measure is robust to units and scales of measurement. It does not require any pre-processing to standardise or normalise data.

Replacing spatial distance with probability mass, m_p-dissimilarity produces similar or better task-specific performance than traditional distance measures over a wide range of datasets from different application domains, particularly in high-dimensional datasets such as bag-of-words (BoW) document collections.

The superior performance of m_p-dissimilarity over distance-based measures in high-dimensional datasets confirms that a measure which is fully data-dependent and robust to units and scales of measurement is less affected by the "curse of dimensionality" issue than distance-based similarity measures.

m_p-dissimilarity is a generic data-dependent measure, of which the existing data-dependent measures of rank difference and Lin's probabilistic similarity are special cases with data-independent self-dissimilarities.

In order to handle datasets with mixed numeric and nominal attributes, existing dissimilarity measures require the conversion of one attribute type to another. In contrast, m_p-dissimilarity provides seamless treatment to numeric and nominal attributes directly using probability mass in mixed domains without the need for any conversion.

m_p-dissimilarity is a non-metric dissimilarity measure as self-dissimilarity is neither zero nor any other fixed constant.

6.1.3 Sp: A new BoW document similarity measure

In the BoW vector space model, existing document similarity measures such as cosine and BM25 use term weighting to measure inter-document similarity. This thesis has shown that term weighting can be detrimental in the inter-document similarity measurement task because the underlying assumptions of term weighting do not hold in this case. This thesis provides a more congruous alternative assumption for inter-document similarity measurement.

This thesis has introduced a new BoW inter-document similarity measure called Sp, based on the new assumption for inter-document similarity measurement. Sp is a simplified version of m_0-dissimilarity. Sp does not require explicit term weighting but produces task-specific results better than or competitive with existing measures.

6.1.4 Relative mass to improve the task-specific performance of iForest

Although iForest has been shown to be effective in anomaly detection and content-based information retrieval (CBIR), this thesis has identified its limitations in both tasks and introduced the notion of "relative mass" to overcome these limitations:

- In anomaly detection, iForest fails to detect local anomalies that lie close to a dense normal cluster but have density similar to other sparse normal clusters. This is mainly because path length used in iForest is a global measure with respect to the root of each tree and it does not consider local variation in the data distribution. As in the case of density-based paradigm where relative density is used to capture the variation in local distribution, relative mass is a ranking measure that considers local variation in the data distribution. Unlike relative density, relative mass can be estimated efficiently, because it does not require pairwise distance calculations. Relative mass enables iForest to detect local anomalies without creating new weaknesses.
- In CBIR, ReFeat, which has iForest at its core, does not guarantee that relevant instances lie in the same local neighbourhood. ReMass-ReFeat based on relative mass guarantees that relevant instances lie in the same local neighbourhood. ReMass-ReFeat produces produces better CBIR results than ReFeat and existing distance-based CBIR systems, and requires much smaller ensemble size than ReFeat.

6.2 Future work

The pairwise (dis)similarity measurements of data instances is a core computation in many data-mining algorithms. This thesis has introduced a data-dependent dissimilarity measure where even self-dissimilarity is data-dependent. It is more effective than distance-based measures in high-dimensional problems. This opens many opportunities for future research in designing (dis)similarity measures which are not geometry-based. Potential areas for future work include the following:

6.2.1 Using m_p-dissimilarity in other data-mining tasks

This thesis has shown that m_p-dissimilarity produces better results than data-independent and other data-dependent measures in CBIR and kNN classification tasks. It will be interesting to evaluate their relative performance in other tasks, such as clustering and anomaly detection.

Using m_p-dissimilarity in some existing algorithms such as kMeans clustering (Macqueen, 1967) is not a simple replacement of distance measure. This is because using an arbitrary (dis)similarity measure in kMeans is not straightforward. Each iteration of kMeans requires updating cluster centres to minimise intra-cluster distances or maximise intra-cluster similarities. This can be done easily if Euclidean distance (ℓ_2-norm) or Manhattan distance (ℓ_1-norm) is used by computing the mean or median vector of the instances assigned to each cluster. With other (dis)similarity measures, new cluster centres have to be searched through optimisation.

6.2.2 Investigating mathematical properties of m_p-dissimilarity

The limitations of distance measures in high-dimensional spaces have been studied in terms of two phenomena:

- distance concentration - under certain assumptions in data distribution, the contrast in distances of different instances diminishes as the number of dimensions increases (Beyer et al., 1999; Aggarwal et al., 2001; François et al., 2007).
- hubness - the distribution of k-occurrences, which is the number of other instances in a given dataset of which an instance is one of their k nearest neighbours, becomes considerably skewed as the number of dimensions increases (Radovanović et al., 2010).

Chapter 3 of this thesis shows empirically that the concentration effect in m_p-dissimilarity is even worse than that in distance-based measures, whereas the effect of hubness is not as severe as in distance-based measures. It is worth investigating this empirical result more closely to provide concrete theoretical evidence.

Some data-mining algorithms exploit the mathematical properties of similarity measures in the learning process. For example, support vector machine (SVM) requires similarity measures to be valid kernels (Cristianini and Shawe-Taylor, 2000). It is not clear if m_p-dissimilarity is a valid kernel.

Therefore, another potential avenue for future research is to study the mathematical properties of m_p-dissimilarity and investigate the implications of having data-dependent similarity and not satisfying the metric axioms.

6.2.3 Developing pruning strategies to speed up nearest neighbour search using m_p-dissimilarity in very large datasets

For distance-based similarity measures, different indexing schemes have been developed using efficient data structures, such as k-d tree (Bentley and Friedman, 1979), R*-tree

(Beckmann et al., 1990), M-tree (Ciaccia et al., 1997) and Cover trees (Beygelzimer et al., 2006) to speed up the nearest neighbour search in very large datasets. These schemes are based on the spatial positions of data in the geometric space. It will be interesting to investigate whether similar efficient data structures can be developed to speed up the nearest neighbour search using m_p-dissimilarity.

Another potential avenue for future research is examining whether pruning strategies such as Locality Sensitive Hashing (LSH) (Indyk and Motwani, 1998) can be used with m_p-dissimilarity. Because of the implementation of m_p-dissimilarity using equal-frequency bin discretisation, it appears to have some similarity with LSH, although the aims of binning are different in the two cases. It is an open question whether LSH can be used to generate candidate nearest neighbour sets quickly for m_p-dissimilarity. LSH has nice theoretical bounds for the Euclidean distance but it is not clear if similar bounds can be derived for m_p-dissimilarity.

6.2.4 Investigating the effectiveness of Sp in measuring similarities of documents using word embedding

This thesis has shown that Sp, a variant of m_p-dissimilarity, produces better task-specific results than widely-used document similarity measures with the traditional BoW (Salton and McGill, 1986) vector representation. It would be interesting to investigate if Sp produces better task-specific results than existing similarity measures if documents are represented as vectors using word embedding techniques such as word2vec (Mikolov et al., 2013a,b; Le and Mikolov, 2014).

References

Aggarwal, C. C., Hinneburg, A. and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space, *In Proceedings of the International Conference on Database Theory*, Springer, Berlin Heidelberg, pp. 420–434.

Beckmann, N., Kriegel, H.-P., Schneider, R. and Seeger, B. (1990). The R*-tree: An Efficient and Robust Access Method for Points and Rectangles, *In Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, ACM, New York, USA, pp. 322–331.

Bentley, J. L. and Friedman, J. H. (1979). Data structures for range searching, *ACM Computing Surveys* **11**(4): 397–409.

Beyer, K. S., Goldstein, J., Ramakrishnan, R. and Shaft, U. (1999). When is "nearest neighbour" meaningful?, *In Proceedings of the 7th International Conference on Database Theory*, Springer-Verlag, London, UK, pp. 217–235.

Beygelzimer, A., Kakade, S. and Langford, J. (2006). Cover trees for nearest neighbour, *In Proceedings of the 23rd International Conference on Machine Learning*, pp. 97–104.

Ciaccia, P., Patella, M. and Zezula, P. (1997). M-tree: An Efficient Access Method for Similarity Search in Metric Spaces, *In Proceedings of the 23rd International Conference on Very Large Data Bases*, pp. 426–435.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, New York, USA.

François, D., Wertz, V. and Verleysen, M. (2007). The concentration of fractional distances, *IEEE Transactions on Knowledge and Data Engineering* **19**(7): 873–886.

Indyk, P. and Motwani, R. (1998). Approximate nearest neighbours: Towards removing the curse of dimensionality, *In Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, ACM, New York, USA, pp. 604–613.

Le, Q. V. and Mikolov, T. (2014). Distributed Representations of Sentences and Documents, *CoRR* **abs/1405.4053**.
URL: *http://arxiv.org/abs/1405.4053*

Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations, *In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013b). Efficient Estimation of Word Representations in Vector Space, *CoRR* **abs/1301.3781**.
URL: *http://arxiv.org/abs/1301.3781*

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J. (2013a). Distributed Representations of Words and Phrases and Their Compositionality, *In Proceedings of the 26th International Conference on Neural Information Processing Systems*, Curran Associates Inc., USA, pp. 3111–3119.

Radovanović, M., Nanopoulos, A. and Ivanović, M. (2010). Hubs in space: Popular nearest neighbours in high-dimensional data, *Journal of Machine Learning Research* **11**: 2487–2531.

Salton, G. and McGill, M. J. (1986). *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, USA.

Appendix A

Conference paper on m_p-dissimilarity

This chapter includes the following conference paper on m_p-dissimilarity where preliminary results were reported. The extended journal version of the paper is presented in Chapter 3.

Aryal, S., Ting, K. M., Haffari, G. and Washio, T. (2014), m_p-dissimilarity: A data-dependent dissimilarity measure, In *Proceedings of the IEEE International conference on data mining (ICDM) 2014*, IEEE, Pages 707-712.

