Similarity Measures for Complex Data

by

Le Si Quang

submitted to
Japan Advanced Institute of Science and Technology
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Supervisor: Professor Ho Tu Bao

School of Knowledge Science
Japan Advanced Institute of Science and Technology

September 5, 2005

Abstract

Similarity measure design remains at the core of many important data mining methods. Distance-based methods such as clustering, classification and nearest neighbor searching use distance functions as a key subroutine in their implementation. Clearly, the quality of the resulting distance function significantly affects the success of the corresponding methods in finding results.

Designing similarity measures for complex data is a challenge because of their particular properties: poorness, heterogeneity, and complexity. These properties make measuring similarities between values or integrating similarity scores on attributes into similarities between objects become difficult tasks. Similarity measures for complex data often require particular designs that are suitable for these properties. This dissertation focuses on similarity measures for three data types: categorical data, heterogeneous data, and graph data.

For categorical data, we investigate characteristics and properties of measures borrowed from binary vector measures to see their advantages and disadvantages. We propose an association-based dissimilarity measure that bases on relations between attributes to measure the dissimilarity between categorical values. The main idea is to estimate the dissimilarity between two values from dissimilarities of probability distributions of attributes conditioned on these two values. Intuitively, the greater the dissimilarities of the probability distributions, the greater the dissimilarity between these two values. This measure does not only overcome poorness in values of binary vector-based measures but also boosts accuracy of the classification nearest neighbor in experiments for a large number of real-life databases.

For heterogeneous data, we report existing similarity measures and point out their advantages and disadvantages. We propose an ordered probability-based similarity measure that is based on order relations and probability distributions. The key idea is to estimate the similarity between two attribute values by the probability of picking up a value pair that is less similar than or as similar as. As similarity scores between attribute values are probabilities, they are then integrated into similarities between objects by methods for integrating probabilities. The main advantage of this measure is that it uses up all particular properties of data types meanwhile still keeps homogeneity of similarity scores on different data types. This measure avoids determining common factors/operators for all data types that is the main drawback of the existing measures.

For graph data, we report my survey on similarity measures and point out their advantages and disadvantages. We propose a nonoverlap connected subgraph-based measure that estimates the similarity between two graphs based on three factors: nodes, edges and connectivity of their common subgraphs. The main idea is that the larger the connected common subgraphs of two graphs, the greater the similarity between these two graphs. Using of these factors makes this similarity measure be suitable to 2D chemical structure data. Experiments with clustering and classifications disclose advantages of this measure in practices. The experiments also reveal interesting relations between compound structures and other chemical properties.

In short, we report similarity measures for complex data and point out their disadvantages and advantages in use. We propose three measures that are particularly designed for categorical data, heterogeneous data, and graph data. The merits of these measures are proven by both theories and experiments.

Key words: Similarity measures, complex data, categorical data, heterogeneous data, graph data, binary vectors, condition probability, order relations, graphs, common subgraphs, chemical structures.

Abstract in Japanese

類似度指標の設計は，重要なデータマイニング・アプリケーションの多くにおいて依然として中核をなす課題である．クラスタリング，分類法，最近隣探索などの多くの手法でもその実装における中心的な下部ルーチンとして距離関数を用いることから，距離関数から導出される結果の品質が対応するアプリケーションの探索の成否に大きく影響することは明らかである．

複合・複雑データに対する類似度指標を設計することは，こうしたデータの情報量の低さ，非均質性，そして複雑性といった特殊な性質のために，野心的な試みとならざるをえない．値以外のデータ型に関してその類似度を測定すること，あるいは複数の属性における類似度をまとめてオブジェクト間の類似度として統合することは困難な課題であり，複雑データに対する類似度指標を設計する際には，この特質に対応することが要求される．本論文では，種々の複雑データのうち，カテゴリデータ，タイプ混交データ，グラフデータの３タイプのデータについて，それぞれを対象とする類似度指標を研究範囲とする．

カテゴリデータに関しては，二進ベクトル指標を取り入れた類似度指標についてその特性を調査し，その長所および短所を明らかにした上で，属性間の関係に基いてカテゴリ値の間の差異度を測定する相関ベース差異度指標を新たに提案する．この指標の要点は，任意の属性における二値間の差異度を，他の属性の条件付確率分布の差異度の総計として推定することにある．直感的に確率分布の差が大きければ，比較対象の二つの値の差異度も大きいと見做せる．多数の実世界データベースに対して実施した最近隣分類の実験において，この指標が二進ベクトル指標で表現できる情報量の少なさを克服するだけでなく，精度をも向上させることを示した．

タイプ混交データに関しては，先行する各種類似度指標についてその長所および短所を指摘しつつ，順序関係および確率分布に則った順序確率ベースの類似度指標を提案する．この指標の要点は，各属性における二つの値の類似性の程度を，順序関係においてより近いその他の値のペアの確率によって推定することである．属性値間の類似性の得点を確率として表すことから，確率を統合する様々な手法を適用することで，二つのオブジェクトの類似度を個々の属性における確率を統合したものとして得ることができる．この指標の主な利点は個々のデータ型の特性を利用しつつも，データ型によらず類似度の構成上の等質性を保持できることにある．また，全データ型に共通する共通要素／操作の決定は既存手法の短所であったが，これが不要であることもこの指標の長所である．

グラフデータに関しても，既存指標を調査し，その長所・短所を報告する．グラフデータに大して新たに提案する指標は，互いに素となる連結部分グラフに基づくものである．この指標は２つのグラフ間の類似度を，その共通部分グラフの節，辺，連結の程度という３つの因子に基づいて推定するもので

あり，その要点は，二つのグラフ間で互いに素となる連結部分グラフが大きければ，グラフ同士の類似度も高いと見做すことにある．グラフの節・辺・連結の程度という因子を用いるため，この類似度指標は二次元データとして表現される化学的構造データに適したものとなっている．クラスタリングおよび分類器による実験は，この指標の実用上の利点を実証するとともに，また化合物の構造とその化学的な特性との間の興味深い関係を明らかにすることにも成功している．

以上，本論文では，複雑なデータに関する既存の類似度指標について，それらの指標を使用する際の長所・短所を報告した．また，複雑なデータの中でも特にカテゴリデータ，タイプ混交データ，グラフデータのそれぞれに特化して設計した新たな類似度指標を提案し，これらの指標が理論的に優位性を備えるだけでなく，実験においても優れていることを実証した．

キーワード：類似度指標，複雑データ，カテゴリデータ，タイプ混交データ（不均質データ），グラフデータ，二進ベクトル，条件付確率，順序関係，グラフ，共通部分グラフ，化学構造

Acknowledgments

This work would not have been possible without the support and encouragement of many people. I want to express my gratitude to all of them, even if I cannot mention everyone here.

It was my good fortune to have Professor Tu Bao Ho as my supervisor while at Japan Advanced Institute of Science and Technology (JAIST). He taught me how to be a fruitful researcher, write good papers and, above all, have a good attitude. His thorough scientific approach and unending quest for excellence have been inspirational in the years of my thesis research. I only wish I had listened to his advice more often.

I would like to express my sincere thanks to Professor Yoshiteru Nakamori of Japan Advanced Institute of Science and Technology (JAIST) his guidance and support for my minor research. I am grateful to my former supervisors, Dr. Luong Chi Mai of Vietnam Institute of Information Technology for her advice and encouragement throughout my studies. I would like to thank Professor Judith Steeh at Japan Advanced Institute of Science and Technology (JAIST) and Phan Thi Thu Hang for kindly helping to comment my papers and dissertation.

I sincerely thank all my friends and colleagues who always supported me in times of need. I greatly appreciate to my lab-mates for their contributions in making a wonderful and supportive academic environment. I deeply thank Vietnamese group in Jaist for giving me the warm family environment during the years.

But the life would be also difficult without financial support. I am deeply indebted to the Japanese Ministry of Education, Culture, Sports, Science and Technology for granting me a scholarship, which made possible for my study in Japan. Thanks also go to CREST (Core Research for Evolutional Science and Technology) of JAIST (Japan Science and Technology Corporation), the Foundation for C&C Promotion for providing me with their travel grants which supported me to attend and present my work at some international conferences.

JAIST offered me the greatest learning environment I have ever had - the computing environment, the brilliant faculty, the hard-working students, and the chance to meet famous researchers all over the world. Among the friendly administrators, I owe a great deal to the International Student Section for the kind and constant assistance they provided. Without them, I would certainly have run into much trouble.

Finally, I have saved the best for the last. I wish to express my endless love and gratitude to my family, Mom, Dad, my elder brother, Cau and his wife, Hanh, their children, little lovely niece and nephew, Thuy and Son, and my younger funny brother Vinh, for always being there when I needed them and supporting me through all my years of school. I am especially grateful to my parents for everything they taught me and for all the sacrifices they made in my upbringing.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This dissertation is a report on a study of similarity measures for complex data. This study focused on similarity measures for categorical data, heterogenous data and graph data.

First, definitions of similarity measures and models of similarity measures are presented. Next, we elucidate roles of similarity measures in real life and computer sciences. In addition, challenges and motivations of similarity measures of complex data (categorical, heterogenous and graph data) are addressed. Finally, my contributions and outlines of this dissertation are summarized.

1.1 What are similarity measures?

Deriving from the Latin word "similis" meaning like or resembling, the word "similar" is often used intuitively to compare or relate objects regarding certain common aspects. These aspects are often unspecified or given based on a loose term. Descriptions of similar and similarity are found in dictionaries. For example, from Cambridge Advanced Learner's Dictionary [1]:

> Similar: looking or being almost, but not exactly, the same.

or in Oxford English Dictionary [2]:

> Similar: of the same kind in appearance, character, quantity, without being identical.

Another description of the term similar is given in American Heritage Dictionary of the English Language [3]

> Similar: related in appearance or nature; alike though not identical.

These loose descriptions raise two interesting points. First, similarity among things, people, concepts, etc. is based on equality of certain aspects or their abstraction. Second, identical objects cannot be similar.

Here, something is related to another by being similar, that is similarity is explicitly considered as a probably binary relation. In one hand, two non-identical objects are similar if some unspecified condition on common aspects is fulfilled. On the other hand, similarity between objects is considered as a measure of their likeness. For example, a simple similarity measure is the number of features two objects have in common, thus the greater this value is, the more similar the two objects are. Another example is the similarity between two shapes that exists if a limited set of transformations, e.g., dilation, rotation, expansion, reflection, etc. can be applied to transform one object to the other. These transformations are referred to the similarity between two shapes [4]. In these cases, not only the fact that objects are similar is of interest, but also the quantifiable degree of similarity. Both points of view are useful, and similarity measures are a common way to specify the conditions of a similarity relation.

In the following section, we provide mathematical foundations of popular similarity measures like the distance in metric spaces.

1.2 Models of similarity measures

1.2.1 Common concepts

Let $A_1, \ldots, A_m$ be m attributes and D be a database, $D \subseteq A_1 \times \ldots \times A_m$. Denote $\mathbf{x} = (x_1, \ldots, x_m)$ and $\mathbf{y} = (y_1, \ldots, y_m)$ two objects of D where x_i and y_i are two values of attribute A_i. A similarity measure is defined as follows:

Definition 1 *A similarity measure is a nonnegative function* $sim : D \times D \mapsto R$ *expressing the similarity between two objects* $\mathbf{x}$ *and* $\mathbf{y} \in D$.