This chapter is a copy of the paper published in the conference proceedings. In order to generate a consistent presentation within the thesis, the format and some notations or symbols used have been changed, minor grammar and spelling mistakes have been corrected, and sections of the published paper have been renumbered.

The original published version of the paper is available at IEEE Xplore Digital Library via `https://doi.org/10.1109/ICDM.2014.33`

m_p-dissimilarity: A data-dependent dissimilarity measure

Sunil Aryal*, Kai Ming Ting†, Gholamreza Haffari* and Takashi Washio‡

*Clayton School of Information Technology, Monash University, Australia
†School of Engineering and Information Technology, Federation University, Australia
‡The Institute of Scientific and Industrial Research, Osaka University, Japan

Abstract:

Nearest neighbour search is a core process in many data-mining algorithms. Finding reliable closest matches of a query in a high-dimensional space remains a challenging task. This is because the effectiveness of many dissimilarity measures, that are based on a geometric model, such as ℓ_p-norm, decreases as the number of dimensions increases.
In this paper, we examine how data distribution can be exploited to measure dissimilarity between two instances and propose a new data-dependent dissimilarity measure called 'm_p-dissimilarity'. Rather than relying on geometric distance, it measures the dissimilarity between two instances in each dimension as a probability mass in a region that encloses the two instances. It deems two instances in a sparse region to be more similar than two instances in a dense region, although these two pairs of instances may have the same geometric distance.
Our empirical results show that the proposed dissimilarity measure indeed provides a reliable nearest neighbour search in high-dimensional spaces, particularly in sparse data. m_p-dissimilarity produced better task specific performance than ℓ_p-norm and cosine distance in classification and information retrieval tasks.

Keywords: Distance measure, ℓ_p-norm, m_p-dissimilarity

A.1 Introduction

In order to make a prediction for a given query, many data-mining algorithms search for the k closest matches or nearest neighbours (kNNs) of the query in a database, and make a prediction based on those kNNs. They use a similarity or dissimilarity measure to find kNNs. Minkwoski distance (also known as ℓ_p-norm) (Deza and Deza, 2009) is a widely-used dissimilarity measure. Although it performs well in many applications, its effectiveness degrades as the number of dimensions increases. In high-dimensional space, data distribution becomes sparse, which makes the concept of distance meaningless: the "*curse of dimensionality*". All pairs of points are almost equidistant for a wide range of data distributions and distance measures (Beyer et al., 1999; Aggarwal et al., 2001), resulting in unreliable closest match that leads to erroneous predictions.

The performance of distance measure depends on the data distribution and task at hand. A distance measure that performs well in one distribution or task may perform poorly in others. A huge variation in performance can be observed when a distance measure is used in different data distributions and tasks. We hypothesize that this variation is because the distance measure computes the dissimilarity between two instances solely based on the geometric positions. The data distribution (i.e., the relative position of the two instances with respect to the rest of the data) is not taken into consideration.

Psychologists have expressed their concerns about the geometric model of dissimilarity measure (Tversky, 1977; Krumhansl, 1978). They have argued that the judged dissimilarity between two instances is influenced by the context of dissimilarity measurement and other instances in proximity. Krumhansl (1978) has suggested a distance-density model of dissimilarity measurements, arguing that two instances in a relatively dense region would be less similar than two instances of equal distance but located in a less dense region. For example, two white persons will be judged as more similar when compared in Africa (where there are fewer white and more black people) than in America (where there are many white people.)

In this paper, we propose a new dissimilarity measure called 'm_p-`dissimilarity`' in which data distribution is the primary factor in measuring dissimilarity between instances. Rather than using a spatial distance in each dimension, m_p-dissimilarity evaluates the dissimilarity between two instances in terms of probability mass in a region covering the two instances in each dimension. The final dissimilarity between the two instances is estimated as a power mean of dissimilarities in each dimension as in ℓ_p-norm. The intuition behind the proposed dissimilarity measure is that two instances are likely to be more dissimilar if there are more instances between and around them in many dimensions. In the proposed data-dependent dissimilarity measure, two instances in a dense region of the distribution are more dissimilar than two instances with the same geometric distance in a sparse region, as suggested by psychologists.

This paper makes the following contributions:

1. It proposes a new data-dependent dissimilarity measure called m_p-dissimilarity.
2. It provides its theoretical basis and interpretation.

3. It compares the performance of m_p-dissimilarity against ℓ_p-norm and cosine distance in moderate-to high-dimensional datasets from text and music domains in classification and information retrieval tasks.

The rest of the paper is organised as follows. Two widely-used geometric distance measures, ℓ_p-norm and cosine distance, are discussed in Section A.2. The proposed data-dependent dissimilarity measure, m_p-dissimilarity, is discussed in Section A.3. Empirical results are provided in Section A.4, followed by conclusions and future work in the last section. Hereafter, we refer m_p-dissimilarity and ℓ_p-norm by m_p and ℓ_p, respectively.

A.2 Measures based on geometric models

A wide range of geometric (proximity-based) dissimilarity measures are used in the literature, which are discussed in Deza and Deza (2009). In this section, we discuss the two most widely-used measures: ℓ_p-norm and cosine distance.

A.2.1 ℓ_p-norm distance

The distance between two M-dimensional vectors $\mathbf{x}$ and $\mathbf{y}$ based on ℓ_p-norm is defined as follows (Deza and Deza, 2009):

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

where $p > 0$, $\| \cdot \|_p$ is the p order norm of a vector, a_i is the ith component of a vector $\mathbf{a}$ and $abs(\cdot)$ is the absolute value. The limit condition is defined as follows:

$$\ell_\infty(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_\infty = \max_i \; abs(x_i - y_i) \tag{A.2}$$

Manhattan distance (ℓ_1), Euclidean distance (ℓ_2) and Chebysev distance (ℓ_∞) are widely-used ℓ_p-norm-based distance functions. Euclidean distance is a popular choice of distance function as it intuitively corresponds to the distance defined in the real three-dimensional world.

A.2.2 Cosine distance

In many high-dimensional problems, data have the same value (0 or any other constant) in many dimensions, creating 'sparseness'. For example, only a few terms in a dictionary appear in each document in a corpus. Many entries of a vector representing a document are zero. ℓ_p-norm is not a good choice of distance measure for such problems. The direction of vectors is more important than their lengths. The angular distance measure (also known as cosine distance) (Deza and Deza, 2009) is a more sensible choice to measure dissimilarity between two documents.

The cosine distance between two vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as follows (Deza and Deza, 2009):

$$\begin{aligned} d_{cos}(\mathbf{x},\mathbf{y}) &= 1 - \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|_2 \times \|\mathbf{y}\|_2} \\ &= 1 - \frac{\sum_{i=1}^{M} x_i \times y_i}{\sqrt{\sum_{i=1}^{M} x_i^2} \times \sqrt{\sum_{i=1}^{M} y_i^2}} \end{aligned} \tag{A.3}$$

Cosine distance has been shown to perform well in many text-mining problems such as text categorisation, text clustering and text retrieval tasks.

A.3 Data-dependent measure

In order to measure dissimilarity between $\mathbf{x}$ and $\mathbf{y}$, instead of using $(x_i - y_i)$ in Eqn A.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 a 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 more dissimilar in dimension i. Using the same power mean formulation as in ℓ_p-norm, the data-dependent dissimilarity measure based on probability mass can be defined as:

$$m_p(\mathbf{x},\mathbf{y}) = \left(\sum_{i=1}^{M} \left(\frac{|R_i(\mathbf{x},\mathbf{y})|}{N} \right)^p \right)^{\frac{1}{p}} \tag{A.4}$$

where $|R_i(\mathbf{x},\mathbf{y})|$ is the data mass in region $R_i(\mathbf{x},\mathbf{y})$, $R_i(\mathbf{x},\mathbf{y}) = [\min(x_i,y_i) - \delta, \max(x_i,y_i) + \delta]$, $\delta \geq 0$ and N is the number of data instances.

An example of $R_i(\mathbf{x},\mathbf{y})$ is shown in Figure A.1. We use $\delta = \frac{\sigma_i}{2}$ (σ_i is the standard deviation of data in dimension i) in this paper.

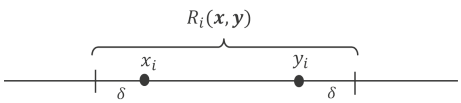


Figure A.1: $R_i(\mathbf{x},\mathbf{y})$

We call the proposed dissimilarity measure $m_p(\mathbf{x},\mathbf{y})$ 'm_p-`dissimilarity`'. This measure captures the essence of the distance-density model proposed by Krumhansl (1978) which suggests that two instances in a sparse region are more similar than two instances with the same distance in a dense region. Although m_p employs the same power mean formulation as ℓ_p, the core calculation is based on mass rather than distance. It signifies the degree of dissimilarity: the higher the measure, the more dissimilar the two instances are, similar to ℓ_p.

The formulation of $m_p(\mathbf{x},\mathbf{y})$ (Eqn A.4) has a probabilistic interpretation. The simplest form of data-dependent dissimilarity measure is to define a region $R(\mathbf{x},\mathbf{y}) \in \mathbb{R}^M$ that encloses $\mathbf{x}$ and $\mathbf{y}$ and estimate the probability of a randomly-selected point $\mathbf{z}$ from the distribution of data, $\phi(\mathbf{x})$, falling in $R(\mathbf{x},\mathbf{y})$, $P(\mathbf{z} \in R(\mathbf{x},\mathbf{y})|\phi(\mathbf{x}))$. Let $R(\mathbf{x},\mathbf{y})$ be centred at $\mathbf{h} = \langle h_1, h_2, \cdots, h_M \rangle$, $h_i = \frac{x_i+y_i}{2}$ and have a length of $R_i(\mathbf{x},\mathbf{y})$ on dimension i. We use

the shorthand R and R_i to represent $R(\mathbf{x}, \mathbf{y})$ and $R_i(\mathbf{x}, \mathbf{y})$, respectively. Assuming that the dimensions are independent, $P(\mathbf{z} \in R|\phi(\mathbf{x}))$ can be approximated as:

$$P(\mathbf{z} \in R|\phi(\mathbf{x})) \approx \prod_{i=1}^{M} P_i(z_i \in R_i|\phi_i(\mathbf{x})) \tag{A.5}$$

where $P_i(z_i \in R_i|\phi_i(\mathbf{x}))$ is the probability of $\mathbf{z}$ falling in R in dimension i.