This definition clearly includes the dissimilarity or distance between two objects, because an interpretation of the result of the function is not yet specified. Typical interpretations of similarity measures are

Definition 2 (Similarity measure) *A normalized similarity measure* $sim : D \times D \mapsto [0, 1]$ *where* $sim(\mathbf{x}, \mathbf{y}) = 0$ *if the objects are least similar and* $sim(\mathbf{x}, \mathbf{y}) = 1$ *if the objects are most similar or identical.*

Definition 3 (Distance or dissimilarity measure) *A distance or dissimilarity measure* $dist : D \times D \mapsto [0, \infty]$ *or* $dist : D \times D \mapsto [0, maxDistance]$ *where* $dist(\mathbf{x}, \mathbf{y}) = 0$

if the objects are most similar and $dist(\mathbf{x}, \mathbf{y}) = maxDistance$ *if the objects are most dissimilar.*

The two concepts of similarity measures and dissimilarity measure are antithetical and can be easily transformed. For example,

$$sim(\mathbf{x}, \mathbf{y}) = 1 - \frac{dis(\mathbf{x}, \mathbf{y})}{maxDistance}$$

or

$$sim(\mathbf{x}, \mathbf{y}) = \frac{1}{1 + dist(\mathbf{x}, \mathbf{y})}$$

when maximum distance is unknown.

Having defined similarity measures, similarity relations can be easily determined.

Definition 4 (Similarity relation) *A similarity relation on D with respect to a similarity measure "sim" or a dissimilarity measure "dist" and a threshold* θ *is defined as*

$$SimRel(x, y) \equiv sim(x, y) \geq \theta$$

or

$$SimRel(x, y) \equiv dist(x, y) \leq \theta$$

Order relations on data object pairs can be defined from similarity measures as:

Definition 5 (Order relation) *The induction order relation of a similarity measure* s*, denoted* $\preceq_s$*, is defined in the following way:*

$$(\mathbf{x}, \mathbf{y}) \preceq_s (\mathbf{x}', \mathbf{y}') \equiv s(\mathbf{x}, \mathbf{y}) \leq s(\mathbf{x}', \mathbf{y}')$$

Definition 6 (Equivalent order relation) *Two similarity measures* s_1 *and* s_2 *are order equivalent if* $\preceq_{s_1} = \preceq_{s_2}$*.*

1.2.2 Metrics as similarity measures

The most common usage of similarity measures refers to distances in metric space [5] defined as follows

Definition 7 (Metric space) *A metric space is a set S with a global distance function (the metric g) which for every two points* $x, y \in S$*, gives the distance between them as a nonnegative real number* $g(x, y) \in R^+$*. A metric space must also satisfy*

1. $\forall x, y \in S : g(x, y) = 0 \Leftrightarrow x = y$*(Constancy of Self-similarity)*

2. $\forall x, y \in S : g(x, y) = g(y, x)$ *(Symmetry)*

3. $\forall x, y, z \in S : g(x, z) \leq g(x, y) + g(y, x)$ *(Triangular Inequality)*

Definition 8 (Similarity Metric) *A similarity metric is a similarity measure that satisfies all axioms for a metric.*

A typical example for a metric space is the n-dimensional Euclidean space R^n consisting of all points $(x_1, \ldots, x_n) \in R^n$, and the Euclidean metric or distance. As the points in the Euclidean space are represented by n-dimensional vectors, the Euclidean space is also referred to as the vector space or the n-dimensional space. A generalized form of metrics for the Euclidean Space is the Minkowski distance. The Manhattan or City-block distance like the Euclidean Distance is a specialization of the Minkowski distance. There are other distance measures for Euclidean Spaces, some of them satisfy the condition for metrics, e.g., the Chebyshev distance. These metrics for any two points $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ are defined as the following functions

- Minkowski distance

$$dist_p(x, y) = \left[\sum_{i=1}^{n} (|x_i - y_i|)^p \right]^{1/p}$$

  describes a general class of distance measures of various orders $p \in N^+$, also called L_p distance.

- Euclidean distance

$$dist_2(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

  is Minkowski distance with $p = 2$ or L_2

- Manhattan distance

$$dist_1(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

  is Minkowski distance with $p = 1$ or L_1

- Chebyshev distance

$$dist_\infty(x, y) = \max_{i=1}^{n} \{|x_i - y_i|\}$$

  is maximum distance in any dimension, and the upper bound for Minkowski distances of growing order p.

In computer science, the research on similarity for continuous data is mostly based on Minkowski space, especially, Euclidean spaces. Such spaces of a fixed dimensionality n can be indexed efficiently using well-known techniques like R-trees and derivatives [6, 7], Grid Files [8], z-ordering [9], etc., based on the neighborhood preserving nature of these structures. However, the usability of these approaches is limited by the number of dimensions n. This effect is known as the Curse of dimensionality. For instance, mapping objects to a Euclidean space is addressed for complex data objects such as multimedia data through terms of Feature extraction [10, 11], where certainly measurable aspects of objects are used to derive the vector representation direct from one object. A concurrent approach is based on the previously mentioned Multidimensional scaling, which itself is computationally expensive and not suitable for large datasets. Therefore, Faloutsos et al. in [12] described FastMap, also deriving an Euclidean representation of objects for which a distance function or matrix is given, but applying some reasonable simplifications. In [13], Jin et al. used this approach for approximate string matching.

There are several advantages of similarity metrics resulting from the metric axioms, especially when the metrics are used for data processing. Considering the definition of similarity relations described before, the constancy of self-similarity and the symmetry direct translate to a reflexive and symmetric similarity relation. Current data processing is often based on equivalence relations, which are reflexive, symmetric, and transitive. All the optimizations resulting from the former two properties can be applied, if a similarity operation is based on a similarity metric.

1.3 Roles of similarity measures

The importance of similarity in our daily life is often underestimated, but it is clearly pointed out in cognitive sciences, comprising psychological and philosophical aspects. The main inspiration for similarity in computer science is researches in psychology. Moreover, there are parallels of the way information which has to be processed based on similarity by computers and humans.

To achieve the capabilities humans have in processing information from the real world and to bridge communication gaps between men and computer, similarity will have to play a key role. The most important application of similarity is taking place in the human brain every millisecond when incoming sensual information is processed. In 1890 William James stated the following [14]:

> This sense of sameness is the very keel and backbone of our thinking

As Robert Goldstone pointed out in [15] intellectual and cognitive processes have to be based on similarity, because we can only store and perceive varying or incomplete representations of aspects of the world. Of course humans are able to recognize a person they have met before, but for every new meeting this other person and the new perception of her or him have changed more or less. So the human brain has to be able to map the perceived to the stored representation [16]. Besides the identification, where two representations refer to the same object in the real-world, similarity is also applied in other intellectual processes like association, classification, generalization, etc., where representations refer to an abstract relationship or concept based on context-specific commonalities.

In computer science, distance function design remains at the core of many important data mining applications. Many applications such as clustering, classification and nearest neighbor searching use distance functions as a key subroutine in their implementation. Clearly, the quality of the resulting distance function significantly affects the success of the corresponding application in finding results. For many data mining applications, the choice of the distance function is not predefined, but is heuristically chosen. The following subsection shows how important similarity measures in searching, clustering, classification/prediction, and identification, categorization and recognition.

1.3.1 Searching

A basic task in similarity search is to find all objects which are within a certain similarity distance from a query object. Examples can be found in DBSCAN [17], similarity searching in time serial or sequence data data [18, 19, 20, 21, 22], searching in image data [23, 24], nearest neighbor searching [25], disk location [26, 27], tree and graph [28], bioinformatics [29] and similarity search in high dimensions [30].

Definition 9 (similarity range query) *For a query object $q \in O$, and a query range $\epsilon \in R^+$, the result of a similarity range query is defined as*

$$sim_\epsilon(q) = \{o \in DB | d_{sim}(q, o) \leq \epsilon\}$$

According to Definition 9, distance d_{sim} is the main factor to assign objects to query result $sim_\epsilon(q)$. Obviously, different measures lead to different query results.

Another important task in similarity search applications is to find the database object which is most similar to a query object. An example for this query type is to find the most similar protein with known function in a database, given a query protein

with unknown function. This query is called nearest-neighbor query and can be defined informally as the task to find the database object with the smallest similarity distance to the query object. The k-nearest-neighbor query is an extension of the nearest-neighbor query in case a result set with more than one element is desired. An example of such a case is the functional classification of proteins. To improve classification accuracy for nearest-neighbor classification, a protein is not assigned to the functional class of the most similar protein in the database but to the class of the majority of the k most similar proteins.

An important similarity query type is the similarity ranking query which is needed in cases where the exact number of desired results is not known in advance. The idea of this query type is to retrieve iteratively the next closest objects of a query object from the database, starting at the nearest neighbor. This query appears, for example when the user interactively explores the database and retrieves the nearest neighbors of a query object one after another.

In short, searching is strongly based on used similarity measures and the more relevant similarity measures result in the more relevant searching results.

1.3.2 Clustering

Data modeling puts clustering in a historical perspective rooted in mathematics, statistics, and numerical analysis. From a machine learning perspective clusters correspond to hidden pattern. The search for clusters is unsupervised learning, and the resulting system represents a data concept. Therefore, clustering is unsupervised learning of a hidden data concept. Data mining deals with large databases that impose on clustering analysis additional severe computational requirements. These challenges led to the emergence of powerful broadly applicable data mining clustering methods (see surveys in [31, 32]).

There are many definition of clustering and here is the definition of Han and Kamber [33].

Definition 10 (Clustering) *The processing of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters.*

It can be induced from Definition 10 that the main objective of clustering is to group similar objects into clusters. Normally, the similarity relations are usually determined based on similarity measures (see Definition 4).

In practice, there are two main approaches for clustering. Methods of the first approach are to optimize target functions that are based on similarity between objects. For example, partitioning methods such as K-means [34] and K-medoid [35] minimize the target function defined as total distances from objects to cluster representatives. Methods of the second approach use similarity measures as a guide for clustering processes. For example, hierarchical methods of the family introduced by Jambu [36] that includes most hierarchical methods using similarity between objects to estimate similarity between subclusters. Based on that, the closest subclusters are consecutively merged to build a hierarchical clustering tree. For example, similarity between two subclusters are considered as similarity between the closest object pair in the single linkage clustering [37]. In the average linkage clustering [38, 39], similarity between subgraphs are defined as the average similarity of object pairs. In the complete linkage clustering [40], similarity between subgraphs are estimated as the total similarity of object pairs.

In [41], Fu et al. examined effectiveness of different similarity measures on clustering results. The found results show that different similarity measures outperform one another in different cluster quality. In short, clustering results of a database mainly base on similarity measures used to cluster the database.

1.3.3 Classification/Prediction

It is natural and reasonable to predict properties/classes of unknown objects by considering the properties/classes of their closest known objects. For example, to predict biological activities of unknown molecules, we should consider the biological activities of the closest molecules to the unknown molecules [42, 43].

Instance-based classification/prediction methods have emerged as a promising approach to machine learning. Research reports show excellent results on many real-world induction tasks [44, 45, 46, 47, 48, 49]. The basic approach involves storing known cases and their associated properties/classes in the memory. Then when given unknown instances, finding the known cases that are nearest to the unknown instances and using them to predict properties/class of the unknown instances. Many applications of this classification/prediction methods have been introduced, i.e. a weighted nearest neighbor algorithm for learning with symbolic features [50], an adaptive nearest neighbor search for parts acquisition ePortal [51], discretization in lazy learning algorithms [52], efficient search for approximate nearest neighbor in high dimensional spaces [53], computing optimal attribute weight settings for nearest neighbor algorithms [54] and locally weighted learning [55].