The approximation using Eqn A.5 is sensitive to outliers. $P(\mathbf{z} \in R|\phi(\mathbf{x}))$ becomes small (or zero) even if only one $P_i(z_i \in R_i|\phi_i(\mathbf{x}))$ is small (or zero). An approximation which is tolerant to outliers can be estimated by replacing the product with a summation (Minka, 2003).

Lemma A.1. *(Minka, 2003) In an outlier model having data distribution $\phi(\mathbf{x})$,*

$$\prod_{i=1}^{M} P_i\left(x_i|\phi_i(\mathbf{x})\right) \propto \sum_{i=1}^{M} P_i\left(x_i|\phi_i(\mathbf{x})\right)$$

Proof. Let us consider a data generation process in which, to sample x_i, a coin with probability of turning heads $(1-\epsilon)$ is flipped. If the coin turns heads, x_i is drawn from the distribution of data in dimension i, $\phi_i(\mathbf{x})$, where the probability of sampling x_i is $P\left(x_i|\phi_i(\mathbf{x})\right)$, otherwise it is drawn from a uniform distribution $1/A$ (A is the area under the domain of $\mathbf{x}$). This model considers outliers as:

$$P_i'\left(x_i|\phi_i(\mathbf{x})\right) = \epsilon/A + (1-\epsilon)P_i\left(x_i|\phi_i(\mathbf{x})\right) \tag{A.6}$$

Using Eqn A.5,

$$\begin{aligned} P'\left(\mathbf{x}|\phi(\mathbf{x})\right) &\approx \prod_{i=1}^{M} P_i'\left(x_i|\phi_i(\mathbf{x})\right) \\ &\approx \prod_{i=1}^{M} \left(\epsilon/A + (1-\epsilon)P_i\left(x_i|\phi_i(\mathbf{x})\right)\right) \end{aligned} \tag{A.7}$$

A Taylor series expansion in $(1-\epsilon)$ leads to:

$$(\epsilon/A)^M + (\epsilon/A)^{M-1}(1-\epsilon)\sum_{i=1}^{M} P_i\left(x_i|\phi_i(\mathbf{x})\right) + O\left((1-\epsilon)^2\right)$$

In the extreme case where there are many outliers, i.e. ϵ is close to 1, only the first two terms matter. The first term is a constant and hence, Lemma A.1 follows.

□

In addition to the above approximation given by Minka (2003), we propose that the chance of x_i being drawn from the outlier model can be further reduced by sampling from $\phi_i(\mathbf{x})^p$, yielding the probability of sampling x_i as $P(x_i|\phi_i(\mathbf{x}))^p$, where $P(\cdot)^p$ is the probability of a random event occurring in p successive trials.

Lemma A.2. *In the outlier model of $\phi(\mathbf{x})$, a more generalised outlier-tolerant approximation can be achieved as:*

$$\prod_{i=1}^{d} P_i\left(\mathbf{x}|\phi(\mathbf{x})\right) \propto \sum_{i=1}^{M} P_i\left(x_i|\phi_i(\mathbf{x})\right)^p$$

Proof. This follows from the proof of Lemma A.1 by simply drawing x_i from $\phi_i(\mathbf{x})^p$ when heads turns up in the coin toss. □

Using Lemma A.2, Eqn A.5 can be expressed as follows:

$$P(\mathbf{z} \in R|\phi(\mathbf{x})) \propto \sum_{i=1}^{M} P_i(z_i \in R_i|\phi_i(\mathbf{x}))^p \tag{A.8}$$

As a result of Eqn A.8 and ignoring the constant which is simply a scaling factor of the dissimilarity, $m_p(\mathbf{x}, \mathbf{y})$ can be estimated as follows:

$$m_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{M} P_i\left(z_i \in R_i|\phi_i(\mathbf{x})\right)^p\right)^{\frac{1}{p}} \tag{A.9}$$

where the outer power of $\frac{1}{p}$ is simply a rescaling of $P(z_i \in R_i|\phi_i(\mathbf{x}))$.

It is important to note the two assumptions made in the above derivation of m_p, i.e., dimension-independence and the outlier model. The assumption of dimension-independence has been applied in data mining, e.g., the Naive Bayes classifier. It has been shown that this assumption does not affect the classification accuracy in many scenarios, even if the assumption is violated.

With the assumption of the outlier model, m_p produces many small $P_i\left(z_i \in R_i(\mathbf{x}, \mathbf{y})|\phi_i(\mathbf{x})\right)$ if $\mathbf{x}$ and $\mathbf{y}$ are similar in dimension i. In other words, instances which are similar are assumed to have small $|R_i|$ in many dimensions. This is not an unrealistic assumption in high-dimensional problems.

In practice, $P_i\left(z_i \in R_i|\phi_i(\mathbf{x})\right)$ can be estimated as:

$$P_i\left(z_i \in R_i|\phi_i(\mathbf{x})\right) = \frac{|R_i|}{N} \tag{A.10}$$

Hence, Eqn A.9 and Eqn A.10 lead to m_p-dissimilarity defined in Eqn A.4. The role of parameter p is similar to that in ℓ_p, i.e., p controls the influence of a dimension by scaling up the degree of dissimilarity.

Figure A.2 shows the contours of dissimilarity measured from an instance at (0.5,0.5) based on m_2 (m_p with $p = 2$) in three different data distributions (uniform, normal and bimodal). In contrast, ℓ_p and cosine distance would produce the same contour in all three distributions. For uniform distribution and infinite samples, m_p will yield the same result as ℓ_p because the data mass in R_i will be proportional to $x_i - y_i$. This is depicted in the first contour plot in Figure A.2 where it approaches the contour plot of ℓ_2.

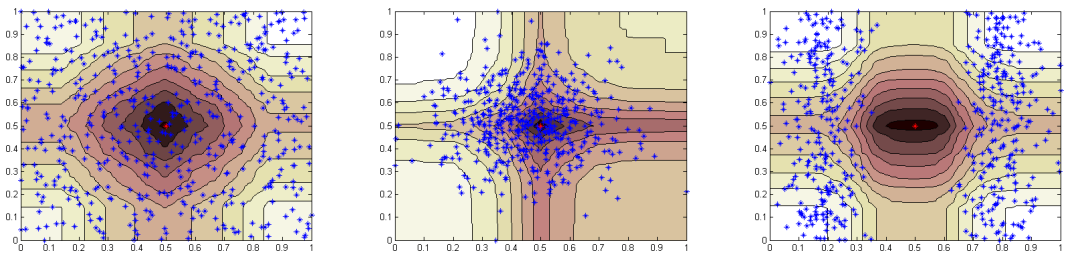


Figure A.2: Contour plots of dissimilarity based on m_2-dissimilarity to the instance at (0.5,0.5) in three different data distributions: uniform, normal and bimodal.

Complexity:

Computationally, m_p is more expensive than ℓ_p as it requires a range search in each dimension. One-dimensional range search can be done in $O(\log N)$ using a binary search tree. Hence, the dissimilarity of a pair of instances can be computed in $O(M \log N)$ against $O(M)$ of ℓ_p. In sparse data, the unique values in each dimension will be much less than N. Hence, the average case runtime will be much less than $O(M \log N)$. In addition, it requires $O(MN)$ time and $O(M \log N)$ space to build and store M binary search trees, respectively.

A.4 Empirical evaluations

This section presents the results of the experiments conducted to evaluate the performance of m_p against ℓ_p and cosine distance in kNN classification and information retrieval.

Eleven datasets from different domains with different sizes ($1000 \leq N \leq 9100$), numbers of dimensions ($188 \leq M \leq 10000$) and numbers of classes ($2 \leq c \leq 52$) were used. All the attributes in the datasets are numeric. Out of 11 datasets used, six are from text mining, two from music classification and retrieval, two from character recognition and the last one is a synthetic dataset from the UCI Machine Learning Repository (Bache and Lichman, 2013). Text data were represented by TFIDF (Salton and Buckley, 1988) weighted 'bag of words' vectors. Other datasets (non-text) were normalised to in the range of [0,1]. The properties and references of the datasets are provided in Table A.1.

We discuss the experimental set-ups and results in classification and information retrieval tasks in the following two subsections.

A.4.1 kNN classification

In the kNN classification context, in order to predict a class label for a test instance $\mathbf{x}$, its k nearest neighbours were searched in the training set based on a dissimilarity measure and the most frequent label of the kNN instances was predicted.