Obviously, the results of classification/prediction methods rely on how we choose the closest known objects of an unknown object. The choice is often based on similarity measures to decide which the closest ones of unknown objects are. In short, similarity measures affect classification/prediction methods through choosing the closest objects of unknown objects. This makes similarity measures become key factors in the classification/prediction methods.

1.3.4 Identification, Categorization and Recognition

Identification

One of the classic models for predicting identification performance is the similarity choice model (SCM) proposed by Shepard [56] and Luce [57], whose formal properties have been further investigated by researchers such as Smith [58, 59], Townsend [60], Townsend and Landon [61] and Nosofsky [62, 63]. According to the model, the probability that object $\mathbf{x}$ is identified as object $\mathbf{y}$ is given by

$$P(R_y|S_x) = \frac{b_{\mathbf{y}} sim(\mathbf{x}, \mathbf{y})}{\sum b_{\mathbf{y}} sim(\mathbf{x}, \mathbf{y})}$$

where $b_{\mathbf{y}} (0 \leq b_{\mathbf{y}} \leq 1)$ is often interpreted as the bias for making response $\mathbf{y}$.

Categorization

A classical issue in cognitive psychology is whether the principals of object generalization and similarity that underlie identification performance also underlie categorization performance. Indeed, perhaps the most straightforward view of categorization, formalized in what are known today as exemplar models [64, 65, 66, 67], is that classification of an object is determined by how similar it is to the individual members of alternative categories.

The evidence favoring Category J given presentation of object i is found by summing the (weighted) similarity of object $\mathbf{x}$ to all exemplars of Category J, and then multiplying by the response bias for Category J. This evidence is then divided by the sum of evidences for all categories to predict the conditional probability with which object $\mathbf{x}$ is classified in Category J:

$$P(R_J|S_{\mathbf{x}}) = \frac{b_J \sum_{\mathbf{y} \in C_j} M_{\mathbf{y}} \;\; sim(\mathbf{x}, \mathbf{y})}{\sum_K b_K \sum_{\mathbf{y}' \in C_K} M_{\mathbf{y}'} \;\; sim(\mathbf{x}, \mathbf{y}')}$$

where b_J and $M_{\mathbf{y}}$ denote the Category J response bias and the strength with which object $\mathbf{y}$ is stored in the memory.

Recognition

The models incorporating deterministic multidimensional scaling has also been used to model old-new recognition memory performance [68, 69, 70]. Following investigations of Gillund and Shiffrin [71], and Hintzman [65], the central assumption is that recognition judgments are based on the overall summed similarity of an item to all exemplars stored in memory. This summed similarity gives a measure of overall "familiarity," with higher familiarity values lead to higher recognition probabilities. Specifically, the familiarity for object $\mathbf{x}, F_{\mathbf{x}}$, is given by

$$F_{\mathbf{x}} = \sum_{K} \sum_{\mathbf{y} \in C_K} M_k sim(\mathbf{x}, \mathbf{y})$$

Clearly, similarity measure, $sim(\mathbf{x}, \mathbf{y})$, play an important role in the identification, categorization and recognition tasks. They are the main factors to decide the identification probability of object $\mathbf{x}$ and object $\mathbf{y}$, the conditional probability with which object $\mathbf{x}$ is classified in Category J, and the familiarity for object $\mathbf{x}$.

1.4 Similarity measures for complex data

Similarity measures/distances for standard data have long been studied (i.e. Euclidean distance, Manhattan distance, L_m distance, Jaccard similarity measure [72], Dice similarity measure [73] (see Table 2.3)). However, the similarity measure problem is still opened for complex data such as categorical data, heterogenous data and graph data. Due to their special properties and characteristics, i.e. poor structures, heterogeneity or complex structure, similarity measures for these data have particular requirements that lead to difficulty in estimating similarity of objects described by these data. The following subsections show the difficulties which are also my motivation to choose this research.

1.4.1 Categorical data

Categorical data is one of the most popular data in real-life. We encounter and work with this data frequently in daily life (i.e. colors of a car (Red, Blue, Green), shapes of an object (circle, triangle, square)). Categorical data is comprehensible to humans as the meaning of categorical values is clearly defined to humans. For example, it is clear to humans that what Red, Blue or Green is, or what circle, triangle or square is.

Due to the poor structure of categorical data, estimating similarity between categorical values becomes difficult, i.e. how to estimate the similarity between Green and

Red or that between Green and Blue. In fact, direct relations between categorical values are only identical or nonidentical. Thus, the similarity (dissimilarity) between two categorical values is considered as 1 (0) when they are identical and 0 (1) otherwise. However, similarities/dissimilarities of categorical value pairs are often different. For example, it is obvious that the similarity between Green and Blue is different from that between Green and Red. Thus, the problem of estimating properly similarity between categorical values and objects described by categorical attributes is a challenging task.

1.4.2 Heterogeneous data

With the explosion in volume of databases, the complexity of data objects are now growing up. An object in today databases is often described by many aspects/attributes. For example, the information of one person may be names, age, sex, blood type, photo, educations, employed history, etc. Obviously, this information belongs to different data types, i.e. names (text), age (continuous), and sex (categorical). The data is named heterogenous data.

The difference between similarity measures for homogeneous data and that for heterogeneous data is that each similarity measure for homogeneous data is required to be suitable for only one data type while each similarity measure for heterogenous data is required to be proper for all data types. Obviously, similarity measures for homogeneous data are obviously inapplicable to heterogeneous data as each of them is designed to be suitable for a particular data type. In addition, since measures for homogeneous data have different meaning, they cannot be properly integrated into into similarity measures for heterogeneous data. For example, similarity (dissimilarity) measures for continuous data is suitable for continuous properties, e.g. Euclidean distance, and similarity (dissimilarity) measures for categorical data is suitable for discrete properties of categorical data, e.g. Jaccard, Dice and Rao. However, similarity measures for data objects described by both categorical and continuous data required to be suitable for both continuous and discrete properties. Thus the measures/distances for continuous data or categorical data are inapplicable to databases described by both continuous and categorical data. Besides, it is meaningless when we add the measures/distances into a new similarity measure for the databases described by both continuous and categorical data. The requirement of being suitable for all data types makes the building or designing similarity measures for heterogeneous data become a challenging task.

1.4.3 Graph data

In recent years, there has been an increased interest in developing data mining algorithms that operate on graphs. Such graphs arise naturally in many different application domains including network intrusion, semantic web, behavioral modeling etc.

Due to the complexity of graph data, similarity measures or distances for standard data (i.e. continuous, categorical) are inapplicable to graph data. Besides, the strategy of similarity measures for standard data that are based on similarities between attribute value pairs cannot be applied to graph data as there is no information about corresponding parts between two graphs. For example, for a node u of one graph, there is no information to tell exactly which nodes of other graphs are its corresponding nodes. In case we want to apply the similarity measure strategy for standard data to graph data, the complexity of graph structures causes difficulty in determining the corresponding/common parts of two graphs. In short, the strategy to estimate the similarity for graph data is different from that for standard data. This makes it a challenging task to design similarity measures of graph data.

1.5 Contribution

Being motivated by challenges of similarity measures for complex data, we conduct this dissertation on similarity measures for complex data and obtain some results. The major contributions presented in this dissertation are summarized as follows:

- Reviewing resemblance similarity measures of binary vectors when applying to categorical data. Introducing important properties and characteristics of the similarity measures when applying to categorical data.

- Development of a new dissimilarity measure for categorical data that bases on association relations among attributes [74, 75]. The main idea is to estimate the dissimilarity between two categorical values based on the dissimilarity between their association relations of other categorical values. This measure enriches the dissimilarity between two categorical values in comparing to other resemblance similarity methods where the similarity/dissimilarity between two categorical values is poorly considered as 0 or 1.

- Development of a similarity measure framework for heterogenous data [76]. The proposed framework uses order relations and probability distributions of value pairs to estimate the similarity between two values. Similarity scores of attribute

value pairs are then integrated by statistical methods into similarity between objects. This framework overcomes the main difficulty of similarity measures for heterogeneous data that algebra-based approaches encounter: to determine proper operators/factors for all data types.

- Development of a similarity measure for graph data that bases on nonoverlap connected subgraphs [77]. The main idea is to take the nodes, edges and connectivity of subgraphs into account when estimating the similarity between two graphs. It makes this similarity measure more suitable than other measures when applying to chemical structure data.

- Discovering surprised results about the similarity between clusters of chemical structures and clusters of enzymes in pathways [77].

1.6 Outline

The rest of this dissertation is organized as follows:

- In Chapter 2, we show investigations of resemblance similarity measures for binary vectors with their properties when applying to categorical data. Then we introduce an association-based similarity measure, its characteristics and an algorithm for the use of this measure. At the end of this chapter, we present experiments with classifications for many datasets that prove the merit of these similarity measures.

- In Chapter 3, we mention similarity measures for heterogenous data including Gowda and Diday methods, Minkowski generated metric methods. we show advantage and disadvantage of the similarity measures in discussion parts. After that we introduce the order probability-based method and its particular properties when applied to real data. Lastly, we present experiments with clustering that were carried out to show the merit of the order probability-based similarity measure.

- In Chapter 4, we report on existing similarity measures for graph data including the ϕ distance similarity measure, the measure of Papadopoulos and Manolopoulos, similarity based on the maximal common subgraph, and the edit distance for graphs. We shortly summarize advantage and disadvantage of each similarity measure in the discussion parts. Then we introduce nonoverlap connected

subgraph-based measure and its characteristics and properties. Besides, we also present a heuristic algorithm for this measures.

- In Chapter 5, we present experiments when applying similarity measures for graph data to chemical structure data. we shortly report on similarity measures for 2D chemical structures. Then we present experiments that were carried out on classification and clustering methods to see the merit of the similarity measures. Surprised experiment results of these experiments found are presented in this chapter.

- In Chapter 6, the last chapter, the contributions and achievements of this dissertation are summarized. Dissertation conclusions, suggestions, and opportunities for further search are also presented.

This chapter is finished; prepare for the next chapter.

Chapter 2

Similarity measures for categorical data

Categorical data is one of the most popular data. Its advantage is comprehensibility to human. However, its structure poorness leads to difficulty in measuring the similarity between categorical objects. In this chapter, we investigate common similarity measures for categorical data including binary vector-based similarity measures and introduce an association-based similarity. Experiments on real-life data were carried to show the merit and the properties of these similarity measures.

2.1 Introduction

2.1.1 Categorical data

A categorical variable is one for which the measurement scale consists of a set of categories. For instance, political philosophy may be measured as *liberal*, *moderate*, or *conservative*; smoking status might be measured using categories *never smoked*, *former smoker*, or *current smoker*; and recovery from an operation might be rated as *completely recovered*, *nearly recovered*, *only somewhat recovered* or *not at all recovered*.

Categorical data occurs frequently in the behavioral sciences, public health, ecology, education, and marketing. It even occurs in highly quantitative fields such as engineering sciences or industrial quality control. Such applications often involve subjective evaluation of some characteristics - how soft the touch of a certain fabric is, how good a particular food product tastes, or how easy a worker finds a certain task to be.

Categorical variables whose levels do not have a natural ordering are called nominal. Examples of nominal variables are religious affiliation (categories *Catholic*, *Jewish*,

Protestant, *others*), means of transportation (*automobile*, *bus*, *subway*, *bicycle*, *others*), choice of residence (*house*, *apartment*, *condominium*, *others*), race, gender, and marital status.

Many categorical variables do have ordered levels and are called ordinal. Examples of ordinal variables are size of automobile ($subcompact$, $compact$, $mid-size$, $large$), social class (upper, middle, lower), attitude toward legalization of abortion (strongly disapprove, disapprove, approve, strongly approve), appraisal of company's inventory level ($too\ low, about\ right, too\ high$), and diagnosis of whether a patient has multiple sclerosis ($certain, probable, unlikely, definitely\ not$). Ordinal variables clearly order the categories, but absolute distances between categories are unknown. While we can conclude that a person categorized as *moderate* is more liberal than a person categorized as *conservative*, we cannot give a numerical value for how much more liberal that person is.

Variables are classified as continuous or discrete, according to the number of values they can attain. Actual measurement of all variables occurs in a discrete manner, due to limitations of measuring instruments.

The position of ordinal variables on the quantitative/qualitative classification is fuzzy. They are often treated as qualitative, being analyzed using methods for nominal variables. But in many respects, ordinal variables more closely resemble interval variables than nominal variables. They possess important quantitative features: each level has a greater or smaller magnitude of the characteristic than another level; and, though not often possibly measured, there is usually an underlying continuous variable present. The racial prejudice classification ($none, low, high$) is a crude measurement of an inherently continuous characteristic.

2.1.2 Similarity measures for categorical data

Measuring (dis)similarity of categorical data is a challenging problem because the categorical data does not have any structures. There are only two operators for categorical data: identical and nonidentical operators (see Table 2.1). Thus, there is no way to estimate the difference between categorical values like continuous values. We can only see if they are identical or not identical. For example, we cannot distinguish the difference between *Green* and *Blue* and that between *Green* and *Red*. However, *Green* is somehow more similar to *Blue* than to *Red*.

The most common similarity measures for categorical data are binary vector-based methods [78, 79, 80, 81, 82, 83, 72, 84, 85]. These methods transform each data object into a binary vector where each bit indicates the presence or absence of a possible

Table 2.1: The data types and its possible operators

Attribute	Numerical	Symbolic	
No structure $=,\neq$		Places, color	Nominal (categorical)
Ordinal structure, $\geq,=,\neq$	Age, Temperature, Taste	Rank, Resemblance	Ordinal
Ring structure $+,-,\geq,=,\neq$	Income, length, height		Measurable

attribute value. Then the similarity between two objects is estimated by the similarity between two corresponding binary vectors. These methods are simple, but they have two main drawbacks: (1) the transformation of data objects into binary vectors where the similarity between two categorical values are made into 0 or 1, may leave out many subtleties of the data; (2) they do not take into account the correlations between attributes that typically exist in real-life data and are potentially concerned with the difference among attribute values.

In addition to the binary vector-based methods, similarity measure methods for mixed numerical data [86, 87, 88, 89, 90, 91, 92] can also be applied to categorical data. In [91], Goodall proposed a statistical approach, in which uncommon attribute values make greater contributions to the overall similarity between two objects than common attribute values. The overall similarity is estimated by combining similarities between values pairs by using Lancasters method [93]. Setting aside the statistical approach, algebraic methods have been also proposed [86, 87, 88, 89, 90, 92]. In [86, 87, 88], the similarity between two values of an attribute is based on three factors: (1) the relative position of two values, *position*; (2) the relative sizes of two values without referring to common parts, *span*; (3) the common parts between two values, *content*. Similarly, the sizes of the union (the joint operation $\otimes$) and the intersection (the meet operation $\oplus$) of two attribute values are also taken into account [89, 90, 92]. Subsequently, similarities of all attributes are integrated into the similarity between objects by using Minkowski distance.

In principle, the methods mentioned above can be considered *direct* methods because the dissimilarity between two attribute values is synthesized directly from the values. In [74], Le and Ho presented an *indirect* method to measure the dissimilarity for categorical data. It is called *indirect* in the sense that the dissimilarity between

Table 2.2: Some well-known similarity measures for binary vectors

Measure	Definition	Range	
Russel and Rao	$\frac{a}{M}$	[1,0]	
Kendall, Sokal-Michener	$\frac{a+d}{M}$	[0,1]	S
Rogers and Tanimoto	$\frac{a+d}{M+b+c}$	[1,0]	S
Hamman	$\frac{a+d-b-c}{M}$	[-1,1]	T
Sokal and Sneath, $un_3^{-1}S$	$\frac{b+c}{a+d}$	$[0,\infty]$	S
Jaccard	$\frac{a}{a+b+c}$	[1,0]	T
Kulczynski, T^{-1}	$\frac{1}{2}\left(\frac{a}{a+b}+\frac{a}{a+c}\right)$	[1,0]	
Dice, Czekanowski	$\frac{a}{a+\frac{1}{2}(b+c)}$	[1,0]	T
Sokal and Sneath, un_4	$\frac{a}{a+2(b+c)}$	[1,0]	T
Q_0	$\frac{bc}{ad}$	$[0,\infty]$	
Yule, ω	$\frac{\sqrt{ad}-\sqrt{bc}}{\sqrt{ad}+\sqrt{bc}}$	[-1, 1]	
Yule, Q	$\frac{ad-bc}{ad+bc}$	[-1, 1]	
-bc-	$\frac{4bc}{M^2}$	[0,1]	
Driver& Kroeber, Ochiai	$\frac{a}{\sqrt{(a+b)(a+c)}}$	[0,1]	
Sokan& Sneath, un_5	$\frac{ad}{\sqrt{(a+b)(a+c)(b+d)(d+c)}}$	[0,1]	
Pearson, ϕ	$\frac{ad-bc}{\sqrt{(a+b)(a+c)(b+d)(d+c)}}$	[-1,1]	
Baroni-Urbani, Buser, S^{**}	$\frac{a+\sqrt{ad}}{a+b+c+\sqrt{ad}}$	[0,1]	
Braun-Blanquet	$\frac{a}{\max\{a+b,a+c\}}$	[0,1]	
Simpson	$\frac{a}{\min\{a+b,a+c}\}$	[0,1]	
Michael	$\frac{4(ad-bc)}{(a+d)^2+(b+c)^2}$	[-1,1]	

two values of an attribute is indirectly estimated by using relations between other attributes given these two values. This method is composed of two iterative steps. First, the dissimilarity between two values of an attribute is estimated as the sum of the dissimilarities between conditional probability distributions of other attributes given these two values. Then, the dissimilarity between two data objects is determined as the sum of dissimilarities of their attribute value pairs.

Table 2.3: Some well-known similarity measures for Categorical data

Measure	Definition
Russel and Rao	$\frac{a}{M}$
Kendall, Sokal-Michener	$\frac{M-2m+2a}{M}$
Rogers and Tanimoto	$\frac{M-2m+2a}{M+2m-2a}$
Hamman	$\frac{M-4m+4a}{M}$
Sokal and Sneath, un_3^{-1}	$\frac{2m-2a}{M-2m+2a}$
Jaccard	$\frac{a}{2m-a}$
Kulczynski, T^{-1}; Dice , Czekanowski; Driver& Kroeber; Ochiai; Braun-Blanquet; Simpson	$\frac{a}{m}$
Sokal and Sneath, un_4	$\frac{a}{4m-3a}$
Q_0	$\frac{(m-a)^2}{a(M-2m+a)}$
Yule, ω	$\frac{\sqrt{a(M-2m+a)}-(m-a)}{\sqrt{a(M-2m+a)}+(m-a)}$
Yule, Q	$\frac{a(M-2m+a)-(m-a)^2}{a(M-2m+a)+(m-a)^2}$
-bc-	$\frac{4(m-a)}{M^2}$
Sokan& Sneath, un_5	$\frac{a(M-2m+a)}{m(M-m)}$
Pearson, ϕ	$\frac{a(M-2m+a)-(m-a)^2}{m(M-m)}$
Baroni-Urbani, Buser, S^{**}	$\frac{a+\sqrt{a(M-2m+a)}}{2m-a+\sqrt{a(M-2m+a)}}$
Michael	$\frac{4(a(M-2m+a)-(m-a)^2)}{(M-2m+2a)^2+(m-a)^2}$

2.2 Binary-based similarity measures

2.2.1 Frameworks

A common way to measure dissimilarities between categorical data objects is to transform them into binary vectors. Dissimilarities between data objects are then considered as dissimilarities between corresponding binary vectors.

Let V_A be the set of all possible attribute values:

$$V_A = \bigcup_{i=1}^{m} U_i$$

where U_i is the domain of attribute A_i. Let M be the size of V_A, $M = |V_A|$. Clearly, $M = \sum_{i=1}^{m} |U_i|$

Assuming that the values of V_A are ordered from 1 to M. Each data objects x is now presented by a binary vector with the length M where x_i is 1 if x contains the i^{th} value of V_A and 0 otherwise.

(Dis)similarity measures between binary vectors have been studied for a long time, [85, 79, 84] (see Table 2.2). The principal ideas of these measures are based on the numbers of common and uncommon values. In other words, the similarity between two values are considered 1 if they are identical and 0 otherwise.

Let $\overline{X}$ be the complementary of X and and $XY = \sum_{i=1}^{M} x_i y_i$. Denote

- $a = XY$ - the number of values which X and Y share.
- $b = X\overline{Y}$ - the number of values which X has and Y lacks.
- $c = \overline{X}Y$ - the number of values which X lacks and Y has.
- $d = \overline{XY}$ - the number of values both which X and Y lack.

Obviously, $M = a + b + c + d$.

In [81], Gower and Legendre introduced two families of similarities:

$$T_\theta = \frac{a}{a + \theta(b + c)}$$

and

$$S_\theta = \frac{a + d}{a + d + \theta(b + c)}$$

where $\theta > 0$ to avoid negative values. Many similarity measures belong to these two families (see Table 2.2).

2.2.2 Characteristics

Obviously, $a + b = m, a + c = m$ and $a + b + c + d = M$. Thus, $b = c = m - a$ and $d = M - 2m + a$. The similarity measures for binary vectors in Table 2.2 can be rewritten for categorical data as in Table 2.3.

Theorem 1 *T_θ is an increasing function with a.*

Proof:
Clearly, $a + b = m, a + c = m$. Therefore, $b = c = m - a$. So, T_θ is rewritten as

$$T_\theta = \frac{a}{a + 2\theta(m - a)}$$

We have

$$\begin{aligned} T'_\theta(a) &= \frac{a + 2\theta(m - a) - a(1 - 2\theta)}{[a + 2\theta(m - a)]^2} \\ &= 2\theta m > 0. \end{aligned}$$

Thus, T_θ is an increasing function with a. ■

T_θ starts at the value 0 when $a = 0$ and increases to 1 when $a = m$. When θ is chosen too small (e.g. 0.01) T_θ increases sharply at first and then slows down. We have $T_{.5} = \frac{a}{m}$, a linear function with a. For $\theta < .5$, the graph lines lie above the linear line of $\theta = .5$. For $\theta > .5$ the graph lines lie under the linear line of $\theta = .5$. Figure 2.1 presents the graphs of T_θ when $m = 20$. We can see that the increasing of T_θ depends strongly on θ.