All classification experiments were conducted using a 10-fold cross-validation. We used four settings of p $(2.0, 1.0, 0.5, 0.1)$ in ℓ_p and m_p and two settings of k ($k = 1$ and $k = 10$) for all classifiers. The average accuracy (%) over a 10-fold cross-validation is reported. The

Table A.1: Characteristics of datasets

Name	Reference	N	M	c	Domain
Amazon	Bache and Lichman (2013)	1500	10000	50	text
CNAE	Bache and Lichman (2013)	1080	856	9	text
Reuter	Bache and Lichman (2013)	5000	9288	50	text
R8	Cardoso-Cachopo (2007)	7674	3497	8	text
R52	Cardoso-Cachopo (2007)	9100	7369	52	text
Webkb	Cardoso-Cachopo (2007)	4199	1818	4	text
HBA	Ariyaratne and Zhang (2012)	1500	188	15	music
GTZAN	Tzanetakis and Cook (2002)	1000	230	10	music
Gisette	Bache and Lichman (2013)	7000	5000	2	digit recognition
Mfeat	Bache and Lichman (2013)	2000	649	10	digit recognition
Madelon	Bache and Lichman (2013)	2600	500	2	artificial data

accuracies of two algorithms are considered to be significantly different if their confidence intervals (based on $\pm$ one standard error) do not overlap.

The best average classification accuracy over a 10-fold cross-validation achieved by m_p, ℓ_p and cosine distance in all 11 datasets is presented in Figure A.3. A red dot on the top of the bar indicates that the best performer had significantly better classification accuracy than the other two contenders.

As shown in Figure A.3, m_p produced better classification accuracies than ℓ_p and cosine distance in eight datasets and similar results in the other three datasets. The result is statistically significant in five datasets (CNAE, R8, R52, Webkb and HBA) and not significantly worse in any dataset.

It is interesting to note that m_p produced significantly better classification accuracy than ℓ_p in all six text (sparse) datasets, and better than cosine distance in four out of six. This is because m_p assigns (i) the maximum dissimilarity (of a dimension) if the majority of instances have the same value, which is often the case in sparse text data where term frequencies are zeros in many dimensions; and (ii) the minimum dissimilarity if the value has the least number of training instances in the local neighbourhood.

In terms of p, m_p produced better results with $p = 2$ in eight out of 11 datasets used with the exceptions of Amazon ($p = 0.5$), CNAE ($p = 0.1$) and Madelon ($p = 0.1$). The result with ℓ_p, was mixed: $p = 0.1$ produced better classification results in four datasets, $p = 2$ was better in four, $p = 1$ was better in two, and 0.5 was better in one dataset.

Generally, we observed that $p = 2$ is a reasonable setting in m_p, but we cannot say anything about setting p in ℓ_p as the accuracy varies significantly with p.

A.4.2 Information retrieval

In information retrieval, given a query $\mathbf{q}$, the relevance of a database instance $\mathbf{x}$, $Rel(\mathbf{x}|\mathbf{q})$, was measured using dissimilarity measure d as:

$$Rel(\mathbf{x}|\mathbf{q}) = -d(\mathbf{x}, \mathbf{q}) \tag{A.11}$$

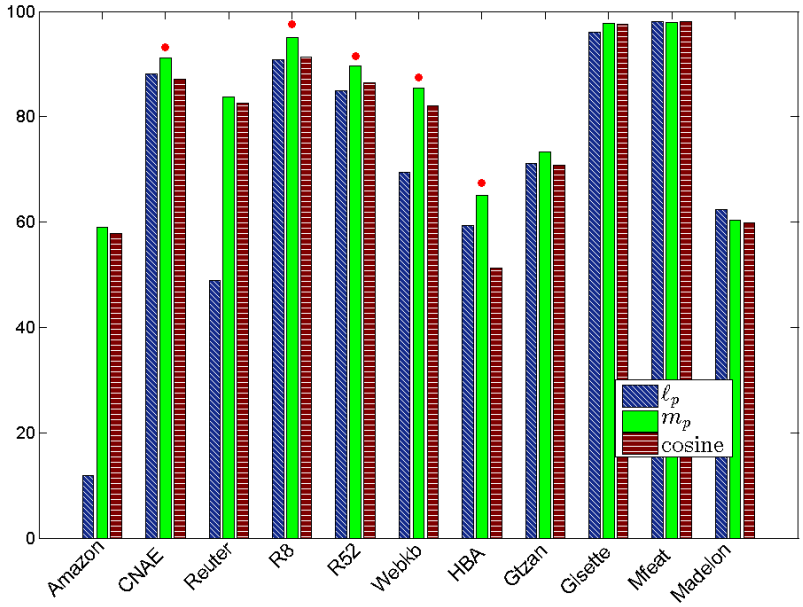


Figure A.3: The best classification accuracies of ℓ_p, m_p and cosine distance in kNN classification. A red dot on the top signifies that the best performer had significantly better classification accuracy than the other two contenders.

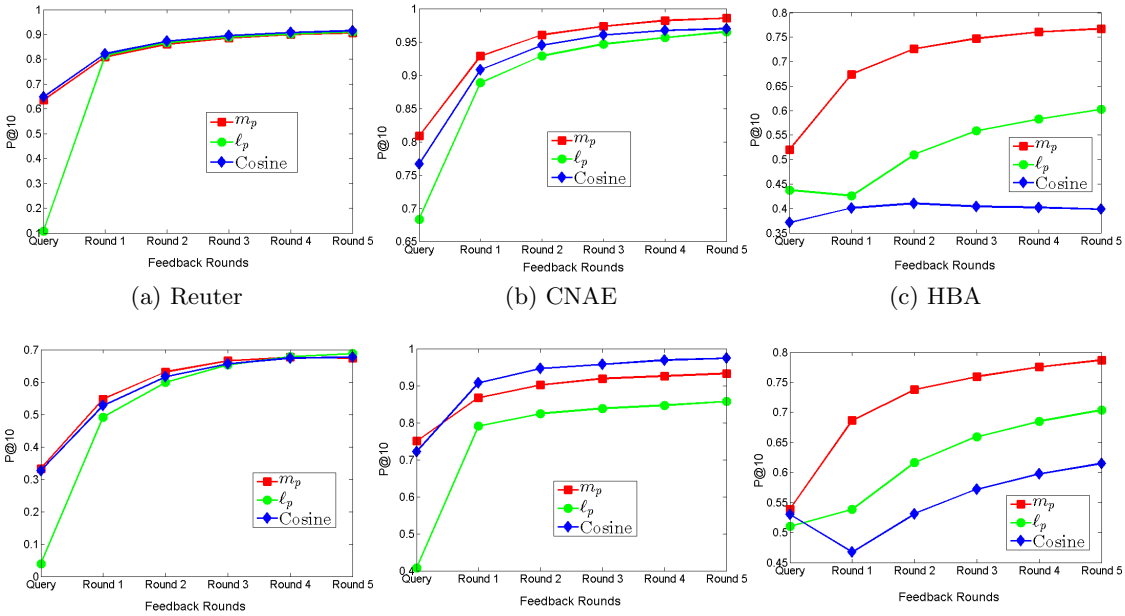
In a relevance feedback process (Rui et al., 1998), a user examines the current retrieval result and provides some 'relevant' and 'irrelevant' examples to the retrieval system. Let $\mathcal{Q} = \mathcal{P} \cup \mathcal{N}$ be a set of feedback instances to the query $\mathbf{q}$ where $\mathcal{P}$ and $\mathcal{N}$ are the sets of positive and negative feedback, respectively. Note that $\mathcal{P}$ includes $\mathbf{q}$. In a relevance feedback round, the relevance score is estimated as follows:

$$Rel(\mathbf{x}|\mathcal{Q}) = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{y}^+ \in \mathcal{P}} Rel(\mathbf{x}|\mathbf{y}^+) - \gamma \frac{1}{|\mathcal{N}|} \sum_{\mathbf{y}^- \in \mathcal{N}} Rel(\mathbf{x}|\mathbf{y}^-) \tag{A.12}$$

where $0 \leq \gamma \leq 1$ is a trade-off parameter for the relative contribution of positive and negative feedback.

We used text and music information retrieval datasets (Reuter, CNAE, HBA, Amazon, R8 and Gtzan) with more than five classes in information retrieval. R52 was not used in information retrieval as the class distribution is heavily skewed and many classes have a few instances, which is not sufficient for query and feedback.

Initially five queries were chosen randomly from each class. For each query, instances from the same class were regarded as relevant and those from the other classes were irrelevant. At each round of feedback, two relevant (instances from the same class) and two irrelevant (instances from the other classes) instances were provided. Five rounds of feedback were conducted for each query. An instance was not used in ranking if it was used as a feedback instance in current or previous feedback rounds. The feedback process was repeated five times with different relevant and irrelevant feedback. This process was repeated 10 times with different queries from each class. The average precision at top 10 (P@10) returned results was reported.



(a) Reuter (b) CNAE (c) HBA

(d) Amazon (e) R8 (f) Gtzan

Figure A.4: Precision at top 10 retrieved results (P@10).

We used the same four settings of p (2.0, 1.0, 0.5, 0.1) and two settings of γ (0,1). Note that when $\gamma = 0$, no negative feedback was needed. The best result achieved at the end of the fifth round of feedback is shown in Figure A.4. m_p produced either better than or similar results to ℓ_p and cosine distance in five datasets. The only exception is in R8 where cosine distance was better than m_p.

It is interesting to note that m_p produced better results with $\gamma = 0$. Its performance degraded in all cases when negative feedback was given. This is because m_p considers the probability of two instances being different and assigns a dissimilarity score according to the distribution of other instances already. Hence, deducting the average relevance score w.r.t irrelevant feedback affects the relevance score of an instance w.r.t $\mathbf{q}$. An instance relevant to a negative feedback may not be equally irrelevant to the query.

On the other hand, ℓ_p-norm significantly improved its performance when negative feedback was given. The performance was improved drastically even in the first round of feedback in the sparse text datasets (Reuter, CNAE, Amazon and R8), whereas this was not the case in the non-sparse music datasets (HBA and Gtzan). In text data, instances are similar in many dimensions with zero values. Initially, in the query round, many irrelevant instances have a high relevance score, as ℓ_p assigns zero distance in many dimensions because of zero frequency. They also have high similarity with negative feedback. Hence, deducting the average relevance w.r.t negative feedback compensates well for the high relevance score given in the first place to irrelevant instances. With negative feedback, ℓ_p produced competitive retrieval results with m_p and cosine distance in the Amazon and Reuter datasets. In the other four datasets, ℓ_p was significantly worse than m_p.