Theorem 2 *S_θ is an increasing function with a.*

Proof:
Clearly, $a + b = m, a + c = m$, and $a + b + c + d = M$. Thus, $b = c = m - a$, and $d = M + a - 2m$. S_θ is rewritten as

$$T_\theta = \frac{2a + M - 2m}{2a + M - 2m + 2\theta(m - a)}$$

We have

$$\begin{aligned} T'_\theta(a) &= \frac{2(2a + M - 2m + 2\theta(m - a)) - (2a + M - 2m)(2 - 2\theta)}{[2a + M - 2m + 2\theta(m - a)]^2} \\ &= \frac{2\theta M}{[2a + M - 2m + 2\theta(m - a)]^2} > 0 \end{aligned}$$

Consequently, S_θ is an increasing function with a. ■

S_θ starts at the value $\frac{M-2m}{M-2m+2\theta m}$ when $a = 0$ and steadily increases to 1 when $a = m$. Figure 2.2 presents the graphs of S_θ when $m = 20$ and $M = 100$.

Theorem 3 *Similarity measure Q_0 is a decreasing function with a when $2m \leq M$.*

Proof:

We have

$$Q_0 = \frac{(m - a)^2}{a(M - 2m + a)}$$

$$\Rightarrow Q'_0(a) = \frac{(a - m)(aM - m(2m - M)}{(a^2(a - 2m + M)^2}$$

So, $Q'_0(a) \leq 0$ when $\frac{m(2m-M)}{M} \leq a \leq m$. Thus, Q_0 is a decreasing function with a on $[0..m]$ when $2m \leq M$. ■

Figure 2.3 shows the graph of Q_0 when $m = 20$ and $M = 100$.

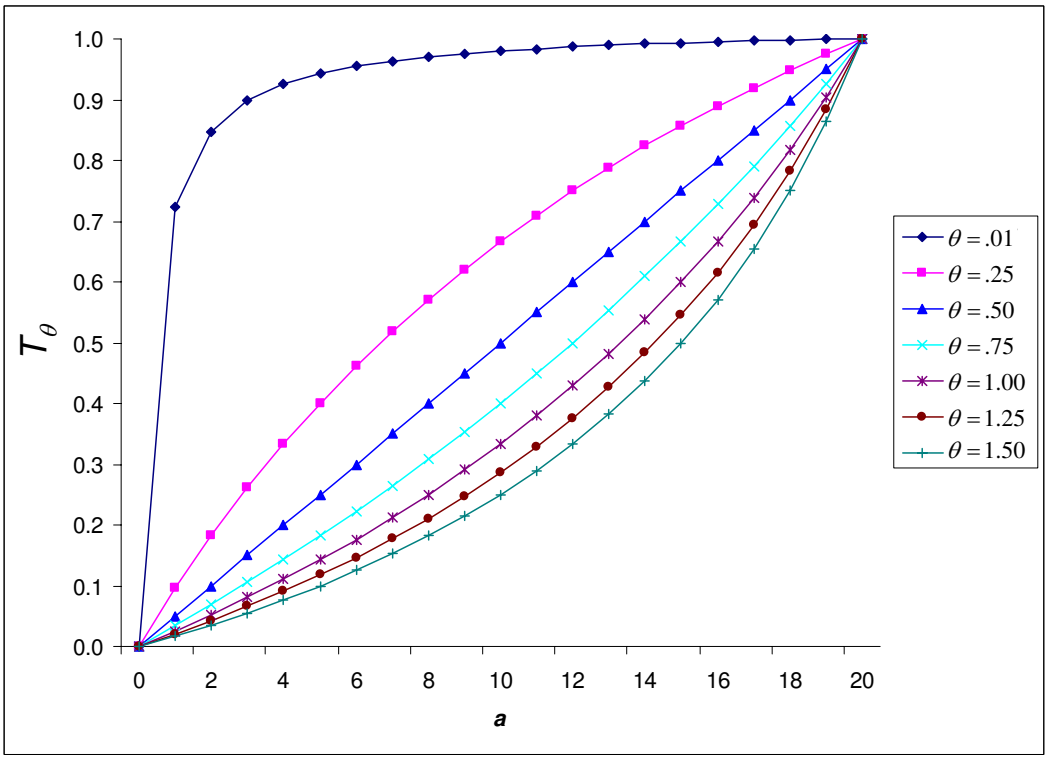


Figure 2.1: The graphs of T_θ with a when $m = 20$

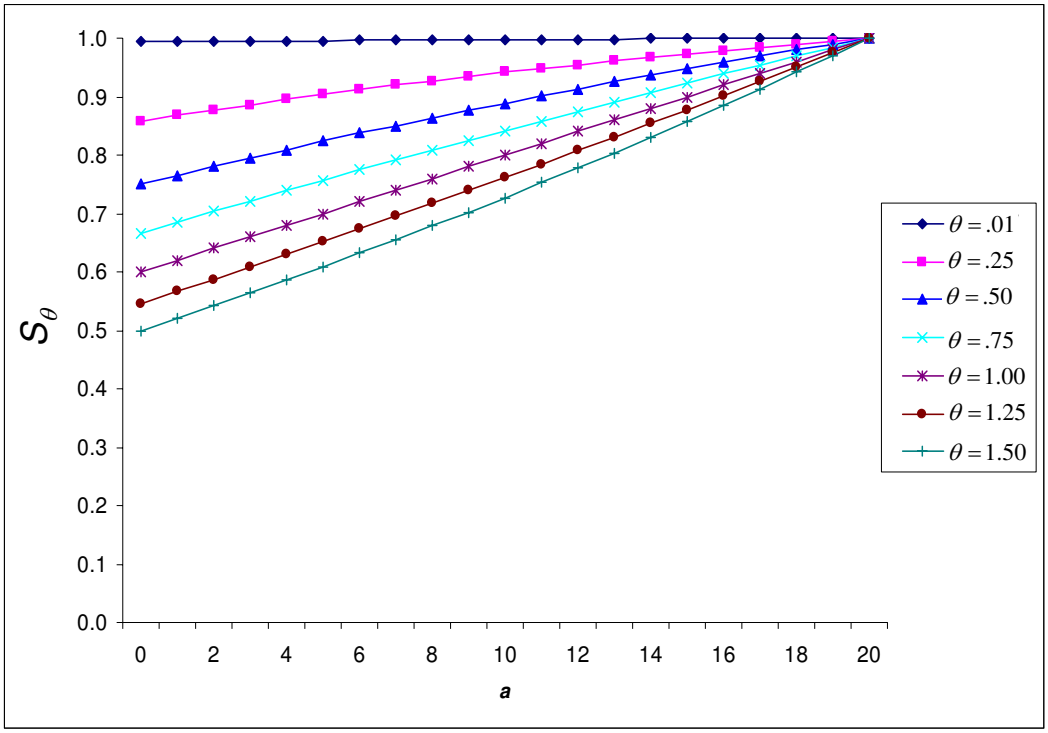


Figure 2.2: The graphs of S_θ with a when $m = 20$ and $M = 100$

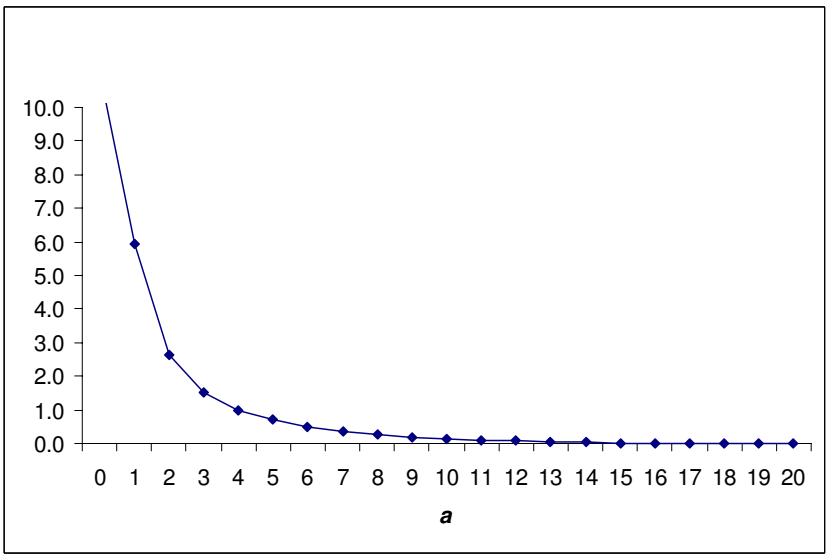


Figure 2.3: The graphs of Q_0 with a when $m = 20$ and $M = 100$

Theorem 4 *Similarity measure Yule(1912), ω, is an increasing function with a when $2m < M$*

Proof:
We have

$$\omega(a) = \frac{\sqrt{a(M-2m+a)} - (m-a)}{\sqrt{a(M-2m+a)} + (m-a)}$$

Therefore,

$$\omega'(a) = \frac{aM - m(2m-M)}{(\sqrt{a(a-2m+M)} - a + m)^2\sqrt{a(a-2m+n)}}$$

Since $2m \leq M$, $S'(a) \geq 0$, $\omega(a)$ is a decreasing function with a.

Theorem 5 *Similarity measure Yule(1927), Q, is an increasing function with a when $2m \leq M$*

Proof:
We have

$$Q(a) = \frac{a(M-2m+a) - (m-a)^2}{a(M-2m+a) + (m-a)^2}$$

$$\Rightarrow Q'(a) = -\frac{2(a^2n - 2am^2 + m^2(2m-n))}{(2a^2 + a(n-4m) + m^2)^2}$$

$Q'(a) > 0$ when $\frac{m*(2m-M)}{M} \leq a \leq m$. It is obviously true when $2m \leq M$. Thus, $Q(a)$ is an increasing function with a when $2m < M$. ■

Theorem 6 *The similarity measure of Baroni-Urbani, Buser (1976), S^{**} is an increasing function with a when $2m \leq M$.*

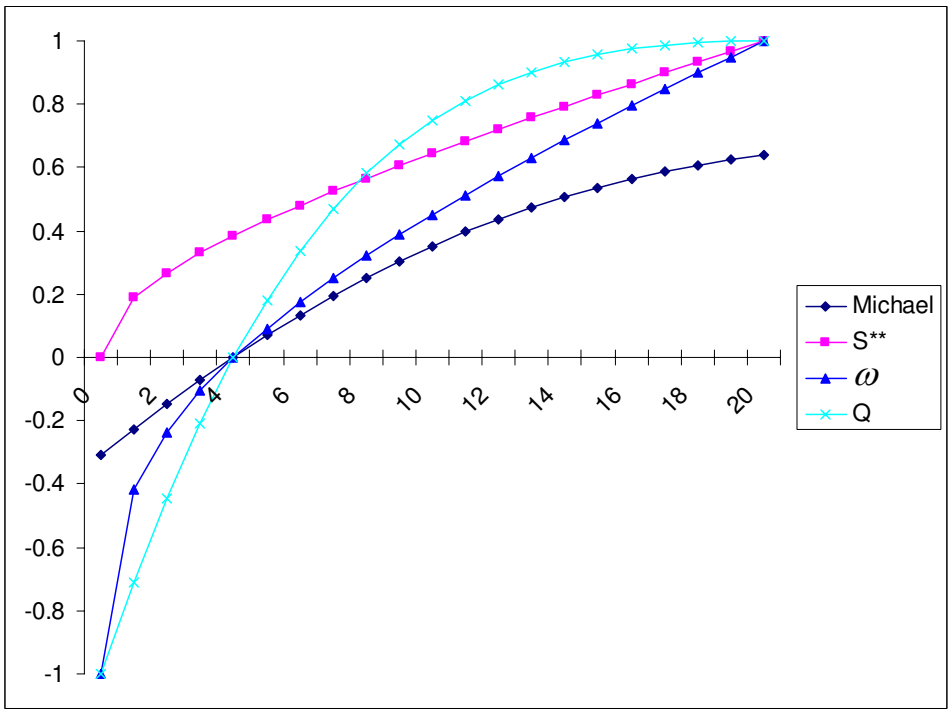


Figure 2.4: The graphs of ω, Q, S^{**}, and Michael's similarity measure with a when $m = 20$ and $M = 100$

Proof
We have

$$S^{**}(a) = \frac{a + \sqrt{a(M - 2m + a)}}{2m - a + \sqrt{a(M - 2m + a)}}$$

$$\Rightarrow S^{**'}(a) = \frac{2m\sqrt{a(a - 2m + M)} + aM - m(2m - M)}{(\sqrt{a(a - 2m + n)} - a + 2m)^2\sqrt{a(a - 2m + n)}}$$

Since $2m \leq M$, $S^{**'}(a) > 0$. Thus, $S^{**}(a)$ is an increasing function with a when $2m \leq M$. ■

Theorem 7 *The similarity measure of Michael is an increasing function with a when* $2m \leq M$.

Proof We have

$$S = \frac{4(a(M - 2m + a) - (m - a)^2)}{a(M - 2m + a) + (m - a)^2}$$

$$\Rightarrow S'(a) = -\frac{8(a^2M - 2am^2 + m^2(2m - M))}{(2a^2 + a(M - 4m) + m^2)^2}$$

So, $S'(a) \geq 0$ when $\frac{m(2m-M)}{M} \leq a \leq m$. $S(a)$ is an increasing function on $[0..a]$ when $2m \leq M$. ■

Figure 2.4 presents graphs of ω, Q, S^{**}, and Michael's similarity measure with a when $m = 20$ and $M = 100$.

Discussion

The advantages of binary-based similarity measures are their simplicity and comprehensibility. Each measure has a clear and simple strategy (i.e., the measure of Russel and Rao is the ratio between values both vectors have and the length of vectors). Different strategies and bias lead to different measures that are suitable for different situations.

However, the measures have a limitation in the range of values (i.e., the only m similarity level). That results in present ability of relations between objects of large databases. For example, the relations between million objects (1 billion relations) can only be categorized into 50 levels if each object is presented by 50 categorical attributes. Besides, transferring categorical data into binary vectors may lose the relations between attribute values that can be taken into account measuring similarity between objects.

2.3 Association probability-based dissimilarity

In [74], Le and Ho presented an *indirect* method to measure the dissimilarity for categorical data. It is called *indirect* in the sense that the dissimilarity between two values of an attribute is indirectly estimated by using relations between other attributes under the condition of giving these two values.

2.3.1 Similarity measure

Let $p(A_j = v_j|A_i = v_i)$ be the conditional probability of $A_j = v_j$ given $A_i = v_i$. More generally, let $cpd(A_j|A_i = v_i)$ be the conditional probability distribution of attribute A_j given that attribute A_i holds value v_i.

The first, and perhaps the most important step, is to estimate the dissimilarity between two values of an attribute.

To motivate the method, consider a data set D with n objects described by two attributes: $Color = \{R, G, B\}$ and $Shape = \{\square, \Diamond, \triangle\}$. Assuming that n is large enough that conditional probabilities $p(A_j = v_j|A_i = v_i)$ and conditional probability distributions $cpd(A_j|A_i = v_i)$ can be approximately estimated from data set D as shown in Table 2.5. Now in considering the relation between the two attributes $Shape$ and $Color$, an important observation is that conditional probability distribution

Table 2.4: The correlation between attributes Color and Shape

	□	◊	△
G	50	40	10
B	40	35	25
R	10	30	60

Table 2.5: The conditional probability of Attribute Color with respect to Attribute Shape

	□	◊	△
G	.50	.40	.10
B	.40	.35	.25
R	.10	.30	.60

$cpd(Shape|Color = Green)$ is closer to $cpd(Shape|Color = Blue)$ than $cpd(Shape|Color = Red)$. It means that the association of attribute Shape with value *Green* is closer to that with value *Blue* than that with value *Red*. On the other hand, the nature of observable dissimilarities also indicates that the dissimilarity between *Green* and *Blue* is somehow smaller than the dissimilarity between *Green* and *Red*. These observations suggest that the dissimilarity between two values of attribute *Color* can be inferred from the conditional probability distributions with respect to attribute *Shape*.

The dissimilarity between two values v_i and v_i' of attribute A_i given that the data set D is composed of m different attributes is defined as following.

Definition 11 *The dissimilarity between two values v_i and v_i' of attribute A_i, denoted by $\phi_{A_i}(v_i, v_i')$, is the sum of dissimilarities between conditional probability distributions of other attributes given that attribute A_i holds values v_i and v_i':*

$$\phi_{A_i}(v_i, v_i') = \sum_{j,j\neq i} \psi(cpd(A_j|A_i = v_i), cpd(A_j|A_i = v_i')) \tag{2.1}$$

where $\psi(.,.)$ is a dissimilarity function for two probability distributions.

Definition 11 means that the dissimilarity between two values v_i and v_i' of attribute A_i is directly proportional to dissimilarities between their respective conditional probability distributions with respect to other attributes. Thus, the great (small) dissimilarity between these conditional probability distributions leads to the great (small) dissimilarity between v_i and v_i'. In other words, when two values leads to the similar (dissimilar) distributions of other attributes, the dissimilarity between two values is low (high).

Up to now, several dissimilarity measures between probability distributions have been proposed [94, 95, 96, 97]. In [74], they used the most popular one, the Kullback-Leiber divergence method [96, 97] (KL)

$$KL(P, P') = \sum_x \left(p(x) \lg \frac{p(x)}{p'(x)} + p'(x) \lg \frac{p'(x)}{p(x)} \right) \tag{2.2}$$

where lg is a logarithm having base 2.

To illustrate the method, the dissimilarities of value pairs (G, B), (B, R), and (R, G) as given in Table 2.5 are computed as follows:

$$\phi_{Color}(G, B) = .5 \lg \frac{.5}{.4} + .4 \lg \frac{.4}{.5} + .4 \lg \frac{.4}{.35} + .35 \lg \frac{.35}{.4} + .1 \lg \frac{.1}{.25} + .25 \lg \frac{.25}{.1} = .24$$

$$\phi_{Color}(B, R) = .4 \lg \frac{.4}{.1} + .1 \lg \frac{.1}{.4} + .35 \lg \frac{.35}{.3} + .3 \lg \frac{.3}{.35} + .25 \lg \frac{.25}{.6} + .6 \lg \frac{.6}{.25} = .1.05$$

$$\phi_{Color}(R, G) = .1 \lg \frac{.1}{.5} + .5 \lg \frac{.5}{.1} + .3 \lg \frac{.3}{.4} + .4 \lg \frac{.4}{.3} + .6 \lg \frac{.6}{.1} + .1 \lg \frac{.1}{.6} = 2.26$$

The dissimilarity of (G, B) is much smaller than that of (G, R) as $cpd(Shape|Color = G)$ is closer to $cpd(Shape|Color = B)$ than $cpd(Shape|Color = R)$ (see Table 2.5).

Having defined the dissimilarity between values of an attribute, dissimilarities of different attributes are combined to the dissimilarity between two data objects.

Definition 12 *The dissimilarity between two data objects* $\mathbf{x}$ *and* $\mathbf{y}$*, denoted by* $\phi(\mathbf{x}, \mathbf{y})$*, is the sum of dissimilarities of their attribute value pairs:*

$$\phi(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} \phi_{A_i}(x_i, y_i) \tag{2.3}$$

Definition 12 means that the smaller the dissimilarities of attribute value pairs of $\mathbf{x}$ and $\mathbf{y}$ are, the smaller the dissimilarity between them.

2.3.2 Algorithm for computing similarities between data objects

In [74], Le and Ho presented a three-step algorithm to measure the dissimilarities of all pairs of data objects of a data set D (see Algorithm 1).

At the first step, all conditional probabilities $p(A_j = v_j | A_i = v_i)$ are estimated from data set D. Then the dissimilarities of the value pairs are computed based on their conditional probabilities $p(A_j = v_j | A_i = v_i)$. The dissimilarities of data object pairs are determined using Equation 2.3 lastly.

Let us now turn our attention to this complexity of the algorithm, given that data set D consists of n objects which are composed of m attributes. At the first step,

Algorithm 1 Algorithm for computing similarities between data objects

1: Estimate all conditional probabilities $p(A_j = v_j|A_i = v_i)$.

2: For any pair of values v_i and v_i' of attribute A_i, compute

$$\phi_{A_i}(v_i, v_i') = \sum_{v_j \in dom(A_j), j \neq i} \left(p(v_j|v_i) \lg \frac{p(v_j|v_i)}{p(v_j|v_i')} + p(v_j|v_i') \lg \frac{p(v_j|v_i')}{p(v_j|v_i)} \right)$$

3: For any data object pairs $(\mathbf{x}\ \mathbf{y})$, compute

$$\phi(\mathbf{x}, \mathbf{y}) = \sum_{i=1} \phi_{A_i}(x_i, y_i)$$

estimating all conditional probabilities $p(A_j = v_j|A_i = v_i)$ is done in $O(nm^2)$ time. Then it takes $O(m_v^3)$ time to compute the dissimilarities of all pairs of attribute values where m_v is the number of attribute values. Finally, all dissimilarities between data objects are determined in $O(n^2m)$ time. Overall, the complexity of the algorithm is $O(nm^2) + O(m_v^3) + O(n^2m) = O(n^2m)$ as m and m_v are typically smaller than n.

2.3.3 Characteristics

Propostion 1 *For any data object pair* $(\mathbf{x}, \mathbf{y})$*, it holds true that:*

1. $\phi(\mathbf{x}, \mathbf{y}) \geq 0$
2. $\phi(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{y}, \mathbf{x})$
3. $\phi(\mathbf{x}, \mathbf{x}) = 0$

Proof

1. $\phi(\mathbf{x}, \mathbf{y}) \geq 0$: Since KL dissimilarity between two probability distributions is non-negative, the dissimilarity of two values x_i and y_i is non-negative

$$\phi_{A_i}(x_i, y_i) = \sum_{j=1, i \neq j}^{m} KL(cpd(A_j|A_i = x_i), cpd(A_j|A_i = y_i)) \geq 0.$$

This implies that

$$\phi(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} \phi_{A_i}(x_i, y_i) \geq 0.$$ ■

2. $\phi(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{y}, \mathbf{x})$: Since KL dissimilarity between two probability distributions is symmetric, the dissimilarity between two values x_i and y_i is also symmetric

$$\begin{aligned}\phi_{A_i}(x_i, y_i) &= \sum_{j=1}^{m} KL(cpd(A_j|A_i = x_i), cpd(A_j|A_i = y_i)) \\ &= \sum_{j=1}^{m} KL(cpd(A_j|A_i = y_i), cpd(A_j|A_i = x_i)) = \phi_{A_i}(y_i, x_i)\end{aligned}$$

It means that

$$\phi(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} \phi_{A_i}(x_i, y_i) = \sum_{i=1}^{m} \phi_{A_i}(y_i, x_i) = \phi(\mathbf{y}, \mathbf{x}). \quad \blacksquare$$

3. $\phi(\mathbf{x}, \mathbf{x}) = 0$: Since KL dissimilarity between two identical probability distributions is equal to 0, the dissimilarity between two identical values is equal to 0.

$$\phi_{A_i}(x_i, x_i) = \sum_{j=1}^{m} KL(cpd(A_j|A_i = x_i), cpd(A_j|A_i = x_i)) = 0$$