Cosine distance produced significantly worse results in the music datasets. In text retrieval, it produced better results than m_p in subsequent feedback rounds in R8, but

was worse than m_p in CNAE. In Amazon and Reuter, they produced similar retrieval results. Note that cosine distance also produced better results with $\gamma = 1$, i.e., with negative feedback. m_p produced significantly better retrieval performance than ℓ_p and cosine distance in the music (non-sparse) datasets (HBA and Gtzan).

Again, $p = 2$ was better in all six datasets for m_p in information retrieval. For ℓ_p-norm, $p = 1$ or 2 achieved the best retrieval results.

A.5 Conclusions and future work

In this paper, we propose a new dissimilarity measure called 'm_p-dissimilarity' that mainly utilises data distribution in its dissimilarity calculations. It estimates the dissimilarity between two instances in each dimension as a probability of data mass that falls in a region enclosing the two instances. The final dissimilarity between the two instances is estimated as the power mean of 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 difference with the probability mass.

Our empirical evaluations in classification and information retrieval suggest that m_p provides more meaningful closest neighbours than those provided by ℓ_p and cosine distance in high-dimensional space, especially in text datasets where sparsity is a dominant data characteristic.

Potential avenues for future work include investigating an efficient implementation of m_p-dissimilarity, its strengths and limitations along with theoretical analysis and applying m_p to tasks such as clustering, anomaly detection and kernel learning.

Acknowledgements

Sunil Aryal is supported by an Australian Postgraduate Award (APA) at Monash University. This project is partially supported by a grant from the U.S. Air Force Research Laboratory, under agreement# FA2386-13-1-4043, awarded to Kai Ming Ting.

References

Aggarwal, C. C., Hinneburg, A. and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space, *In Proceedings of the International Conference on Database Theory*, Springer, Berlin Heidelberg, pp. 420–434.

Ariyaratne, H. B. and 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.

Bache, K. and Lichman, M. (2013). UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences.
URL: *http://archive.ics.uci.edu/ml*

Beyer, K. S., Goldstein, J., Ramakrishnan, R. and Shaft, U. (1999). When is "nearest neighbour" meaningful?, *In Proceedings of the 7th International Conference on Database Theory*, Springer-Verlag, London, UK, pp. 217–235.

Cardoso-Cachopo, A. (2007). *Improving Methods for Single-label Text Categorization*, PhD thesis, Instituto Superior Tecnico, Technical University of Lisbon, Lisbon, Portugal.

Deza, M. M. and Deza, E. (2009). *Encyclopedia of Distances*, Springer, Berlin Heidelberg.

Krumhansl, C. L. (1978). Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density, *Psychological Review* **85**(5): 445–463.

Minka, T. P. (2003). The 'summation hack' as an outlier model. Microsoft Research. **URL:** *http://research.microsoft.com/en-us/um/people/minka/papers/minka-summation.pdf*

Rui, Y., Huang, T., Ortega, M. and Mehrotra, S. (1998). Relevance feedback: a power tool for interactive content-based image retrieval, *IEEE Transactions on Circuits and Systems for Video Technology* **8**(5): 644–655.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval, *Information Processing and Management* **24**(5): 513–523.

Tversky, A. (1977). Features of Similarity, *Psychological Review* **84**(2): 327–352.

Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals, *IEEE Transactions on Speech and Audio Processing* **10**(5): 293–302.

Appendix B

Conference paper on inter-document similarity

This chapter includes the following conference paper on inter-document similarity measurement which shows that tf-idf-based term weighting with cosine (dis)similarity can be detrimental. The extended version of the paper submitted to a journal is presented in Chapter 5.

Aryal, S., Ting, K. M., Washio, T. and Haffari, G. (2015), Beyond tf-idf and cosine distance in document dissimilarity measures, In *Proceedings of the 11th Asia Information Retrieval Societies Conference (AIRS) 2015*, Springer, Cham, Pages 400-406.

This chapter is a copy of the paper published in the conference proceedings. In order to generate a consistent presentation within the thesis, the format and some notations or symbols used have been changed, minor grammar and spelling mistakes have been corrected, and sections of the published paper have been renumbered.

The original published version of the paper is available at Springer via `https://doi.org/10.1007/978-3-319-28940-3_33`

Beyond tf-idf and cosine distance in document dissimilarity measures

Sunil Aryal*, Kai Ming Ting†, Gholamreza Haffari* and Takashi Washio‡

*Clayton School of Information Technology, Monash University, Australia
†School of Engineering and Information Technology, Federation University, Australia
‡The Institute of Scientific and Industrial Research, Osaka University, Japan

Abstract:

In a vector space model, different types of term-weighting schemes are used to adjust bag-of-words document vectors in order to improve the performance of the most widely-used cosine distance. Although cosine distance with some term-weighting schemes results in more reliable (dis)similarity measures in some datasets, it may not perform well in others because of the underlying assumptions of the term-weighting schemes. In this paper, we argue that the explicit adjustment of bag-of-words document vectors using term weighting is not required if a data-dependent dissimilarity measure called m_p-dissimilarity is used. Our empirical results in document retrieval task reveal that m_p with the simplest binary bag-of-words representation is either better than or competitive with cosine distance with the best performing state-of-the-art term-weighting schemes in four widely-used benchmark document collections.

Keywords: Cosine distance, term weighting, m_p-dissimilarity

B.1 Introduction

Using bag-of-words (Salton and McGill, 1986) vector representation, a document d_i in a collection of N documents ($i = 1, 2, \cdots, N$) is represented by an M-dimensional vector (where M is the number of terms in the dictionary), i.e., $d_i = \langle d_{i1}, d_{i2}, \cdots, d_{iM} \rangle$, where each entry d_{ij} represents the frequency of occurrence of term t_j in d_i. As the most widely-used cosine distance (Salton and McGill, 1986) estimates the dissimilarity of two vectors

Table B.1: Dissimilarity between d_q and other documents in a dataset.

doc	t_1	t_2	t_3	$dist_{cos}$
d_1	5	2	0	0.82
d_2	2	2	0	0.58
d_3	2	0	0	1.00
d_4	1	2	0	0.32
d_5	0	2	1	0.74
d_6	3	2	4	0.93
d_7	1	6	2	0.63
d_q	0	2	0	-

using their geometric positions only, it is important to adjust their positions in the space according to the importance of their terms. Two types of term-weighting factors are used in the literature to estimate the importance of term t_j in document d_i (w_{ij}) (Salton and Buckley, 1988). First, a term frequency (tf)-based factor of t_j in d_i (tf_{ij}) can be estimated in different ways: (a) Binary (bin_tf): $tf_{ij} = 1$ if t_j is in d_i and 0 otherwise; (b) Raw term frequency (Raw_tf): $tf_{ij} = d_{ij}$; and (c) Logarithmic (log_tf): $tf_{ij} = \log(1 + d_{ij})$. Second, the inverse document frequency (idf)-based weighting factor of a term t_j (idf_j) is estimated using the number of documents in the given collection having term t_j (n_j) as $idf_j = \log\left(\frac{N}{n_j}\right)$. Using tf_{ij} and idf_j, a term-weighted document vector of document d_i is represented as having component $w_{ij} = tf_{ij} \times idf_j$. The dissimilarity between document vectors d_1 and d_2 using the cosine distance is estimated as follows:

$$dist_{cos}(d_1, d_2) = 1 - \frac{\sum_j w_{1j} \times w_{2j}}{\sqrt{\sum_j {w_{1j}}^2} \times \sqrt{\sum_j {w_{2j}}^2}} \tag{B.1}$$

It has been shown that the above cosine distance is more meaningful than the traditional ℓ_2-norm in text retrieval (Salton and McGill, 1986; Salton and Buckley, 1988). The only difference between the cosine distance and ℓ_2-norm is that it uses the length normalised vector which is referred to as cosine normalisation in the literature.

The ideas of term weighting and cosine normalisation are based on the following three monotonic assumptions (Zobel and Moffat, 1998): (i) Multiple appearances of a term in a document are no less important than single appearance (the tf assumption). (ii) Rare terms are no less important than frequent terms (the idf assumption). (iii) For the same quantity of term matching, long documents are no more important than short documents (the cosine normalisation assumption).

Although these assumptions appear reasonable, the similarity measure biases toward smaller documents, documents with infrequent terms, and documents with multiple occurrences of terms, which can be disadvantageous in some cases (Polettini, 2004). The cosine distance with term-weighted document vectors may perform well in some datasets or domains where the above assumptions hold, but it may perform worse in other datasets where the assumptions do not hold (see the experimental results in Section B.3).

In the example shown in Table B.1, the dissimilarity between d_q and each of the documents d_1-d_7 using the cosine distance with raw_tf-idf term weighting is provided

in the fourth column. Although d_4 and d_5 have the same occurrences of the common term t_2 as d_q, d_4 is considered to be more similar to d_q than d_5 for no particular reason because of the idf assumption (d_5 is penalised more due to the mismatch in infrequent term t_3). Similarly, d_2 is considered to be more similar to d_q than d_1 because of the cosine normalisation assumption (d_2 is shorter than d_1). Although d_6 has the same occurrences of the common term t_2 as d_q, d_7 is considered to be more similar to d_q than d_6 because of the tf assumption (d_7 has more occurrences of t_2).

Furthermore, in order to use the tf-based weights such as raw_tf and log_tf, the frequency of each term in each document is required. However, in some application domains such as legal and medical, it may not be possible to have the exact term frequencies, due to privacy issues, because it is possible to infer information in the document from its term frequencies (Zhu et al., 2008). Hence, in some domains, only binary representation of documents is available rather than their raw term frequencies.