It means that

$$\phi(\mathbf{x}, \mathbf{x}) = \sum_{i=1}^{m} \phi_{A_i}(x_i, x_i) = 0. \quad \blacksquare$$

Propostion 2 *The dissimilarity between two values v_i and v_i' of attribute A_i is zero if and only if the conditional probability distributions of other attributes, given that attribute A_i holds values v_i and v_i', are identical.*

$$\phi_v(v_i, v_i') = 0 \Leftrightarrow cpd(A_j|A_i = v_i) \equiv cpd(A_j|A_i = v_i') \text{ for } j = 1 \ldots m, j \neq i$$

Proof:
Since KL dissimilarity between two probability distributions is non-negative, and equal to 0 if and only if the distributions are identical, the dissimilarity between two values v_i and v_i' is equal to 0 if and only if the conditional probability distributions of other attributes when A_i holds values v_i and v_i' are identical. It implies that

$$\Rightarrow \phi_{A_i}(v_i, v_i') = \sum_{j=1, j\neq i}^{m} KL(cpd(A_j|A_i = v_i), cpd(A_j|A_i = v_i')) = 0$$

which is equivalent to

$$cpd(A_j|A_i = v_i) \equiv cpd(A_j|A_i = v_i') \text{ for } j = 1 \ldots m, j \neq i. \quad \blacksquare$$

Propostion 3 *If all attribute pairs are independent, dissimilarities between data objects are all equal to zero.*

Proof:
Since A_i and A_j are independent for all i and j,

$$P(A_j = v_j|A_i = v_i) = P(A_j = v_j) = p(A_j = v_j|A_i = v_i'), \ \ \forall v_i, v_j, v_i'.$$

It means that $cpd(A_j|A_i = v_i)$ and $cpd(A_j|A_i = v_i')$ is identical. It leads to

$$KL(cpd(A_j|A_i = x_i), cpd(A_j|A_i = y_i)) = 0, \ \ \forall x_i, y_i$$

and therefore

$$\phi_{A_i}(x_i, y_i) = \sum_j KL(cpd(A_j|A_i = x_i), cpd(A_j|A_i = y_i)) = 0$$

That is equivalent to

$$\phi(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} \phi_{A_i}(x_i, y_i) = 0, \ \ \forall \mathbf{x}, \mathbf{y}.$$ ■

Discussion

The advantage of using association relations between attribute values to estimate the similarity between values is to enrich the relations between categorical values. Different from 0 or 1 as binary-based measures, the similarity score between two values is a real number. When attributes are not independent, the dissimilarity between association values presents better for the relation between categorical values than the dissimilarity based on the identity or nonidentity as used in binary-based measures.

However, the disadvantage of this measure is that it is applicable only to data sets whose attributes depend on each other. This limits the applications of this measure to real-life databases.

2.4 Evaluations

This section presents experiments to show the merit of the association-based dissimilarity measure when applied to real data. To this end, four experiments are carried out: The first one is to show the variance in values of the association-based dissimilarity measure in comparing with binary-based dissimilarity measures. The second

experiment analyzes the dependency between attributes of real data sets to investigate its impact to the association-based measure. The third experiment compares the association-based measure with the most popular similarity measures, similarity measures presented in Table 2.2, and Goodall [91], by combining them with the popular distance-based data mining method, nearest neighbor classifier (NN)[44]. The last one analyzes time consumption when applied to large databases

2.4.1 Real-life data sets

30 diverse data sets from UCI [98], for which numerical attributes are automatically discretized using the data mining system CBA [99], are used in these experiments. The large number of data sets helps to avoid bias of data selections. Details of these data sets can be found in Table 2.8.

2.4.2 The first experiment: Evaluation of variance

Tables 2.6 and 2.7 present the dissimilarity between objects described by two attributes *color* and *shape* of the association-based dissimilarity measure and the binary-based dissimilarity measures. It is clear that the dissimilarity between objects of the association-based dissimilarity measure varies and is different from object pairs to object pairs. For example, the dissimilarity between $(R, \square)$ and $(R, \Diamond)$ is .15 while the dissimilarity between $(R, \square)$ and $(G, \square)$ is .9. However, there are only 3 dissimilarity levels of the binary-based similarity measures: 0, 1, and 2. For example, the dissimilarity between $(R, \square)$ and $(R, \Diamond)$ is the same as the dissimilarity between $(R, \square)$ and $(G, \square)$, 1.

2.4.3 The second experiment: Dependency analysis

Methodology

For each data set D, the dependency between attributes is estimated by a dependency factor $\rho(D)$ which is the proportion of the number of dependent attribute pairs and the total number of attribute pairs

$$\rho(D) = \frac{|\{(A_i, A_j) : A_i \text{ and } A_j \text{ are dependent}\}|}{m(m-1)}$$

where $\rho(D)$ is directly proportional to the dependency between attributes of D. Thus, $\rho(D)$ is 100% when all attribute pairs are dependent and 0% when they are all independent.

Table 2.6: binary-based dissimilarity scores

		G □	G ◊	G △	B □	B ◊	B △	R □	R ◊	R △
G	□	0	1	1	1	2	2	1	2	2
G	◊	1	0	1	2	1	2	2	1	2
G	△	1	1	0	2	2	1	2	2	1
B	□	1	2	2	0	1	1	1	2	2
B	◊	2	1	2	1	0	1	2	1	2
B	△	2	2	1	1	1	0	2	2	1
R	□	1	2	2	1	2	2	0	1	1
R	◊	2	1	2	2	1	2	1	0	1
R	△	2	2	1	2	2	1	1	1	0

Table 2.7: Association-based dissimilarity scores

		G □	G ◊	G △	B □	B ◊	B △	R □	R ◊	R △
G	□	0	0.35	2.38	0.24	0.59	2.62	2.26	2.61	4.65
G	◊	0.35	0	0.93	0.59	0.24	1.17	2.61	2.26	3.19
G	△	2.38	0.93	0	2.62	1.17	0.24	4.65	3.19	2.26
B	□	0.24	0.59	2.62	0	0.35	2.38	1.05	1.4	3.44
B	◊	0.59	0.24	1.17	0.35	0	0.93	1.4	1.05	1.98
B	△	2.62	1.17	0.24	2.38	0.93	0	3.44	1.98	1.05
R	□	2.26	2.61	4.65	1.05	1.4	3.44	0	0.35	2.38
R	◊	2.61	2.26	3.19	1.4	1.05	1.98	0.35	0	0.93
R	△	4.65	3.19	2.26	3.44	1.98	1.05	2.38	0.93	0

To estimate the dependency of two attributes, we used the χ^2 test with a 95% significance level.

Experiment results and conclusions

Experiment results are presented in Table 2.8, including:

- Data set information: name of the data set (name), number of objects (n), number of attributes (m), number of attribute values (m_v).
- Dependency factors $\rho(D)$.

As can be seen from Table 2.8, for most of all data sets, attributes are strongly dependent on each other. In particular, there are 14 data sets whose dependency factors are greater than 90%, and only one data set whose dependency factor is less than 50%. This proves the experimental applicability of the association measure to real data.

2.4.4 The third experiment: Analyzing with NN

In [84], Batagel and Bren showed that most of the similarity measures presented in Table 2.2 are order equivalent. It means that the closest objects of an object are identical with respect to any of the measures. It implies that NN produces the same accuracy when using any of these similarity measures.

Nearest neighbor classification

The nearest neighbor algorithm [44] is a supervised learning algorithm that simply retains the entire training set during learning. During execution, the new input vector is compared to each instance in the training set. The class of the instance that is most similar to the new vector (using some distance function) is used as the predicted output class. The nearest neighbor algorithm has several strengths when compared to many other learning models:

- It learns very quickly ($O(n)$ for a training set of n instances).
- It is guaranteed to learn a consistent training set (i.e., one in which there are no instances with the same input vector and different outputs) and will not get stuck in local minima.
- It is intuitive and easy to understand, which facilitates implementation and modification.

Table 2.8: Database information and attribute independence

	Name	Size n	No. Atts m	No.Vals M	Ind. $\rho(.)$
1	allbpCleand	2800	30	70	77
2	anneal	898	39	100	58
3	breast	699	11	31	80
4	bridges	106	13	199	100
5	cleve	303	14	31	71
6	crx	690	16	60	98
7	diabetes	768	9	17	57
8	flare	1066	13	42	80
9	german	1000	21	63	59
10	glass	214	10	22	61
11	heart	270	14	22	54
12	hepatitis	155	20	51	81
13	hypo	3163	26	61	92
14	iris	151	5	12	100
15	krvskp	3196	37	73	93
16	lymphography	148	19	59	92
17	monks	432	7	17	0
18	mushroom	8124	23	117	91
19	pima	768	9	17	57
20	post-operative	90	9	24	82
21	primary-tumor	339	18	42	92
22	promoters	106	58	228	86
23	sick	2800	30	66	78
24	splice	3190	61	296	100
25	ttt	958	10	27	94
26	vehicle	846	19	71	99
27	vote	435	17	48	100
28	waveform	5000	22	106	82
29	wine	178	14	37	97
30	zoo	101	18	136	92

- It provides good generalization accuracy on many applications. For example, see [100].

The nearest neighbor paper states that any distance function can be used to determine how close one instance is to another. The author also mentioned the kNN rule, in which the majority class of the k closest neighbors is used for classification. This reduces susceptibility to noise in some cases but may also result in lower accuracy in others. Experiments in this dissertation use NN with value $k = 1$.

Since kNN uses dissimilarity measures to decide the closest instances of one instance, the dissimilarity measures strongly effect on the quality of kNN. Thus the higher accuracy of kNN somehow means the more proper similarity measures. That is why we choose kNN to validate similarity measures.

Validation method

In a 10-by-10 cross-validation study, the ordering of examples in a data set is randomized 10 times and a separate 10-way study is conducted for each of the ten random orderings. Such 10-by-10 study generates 100 training and 100 test sets and each of these should be passed to the different learners being studied.

Let (μ_0, δ_0) and (μ_1, δ_1) be the average accuracies and deviations of 100 trials for two different similarity measures. To test whether μ_1 is greater than μ_0, we use the hypothesis:

$$H_0 : \mu_0 = \mu_1 \quad \text{vs.} \quad H_1 : \mu_0 < \mu_1$$

Since each 10-trial 10-fold cross-validation result contains 100 trials, the difference between μ_0 and μ_1 follows the normal distribution

$$z = \frac{\mu_1 - \mu_0}{\sqrt{\frac{\delta_0^2}{100} + \frac{\delta_1^2}{100}}}$$

The significance probability for H_1 (P_{value}) is $Norm(Z < z)$ where $Norm(.)$ is the standard normal distribution.

Experimental results and discussion

Experiment results are presented in Table 2.9, including:

- Names of the data sets.
- Average accuracy μ_1 of NN with the association measure ($\phi(.,.)$).