In this paper, we investigate a dissimilarity measure that does not require the adjustment of bag-of-words vectors and demonstrate that the recently proposed data-dependent dissimilarity measure called m_p-dissimilarity (Aryal et al., 2014b) is one such measure. It uses a similar statistic as that used in idf_j but it is used as the measure of dissimilarity directly rather than for vector adjustment in the space. Our empirical evaluation shows that m_p-dissimilarity with the simplest binary representation performs either better than or competitively with the cosine distance with different term-weighting schemes in document retrieval tasks. Its performance is more consistent across different datasets than that of cosine distance with any term-weighting scheme.

B.2 m_p-dissimilarity in bag-of-words document vectors

In order to measure dissimilarity between two M-dimensional data points x and y, rather than simply relying on the positions of x and y in the space, m_p-dissimilarity (Aryal et al., 2014b) (we refer to it as m_p hereafter) considers the probability data mass in the range $R_j(x, y)$ that encloses x and y in each dimension j. It estimates the final dissimilarity as follows (Aryal et al., 2014b):

$$m_p(x, y) = \left(\frac{1}{M} \sum_{j=1}^{M} \left(\frac{|R_j(x, y)|}{N} \right)^p \right)^{\frac{1}{p}} \tag{B.2}$$

where $|R_j(x, y)|$ is the number of data points falling in $R_j(x, y)$; N is the total number of data points, and $p > 0$ is a parameter.

By simply replacing the geometric distance in each dimension by the probability mass in the range, m_p has been shown to provide more reliable nearest neighbours than ℓ_p-norm in high-dimensional spaces (Aryal et al., 2014b). However, it is very expensive to compute as it requires a range search to determine how many instances fall in each $R_j(x, y)$. Using a binary search tree, one-dimensional range search can be done in $O(\log N)$, resulting in the run-time complexity of $O(M \log N)$ to measure the dissimilarity of a pair of vectors.

In a document collection, only a few terms in the dictionary appear in each document. Many terms do not appear in either of the two documents provided for dissimilarity measurement. Since the absence of a term in both documents does not provide any information about the (dis)similarity of documents, those terms should be ignored. Hence, we make a simple modification in the formulation of m_p shown in Eqn B.2 by considering only those terms that occur in either of the two documents as follows:

$$m_p(d_1, d_2) = \left(\frac{1}{|T_{1,2}|} \sum_{t_j \in T_{1,2}} \left(\frac{|R_j(d_1, d_2)|}{N} \right)^p \right)^{\frac{1}{p}} \tag{B.3}$$

where $|T_{1,2}| = |T_1 \cup T_2|$ (T_i is the set of terms that appear in d_i) is the normalisation term employed to account for different numbers of terms used for any two documents.

Using the simplest binary representation, where each d_{ij} in a document vector d_i has only two values {1,0} indicating whether the term t_j exists in the document d_i, $|R_j(d_1, d_2)|$ can be estimated easily using the total number of documents in the collection (N) and the number of documents where t_j occurs (n_j) as follows:

$$|R_j(d_1, d_2)| = \begin{cases} N & \text{if } d_{1j} \neq d_{2j} \\ n_j & \text{if } d_{1j} = d_{2j} = 1 \end{cases} \tag{B.4}$$

Note that the case where $d_{1j} = d_{2j} = 0$ is not required because Eqn B.3 does not measure dissimilarity of d_1 and d_2 w.r.t a term which does not appear in both d_1 and d_2. n_j can be precomputed for all t_j in pre-processing; thus, $|R_j(d_1, d_2)|$ can be estimated in $O(1)$ resulting in $O(M)$ complexity to compute m_p-dissimilarity of a pair of documents using Eqn B.3 which is equivalent to that of the cosine distance. The pre-processing to compute n_j for all t_j requires $O(MN)$ time and $O(M)$ space complexities. Note that the same complexities are involved in computing the idf factor.

Note that Eqn B.3 does not require the adjustment of the positions of documents in the vector space, because it does not use the absolute positions of two vectors in the dissimilarity measure. It estimates dissimilarity w.r.t each term t_j that appears in both the documents, based on the number of documents having the term (i.e., high dissimilarity if t_j appears in many documents, and low dissimilarity if it appears only in a few documents) and assigns maximal dissimilarity of 1 w.r.t terms that appear in only one of them. Although a similar statistic to that in idf-based weighting is used in the case of matching terms, it is not used to transform vectors in the space but it is used as a measure of (dis)similarity between two documents w.r.t. t_j directly. In the example shown in Table B.1, $m_p(d_q, d_4) = m_p(d_q, d_5)$, $m_p(d_q, d_1) = m_p(d_q, d_2)$ and $m_p(d_q, d_6) = m_p(d_q, d_7)$.

B.3 Empirical evaluation

In this section, we present the empirical results of m_p (using the binary representation) and the cosine distance with the six different term-weighting schemes discussed in Section B.1 (bin_tf, raw_tf and log_tf with and without idf) in relevant document retrieval tasks. Since we want to capture the contrast between two documents with low dissimilarity in a

Table B.2: Datasets

Name	*#docs*	*#terms*	*#cat*
NG20	18,821	5,489	20
R52	9,100	7,379	52
Ohscal	11,162	11,465	10
Wap	1,560	8,460	20

Table B.3: P@10 with average over four datasets in the fourth column ($*$: best, †: second best and ‡: third best).

Contenders	NG20	R52	Ohscal	Wap	Avg.
raw_tf	0.56	$0.85^{\dagger}$	0.53	0.63	0.64
raw_tf-idf	0.71	0.81	0.48	0.64	0.66
log_tf	0.70	$\mathbf{0.87}^{*}$	$\mathbf{0.61}^{*}$	0.64	0.71
log_tf-idf	$\mathbf{0.76}^{*}$	0.81	0.54	$0.67^{\ddagger}$	0.70
bin_tf	0.66	0.84	$0.59^{\dagger}$	0.60	0.67
bin_tf-idf	$0.75^{\dagger}$	0.79	0.56	$0.68^{\dagger}$	0.70
$m_{0.1}$	$0.74^{\ddagger}$	$0.85^{\dagger}$	$\mathbf{0.61}^{*}$	$\mathbf{0.72}^{*}$	**0.73**

few common terms and maximal dissimilarity w.r.t many terms that appear in either of them, $p < 1$ is preferred to amplify the effect of low dissimilarities in the average. Hence, we used $p = 0.1$ for m_p (i.e. $m_{0.1}$) in our experiments[1].

We used four different datasets from four benchmark document collections that are used in the text-mining literature. The data characteristics are provided in Table B.2. NG20[2] is the widely-used 20 Newsgroup dataset and R52[2] is a subset of another widely used Reuters document collection (Cardoso-Cachopo, 2007). Ohscal[3] is a dataset from the Ohsumed patients' medical information collection and Wap[3] is a collection of web pages from Yahoo (Han and Karypis, 2000).

Given a query document d_q, documents in a dataset were ranked in ascending order of their distance/dissimilarity to d_q, and the first k documents were presented as the relevant documents. For performance evaluation, a document was considered to be relevant to d_q if they have the same category label. A good retrieval system returns relevant documents at the top. Hence, the precision in the top 10 (P@10) retrieved documents was used as the performance measure. The same process was repeated for every document in a dataset as a query and the rest of the documents were ranked. The average P@10 over N (the number of documents in a collection) queries of $m_{0.1}$ and cosine with six different term-weighting schemes are provided in Table B.3. Note that all the differences are statistically significant as they are averaged over N (≥ 1560) queries and the standard error is negligible (up to two decimal places) in each case.

[1]The parameter p in m_p has the same role as in the case of traditional ℓ_p-norm. The performance of m_p may be changed slightly using different p values in some datasets. Empirically, we observed that $p = 0.1$ is a reasonably good setting.

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

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

The performance of $m_{0.1}$ was more consistent than the cosine distance with any term-weighting scheme across four datasets (see the average result in the last column in Table B.3). It was among the top three performers in each dataset (and had the best performance in Wap and Ohscal, the second best in R52 and the third best in NG20), whereas none of the term-weighting schemes were among the top three performers in all datasets.

It is interesting to note that the idf-based weighting does not always result in good performance as it produced poor results in R52 and Ohscal with any of the three tf representations. Although log_tf produced the best result in R52 and Ohscal, it did not produce the top three results in the other two datasets. Similarly, log_tf-*idf* produced the best result in NG20 and the third best in Wap but did not produce the top three results in Ohscal and R52. Cosine with bin_tf-*idf* was among the top three performers in two datasets, whereas bin_tf and raw_tf in one dataset each and raw_tf-*idf* did not produce the top three results in any dataset.

B.4 Concluding remarks

Since the cosine distance measures (dis)similarity solely based on the positions of two vectors, it is important to adjust the positions of document vectors in the space w.r.t the importance of the terms in those documents. In the literature, many term-weighting schemes are proposed using the tf and idf factors based on certain assumptions. Although these methods perform well in some datasets when the assumptions hold, they may perform poorly when the assumptions do not hold.

Rather than focusing on researching an effective term-weighting scheme to improve the performance of the cosine distance, this paper opens a different avenue for research by investigating an alternative dissimilarity measure that does not require the adjustment of document vectors using a term-weighting scheme. We show that a data-dependent dissimilarity measure called m_p-dissimilarity is one such effective alternative. It considers (dis)similarity between a pair of documents w.r.t each term based on the number of documents having the term.

Our empirical results of relevant document retrieval tasks show that m_p-dissimilarity with the binary bag-of-words representation produces either better or competitive results in comparison to cosine distance with the state-of-the-art term-weighting schemes.

References

Aryal, S., Ting, K. M., Haffari, G. and Washio, T. (2014b). Mp-dissimilarity: A data dependent dissimilarity measure, *In Proceedings of the IEEE International Conference on Data Mining*, IEEE, pp. 707–712.

Cardoso-Cachopo, A. (2007). *Improving Methods for Single-label Text Categorization*, PhD thesis, Instituto Superior Tecnico, Technical University of Lisbon, Lisbon, Portugal.

Han, E.-H. and 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-Verlag, London, UK, pp. 424–431.

Polettini, N. (2004). The Vector Space Model in Information Retrieval - Term Weighting Problem. University of Trento, Italy.
URL: *https://wiki.eecs.yorku.ca/course_archive/2014-15/W/6339/_media/polettini_information_retrieval.pdf*

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval, *Information Processing and Management* **24**(5): 513–523.

Salton, G. and McGill, M. J. (1986). *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, USA.

Zhu, X., Goldberg, A. B., Rabbat, M. and Nowak, R. (2008). Learning Bigrams from Unigrams, *In Proceedings of ACL-08: HLT*, Association for Computational Linguistics, pp. 656–664.

Zobel, J. and Moffat, A. (1998). Exploring the similarity space, *SIGIR Forum* **32**(1): 18–34.

Vita

Publications arising from this thesis are as follows:

Aryal, S., Ting, K. M., Wells, J. R. and Washio, T. (2014), Improving iForest with relative mass. In *Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. Tainan, Taiwan, May 13-16, 2014. Springer International Publishing, Switzerland. pp. 510-521.

Aryal, S., Ting, K. M., Haffari, G. and Washio, T. (2014), Mp-dissimilarity: A data-dependent dissimilarity measure. In *Proceedings of the IEEE International conference on data mining (ICDM)*. Shenzhen, China, December 14-17, 2014. IEEE Computer Society. pp. 707-712.

Aryal, S., Ting, K. M., Haffari, G. and Washio, T. (2015), Beyond tf-idf and Cosine Distance in Document Dissimilarity Measures. In *Proceedings of the 11th Asia Information Retrieval Societies Conference (AIRS)*. Brisbane, Australia, December 2-4, 2015. Springer International Publishing, Switzerland. pp. 400-406.

Aryal, S., Ting, K. M., Haffari, G. and Washio, T. (2017), Data-dependent dissimilarity measure: an effective alternative to geometric distance measures. *Knowledge and Information Systems (KAIS)*. Springer, London. pp. 1-28. doi: 10.1007/s10115-017-1046-0.

Other publications produced during the PhD candidature which are not directly related to this thesis are as follows:

Aryal, S. and Ting, K. M. (2016), A Generic Ensemble Approach to Estimate Multi-dimensional Likelihood in Bayesian Classifier Learning. *Computational Intelligence*. Volume 32, Issue 3. Wiley Publishing. pp. 458-479.

Aryal, S., Ting, K. M. and Haffari, G. (2016), Revisiting attribute independence assumption in probabilistic unsupervised anomaly detection. In *Proceedings of the Pacific Asia workshop on Intelligence and Security Informatics (PAISI)* at the 20th Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD). Auckland, New Zealand, April 19, 2016. Springer International Publishing, Switzerland. pp. 73-86.

Ting, K. M., Washio, T., Wells, J. R. and Aryal, S. (2017), Defying the Gravity of Learning Curve: A Characteristic of Nearest Neighbour Anomaly Detectors. *Machine Learning*. Volume 106, Issue 1. Springer, USA. pp. 55-91.

Permanent Address: School of Information Technology
Monash University
Clayton Campus
Australia

This thesis was typeset using LaTeX 2_ε[4] by the author.

[4] LaTeX 2_ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this thesis were written by Glenn Maughan and modified by Dean Thompson and David Squire of Monash University.

Aggregated list of references

Achlioptas, D. (2001). Database-friendly random projections, *In Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, ACM, New York, USA, pp. 274–281.

Achtert, E., Hettab, A., Kriegel, H.-P., Schubert, E. and Zimek, A. (2011). Spatial outlier detection: Data, algorithms, visualizations, *In Proceedings of the 12th International Conference on Advances in Spatial and Temporal Databases*, Springer, Berlin Heidelberg, pp. 512–516.

Aggarwal, C. C., Hinneburg, A. and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space, *In Proceedings of the International Conference on Database Theory*, Springer, Berlin Heidelberg, pp. 420–434.

Aha, D. and Kibler, D. (1991). Instance-based learning algorithms, *Machine Learning* **6**: 37–66.

Ariyaratne, H. B. and 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.

Aryal, S., Ting, K. M., Haffari, G. and Washio, T. (2014b). Mp-dissimilarity: A data dependent dissimilarity measure, *In Proceedings of the IEEE International Conference on Data Mining*, IEEE, pp. 707–712.

Aryal, S., Ting, K. M., Haffari, G. and Washio, T. (2015). Beyond tf-idf and cosine distance in document dissimilarity measures, *In Proceedings of the 11th Asia Information Retrieval Societies Conference*, Springer, Cham, pp. 400–406.

Aryal, S., Ting, K. M., Washio, T. and Haffari, G. (2017). Data-dependent dissimilarity measure: an effective alternative to geometric distance measures, *Knowledge and Information Systems* pp. 1–28, doi:10.1007/s10115–017–1046–0.

Aryal, S., Ting, K. M., Wells, J. R. and Washio, T. (2014a). Improving iForest with Relative Mass, *In Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, Cham, pp. 510–521.

Bache, K. and Lichman, M. (2013). UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences.
URL: *http://archive.ics.uci.edu/ml*

Bay, S. D. and Schwabacher, M. (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule, *In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 29–38.

Beckmann, N., Kriegel, H.-P., Schneider, R. and Seeger, B. (1990). The R*-tree: An Efficient and Robust Access Method for Points and Rectangles, *In Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, ACM, New York, USA, pp. 322–331.

Bellet, A., Habrard, A. and Sebban, M. (2013). A survey on metric learning for feature vectors and structured data, *CoRR* **abs/1306.6709**.
URL: *http://arxiv.org/abs/1306.6709*

Bentley, J. L. and Friedman, J. H. (1979). Data structures for range searching, *ACM Computing Surveys* **11**(4): 397–409.

Beyer, K. S., Goldstein, J., Ramakrishnan, R. and Shaft, U. (1999). When is "nearest neighbour" meaningful?, *In Proceedings of the 7th International Conference on Database Theory*, Springer-Verlag, London, UK, pp. 217–235.

Beygelzimer, A., Kakade, S. and Langford, J. (2006). Cover trees for nearest neighbour, *In Proceedings of the 23rd International Conference on Machine Learning*, pp. 97–104.

Black, M. (1952). The identity of indiscernibles, *MIND: A Quarterly Review of Psychology and Philosophy* **61**(242): 153–164.

Boriah, S., Chandola, V. and 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.

Breiman, L. (2001). Random forests, *Machine Learning* **45**(1): 5–32.

Breunig, M. M., Kriegel, H.-P., Ng, R. T. and Sander, J. (2000). LOF: Identifying Density-Based Local Outliers, *In Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 93–104.

Cardoso-Cachopo, A. (2007). *Improving Methods for Single-label Text Categorization*, PhD thesis, Instituto Superior Tecnico, Technical University of Lisbon, Lisbon, Portugal.

Chierichetti, F., Kumar, R., Pandey, S. and Vassilvitskii, S. (2010). Finding the Jaccard Median, *In Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, Society for Industrial and Applied Mathematics, pp. 293–311.

Ciaccia, P., Patella, M. and Zezula, P. (1997). M-tree: An Efficient Access Method for Similarity Search in Metric Spaces, *In Proceedings of the 23rd International Conference on Very Large Data Bases*, pp. 426–435.

Conover, W. J. and Iman, R. L. (1981). Rank transformations as a bridge between parametric and nonparametric statistics, *The American Statistician* **35**(3): 124–129.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, New York, USA.

Debole, F. and Sebastiani, F. (2003). Supervised term weighting for automated text categorization, *In Proceedings of the 2003 ACM Symposium on Applied Computing*, ACM, New York, USA, pp. 784–788.

Deza, M. M. and Deza, E. (2009). *Encyclopedia of Distances*, Springer, Berlin Heidelberg.

Duda, R. O., Hart, P. E. and Stork, D. G. (2000). *Pattern Classification (2nd Edition)*, Wiley-Interscience, New York, USA.

Fernando, T. L. and Webb, G. I. (2017). SimUSF: an efficient and effective similarity measure that is invariant to violations of the interval scale assumption, *Data Mining and Knowledge Discovery* **31**(1): 264–286.

Fodor, I. (2002). A survey of dimension reduction techniques, *Technical Report UCRL-ID-148494*, Lawrence Livermore National Laboratory, University of California, USA.

François, D., Wertz, V. and Verleysen, M. (2007). The concentration of fractional distances, *IEEE Transactions on Knowledge and Data Engineering* **19**(7): 873–886.

Giacinto, G. and Roli, F. (2005). Instance-based relevance feedback for image retrieval, *In Proceedings of the 17th International Conference on Neural Information Processing Systems*, MIT Press, pp. 489–496.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection, *Journal of Machine Learning Research* **3**: 1157–1182.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. (2009). The WEKA Data Mining Software: An Update, *SIGKDD Exploration Newsletter* **11**(1): 10–18.

Han, E.-H. and 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-Verlag, London, UK, pp. 424–431.

Han, J. and Kamber, M. (2006). *Data mining concepts and techniques*, Morgan Kaufmann Publishers, San Francisco, USA.

Han, X., Li, S. and Shen, Z. (2012). A k-NN Method for Large Scale Hierarchical Text Classification at LSHTC3, *In Proceedings of the Workshop on Large Scale Hierarchical Classification at European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pp. 1–12.

He, J., Li, M., Zhang, H.-J., Tong, H. and Zhang, C. (2004). Manifold-ranking based image retrieval, *In Proceedings of the 12th annual ACM international conference on Multimedia*, ACM, New York, USA, pp. 9–16.

Indyk, P. and Motwani, R. (1998). Approximate nearest neighbours: Towards removing the curse of dimensionality, *In Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, ACM, New York, USA, pp. 604–613.

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura, *Bulletin de la Socit Vaudoise des Sciences Naturelles* **37**: 547–579.

Joachims, T. (1997). A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization, *In Proceedings of the Fourteenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, USA, pp. 143–151.

Jolliffe, I. (2005). *Principal component analysis*, Wiley Online Library.

Jones, K. S., Walker, S. and Robertson, S. E. (2000). A probabilistic model of information retrieval: Development and comparative experiments, *Information Processing and Management* **36**(6): 779–808.

Kaski, S. (1998). Dimensionality reduction by random mapping: fast similarity computation for clustering, *In Proceedings of the IEEE World Congress on Computational Intelligence, IEEE International Joint Conference on Neural Networks.*, Vol. 1, pp. 413–418.

Krumhansl, C. L. (1978). Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density, *Psychological Review* **85**(5): 445–463.

Kulis, B. (2013). Metric learning: A survey, *Foundations and Trends in Machine Learning* **5**(4): 287–364.

Lan, M., Tan, C. L., Su, J. and Lu, Y. (2009). Supervised and traditional term weighting methods for automatic text categorization, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31**(4): 721–735.

Le, Q. V. and Mikolov, T. (2014). Distributed Representations of Sentences and Documents, *CoRR* **abs/1405.4053**.
URL: *http://arxiv.org/abs/1405.4053*

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, USA, pp. 296–304.

Liu, F. T., Ting, K. M. and Zhou, Z.-H. (2012). Isolation-based anomaly detection, *ACM Transactions on Knowledge Discovery from Data* **6**(1): 3:1–3:39.

Liu, F., Ting, K. M. and Zhou, Z.-H. (2008). Isolation forest, *In Proceedings of the Eighth IEEE International Conference on Data Mining*, pp. 413–422.

Lundell, J. and 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.

Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations, *In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297.

Mahalanobis, P. C. (1936). On the generalized distance in statistics, *In Proceedings of the National Institute of Sciences of India*, Vol. 2, pp. 49–55.

Manning, C. D., Raghavan, P. and Schtze, H. (2008). *Introduction to Information Retrieval*, Cambridge University Press, New York, USA.

Mansouri, J. and Khademi, M. (2015). Multiplicative distance: a method to alleviate distance instability for high-dimensional data, *Knowledge and Information Systems* **45**(3): 783–805.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013b). Efficient Estimation of Word Representations in Vector Space, *CoRR* **abs/1301.3781**.
URL: *http://arxiv.org/abs/1301.3781*

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J. (2013a). Distributed Representations of Words and Phrases and Their Compositionality, *In Proceedings of the 26th International Conference on Neural Information Processing Systems*, Curran Associates Inc., USA, pp. 3111–3119.

Minka, T. P. (2003). The 'summation hack' as an outlier model. Microsoft Research.
URL: *http://research.microsoft.com/en-us/um/people/minka/papers/minka-summation.pdf*

Molina, L. C., Belanche, L. and Nebot, A. (2002). Feature selection algorithms: A survey and experimental evaluation, *In Proceedings of the IEEE International Conference on Data Mining*, IEEE Computer Society, Washington DC, USA, pp. 306–313.

Paltoglou, G. and Thelwall, M. (2010). A study of information retrieval weighting schemes for sentiment analysis, *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 1386–1395.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* **12**: 2825–2830.

Polettini, N. (2004). The Vector Space Model in Information Retrieval - Term Weighting Problem. University of Trento, Italy.
URL: *https://wiki.eecs.yorku.ca/course_archive/2014-15/W/6339/_media/polettini_information_retrieval.pdf*

Radovanović, M., Nanopoulos, A. and Ivanović, M. (2010). Hubs in space: Popular nearest neighbours in high-dimensional data, *Journal of Machine Learning Research* **11**: 2487–2531.

Roberston, S. and Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond, *Foundations and Trends in Information Retrieval* **3**(4): 333–389.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. and Gatford, M. (1994). Okapi at trec-3, *In Proceedings of the Third Text Retrieval Conference (TREC)*, pp. 109–126.

Rui, Y., Huang, T., Ortega, M. and Mehrotra, S. (1998). Relevance feedback: a power tool for interactive content-based image retrieval, *IEEE Transactions on Circuits and Systems for Video Technology* **8**(5): 644–655.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval, *Information Processing and Management* **24**(5): 513–523.

Salton, G. and McGill, M. J. (1986). *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, USA.

Schleif, F.-M. and Tino, P. (2015). Indefinite proximity learning: A review, *Neural Computation* **27**(10): 2039–2096.

Schneider, P., Bunte, K., Stiekema, H., Hammer, B., Villmann, T. and Biehl, M. (2010). Regularization in matrix relevance learning, *IEEE Transactions on Neural Networks* **21**(5): 831–840.

Shi, T. and Horvath, S. (2006). Unsupervised learning with random forest predictors, *Journal of Computational and Graphical Statistics* **15**(1): 118–138.

Singhal, A., Buckley, C. and Mitra, M. (1996). Pivoted document length normalization, *In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New York, USA, pp. 21–29.

Singhal, A. K. (1997). *Term Weighting Revisited*, PhD thesis, The Faculty of the Graduate School, Cornell University.

Stevens, S. S. (1946). On the theory of scales of measurement, *Science* **103**(2684): 677–680.

Stewart, M. (2015). Metric learning algorithms in Python. GitHub repository. **URL:** *https://github.com/michaelstewart/metric-learn*

Sturges, H. A. (1926). The choice of a class interval, *Journal of the American Statistical Association* **21**(153): 65–66.

Tan, P.-N., Steinbach, M. and Kumar, V. (2006). *Introduction to Data Mining*, Addison-Wesley Longman Publishing Corporation, Boston, USA.

Tanimoto, T. T. (1958). An elementary mathematical theory of classification and prediction, *Technical report*, International Business Machines Corporation, USA.

Ting, K. M., Fernando, T. L. and Webb, G. I. (2013). Mass-based Similarity Measure: An Effective Alternative to Distance-based Similarity Measures, *Technical Report 2013/276*, Clayton School of IT, Monash University, Australia.

Ting, K. M., Washio, T., Wells, J., Liu, F. T. and Aryal, S. (2013b). DEMass: A new density estimator for big data, *Knowledge and Information Systems* **35**(3): 493–524.

Ting, K. M., Zhou, G.-T., Liu, F. and Tan, S. (2013a). Mass estimation, *Machine Learning* **90**(1): 127–160.

Ting, K. M., Zhu, Y., Carman, M., Zhu, Y. and Zhou, Z.-H. (2016). Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure, *In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1205–1214.

Torkkola, K. and Tuv, E. (2005). Ensemble learning with supervised kernels, *In Proceedings of the 16th European Conference on Machine Learning*, Springer-Verlag, Berlin, Heidelberg, pp. 400–411.

Tuytelaars, T., Lampert, C., Blaschko, M. B. and Buntine, W. (2010). Unsupervised object discovery: A comparison, *International Journal of Computer Vision* **88**(2): 284–302.

Tversky, A. (1977). Features of Similarity, *Psychological Review* **84**(2): 327–352.

Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals, *IEEE Transactions on Speech and Audio Processing* **10**(5): 293–302.

Van der Maaten, L., Postma, E. O. and Van den Herik, H. J. (2009). Dimensionality reduction: A comparative review, *Technical Report TiCC TR 2009-005*, Tilburg centre for Creative Computing, Tilburg University, The Netherlands.

Wang, D. and Zhang, H. (2013). Inverse-category-frequency based supervised term weighting schemes for text categorization, *Journal of Information Science and Engineering* **29**(2): 209–225.

Wang, F. and Sun, J. (2015). Survey on distance metric learning and dimensionality reduction in data mining, *Data Mining and Knowledge Discovery* **29**(2): 534–564.

Weinberger, K., Blitzer, J. and Saul, L. (2006). Distance metric learning for large margin nearest neighbour classification, *In Proceedings of the 18th International Conference on Neural Information Processing Systems*, MIT Press, pp. 1473–1480.

Weinberger, K. Q. and Saul, L. K. (2009). Distance metric learning for large margin nearest neighbour classification, *Journal of Machine Learning Research* **10**: 207–244.

Yang, L. (2006). Distance metric learning: A comprehensive survey, *Technical report*, Michigan State University, USA.

Zhou, G.-T., Ting, K. M., Liu, F. T. and Yin, Y. (2012). Relevance feature mapping for content-based multimedia information retrieval, *Pattern Recognition* **45**(4): 1707–1720.

Zhou, Z.-H., Chen, K.-J. and Dai, H.-B. (2006). Enhancing relevance feedback in image retrieval using unlabelled data, *ACM Transactions on Information Systems* **24**(2): 219–244.

Zhou, Z.-H. and Dai, H.-B. (2006). Query-sensitive similarity measure for content-based image retrieval, *In Proceedings of the Sixth International Conference on Data Mining*, pp. 1211–1215.

Zhu, X., Goldberg, A. B., Rabbat, M. and Nowak, R. (2008). Learning Bigrams from Unigrams, *In Proceedings of ACL-08: HLT*, Association for Computational Linguistics, pp. 656–664.

Zobel, J. and Moffat, A. (1998). Exploring the similarity space, *SIGIR Forum* **32**(1): 18–34.