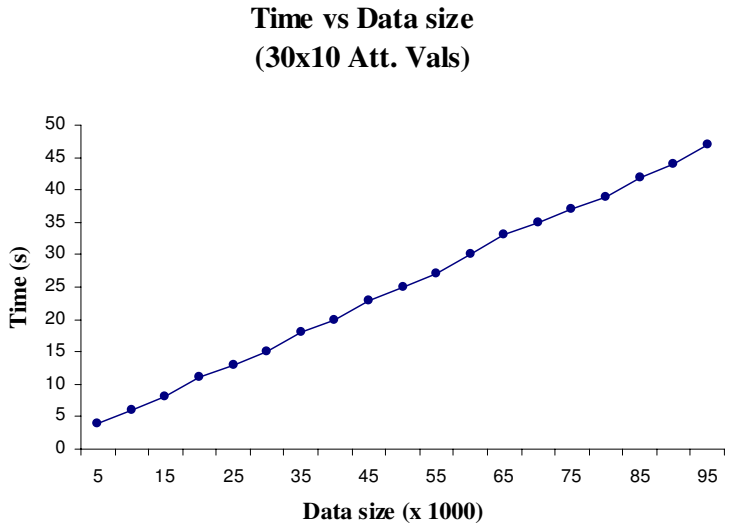


Figure 2.5: Running time versus data sizes

- Average accuracy (μ_0) of NN with any of the binary-based measures significant probability (P_{value}) that indicates the difference between the accuracy of NN with our method and the accuracy of NN with any of the binary-based measures .

- Average accuracy (μ_0) of NN with Goodall significant probability (P_{value}) that indicates the difference between the accuracy of NN with our method and the accuracy of NN with Goodall.

It can be seen from Table 2.9 that in 27 and 24 out of 30 cases, the combination of NN and the proposed method achieves a higher accuracy than the combination of NN and any of the binary-based measures as well as Goodall. In addition, NN with our method is significantly more accurate than NN with any of the binary-based measures, and Goodall in 27 and 21 out of 30 cases, respectively (P-values are greater than 95%).

Moreover, Table 2.9 shows that for data sets with high dependency between attributes (e.g. data sets *ttt*, *spice*), NN with the proposed measure are much more accurate than NN with Jaccard and Goodall (e.g. *ttt*: 97% versus 69% and 90%, splice: 85% versus 66% and 47%). However, for some data sets with low dependency factors (e.g. *monks*, *german*), the combination of NN and association-bases measure is slightly worse than the combination of NN and Jaccard (e.g. *monks*:50% versus 56% and 36% , *german*: 68% versus 69% and 70%).

Table 2.9: Experiment results

	Name	Association measures		binary-based measures			Goodall		
		Ave.	Dev.	Ave.	Dev.	P-value	Ave.	Dev.	P-value
1	allbpCleand	96	0.0095	97	0.0145	20	97	0.0118	2
2	anneal	99	0.0140	98	0.0158	100	96	0.0252	100
3	breast	96	0.0207	93	0.0249	100	96	0.0197	96
4	bridges	71	0.1437	61	0.1623	100	60	0.1395	100
5	cleve	79	0.0761	77	0.0734	99	77	0.0675	99
6	crx	83	0.0381	77	0.0462	100	78	0.0483	100
7	diabetes	67	0.0545	65	0.0548	100	66	0.0524	94
8	flare	70	0.0398	65	0.0427	100	67	0.0469	100
9	german	68	0.0442	69	0.0414	4	70	0.0488	0
10	glass	64	0.1216	60	0.1015	98	61	0.1061	96
11	heart	76	0.0720	74	0.0907	97	76	0.0719	26
12	hepatitis	85	0.0778	80	0.097	100	81	0.1001	99
13	hypo	99	0.0064	98	0.0100	100	98	0.0066	100
14	iris	92	0.0637	88	0.0754	100	91	0.0626	67
15	krvskp	88	0.0178	80	0.0226	100	87	0.0189	100
16	lymphography	85	0.0826	76	0.1125	100	76	0.1108	100
17	monks	50	0.0775	56	0.0810	0	36	0.0740	100
18	mushroom	96	0.0097	92	0.0112	100	95	0.0091	91
19	pima	74	0.0464	71	0.0492	100	73	0.0467	96
20	post-operative	61	0.1637	52	0.1508	100	55	0.1664	100
21	primary-tumor	33	0.0735	31	0.0721	98	33	0.078	50
22	promoters	82	0.1451	73	0.1378	100	54	0.1608	100
23	sick	97	0.0117	95	0.0138	100	96	0.0098	100
24	splice	85	0.0229	66	0.0273	100	47	0.0255	100
25	ttt	97	0.0257	69	0.0476	100	90	0.0380	100
26	vehicle	66	0.0471	63	0.0532	100	66	0.0447	64
27	vote	94	0.0352	90	0.0410	100	95	0.0313	1
28	waveform	77	0.0179	70	0.0205	100	65	0.0209	100
29	wine	99	0.0230	92	0.0616	100	91	0.0791	100
30	zoo	98	0.0541	88	0.1015	100	92	0.0825	100

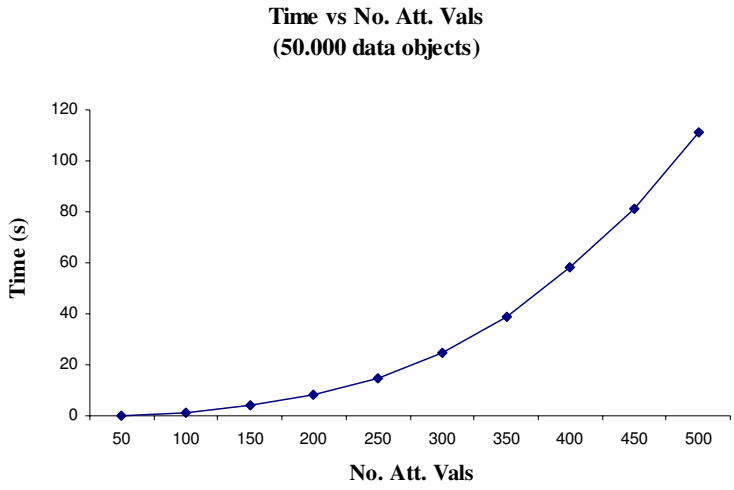


Figure 2.6: Running time versus numbers of attribute values

2.4.5 The last experiment: Analyzing time consumption

The target of the last experiment is to analyze the time consumption of the association-based measure with respect to data size and number of attributes. In the first test, the association-based measure is applied to different data sets generated from 30 attributes each contains 10 values whose sizes range from 5.000 to 100.000. The running time is reported in Figure 2.5. The second test analyzes the running time with respect to the number of attribute values. The association-based measure is applied to 50.000-object data sets whose number of attribute values range from 50 to 500. The running time is reported in Figure 2.6.

It can be seen from Figures 2.5 and 2.6 that the running time is small. It takes less than 1 minute when applying to databases of 100.000 data objects with 300 attribute values or less than 2 minutes when applying to databases of 50.000 data objects with 500 attribute values. Besides, for each database we compute the dissimilarity between attribute values one time only. After that the dissimilarity between two objects costs $O(m)$ as other measures. It means that we need a small time (1 or 2 minutes) to prepare the dissimilarity between attribute values for each database.

2.5 Conclusions

In this chapter we discussed about dissimilarity and similarity measures for categorical data. The binary-based similarity measures have advantage in simplicity and clearness. These measures are mainly based on the number of common and uncommon values. However, they are limited in variation since there are only m discriminating levels

where m is the number of attribute values. In fact, the similarity scores are m discrete numbers. The reason is that the responding binary vectors of categorical objects are constrained by the number of value 1 which is always m. Besides, since the binary-based similarity measures increase with the number of attributes in which objects have the same values, many techniques or methods such as searching, ranking and classifications using these measures produce the same results.

The association-based dissimilarity measure estimates the similarity between two values based on the relations between attributes. Since the relations between attributes vary, the dissimilarity between two objects is presented by a real number. This overcomes the variation limitation of binary-based measures. However, this similarity measure is limited to databases whose attributes are dependent.

The experiments show that the association-based dissimilarity measure improves the accuracy of NN when applying to real-life databases in comparing with binary-based measures when attributes of databases are not independent.

Chapter 3

Similarity measures for heterogeneous data

One real-life object can be reflected in many aspects that can be measured by many data types. That leads to the popularity of heterogeneous data. Measuring the similarity of heterogeneous data is a challenging task due to its heterogeneity. This chapter presents similarity measures/distances based on algebra frameworks and order-probability. Experiments with clustering for real-life data demonstrate how useful the measures.

3.1 Introduction

3.1.1 Heterogeneous data

Many real-life data sets are described by different data types such as continuous, nominal, categorical, text, or image. For example, databases of personal information may include name (text), sex (categorical: male, female), age (nominal: [0..200]), height (continuous:[0..300cm]), etc. (see Table 3.1). Databases of city information contain the name of cities (text), area (continuous), population (continuous), etc. (see Table 3.2). It is natural to map real objects by many aspects to virtual objects that are stored in databases. Obviously, those aspects are not attributed to a certain data type but to various types. For example, the name of person is text data, the sex is categorical data and the height should be continuous data.

Definition 13 *Heterogeneous data is the data where each object is described by more than one data types.*

The following data types are often used in real-life databases

Table 3.1: Personal information

Attribute	Data type	Range
Name	text	
Sex	Categorical	Male, female
Age	Nominal	[0..200]
Height: continuous	[1..300]	
Blood type	categorical	A, B, AB, O
Nation	categorical	Japanese, Vietnamese, etc.
Language	transaction	{Japanese, Vietnamese, English, etc.}
Picture	image	

Table 3.2: City information

Attribute	Data type	Range
Name	text	
Area	Continuous	
Population	Continuous	
Capital	Categorical	Yes, No
Crime level	Nominal	Low, moderate, normal, high
Map	image	

- Continuous data: data whose attribute values are continuous (i.e. the height and the blood pressure for a person).
- Discrete data: data whose attribute values are discrete values (i.e. the number of cities in a country, the number of students in a class, the number of children of a family).
- Ordinal type: data whose attribute values are ordinal values. It may be ones academic background {junior high school, high school, college or university, graduate school} or military ranks.
- Categorical type: data whose attribute values are not ordinal values (i.e. the distinction of sex $male, female$, blood types A, B, AB, O, eye color (black, blue, green, brown)). The main difference between ordinal type and categorical type is that there is an order relation between ordinal values while none for categorical values.
- Structures: data whose attribute values are structured values. For instance, graphs of city roads and 2D structures of chemical are structured data.

- Text: data whose attribute values are textual documents (i.e. articles, books and newspapers).

3.1.2 Similarity for heterogeneous data

There are many qualified similarity measures for homogenous data such as Euclidean or Minkowski distances for continuous data, Jaccard, Dice and Rao for categorical and binary data, or edit distance for sequence data. Each of these measures has its own meaning and particular properties that match the corresponding data types. For example, the dissimilarity between two continuous values is often considered as their absolute difference due to the continuous property of continuous data. Edit distance between two sequences are the minimum number of changes such that one becomes exactly to the other. Since each of the measures is based on particular properties of the corresponding data type, it is only suitable to this data type and cannot be applied to others. For instance, edit distances cannot be applied to continuous data as well as Euclidean distance cannot be applied to string data. Thus neither of them can be applied to databases that are described by both continuous and sequence data. Besides, since the similarity measures for different data types are different in meaning, it is unreasonable to integrate the similarity measures into a similarity measure for heterogeneous data. For example, it is meaningless when adding the absolute difference of continuous data and edited distance of sequence data into a similarity measure for data objects described by both continuous and sequence data.

Similarity measures for heterogenous data need to overcome differences between data types and to use up particular properties of data types. In this point of view, two main tasks of determining similarity measures for heterogeneous data are:

1. To determine the same essential (dis)similarity measures for different homogeneous data types. The (dis)similarity measures may be defined differently in each data type but should have the same meaning. Besides, the measures should be suitable for particular properties of data types.

2. To integrate properly similarity scores between attribute values into the similarity between objects. Since the measures defined in the former step may have special properties, the integration should be suitable for these properties.

Based on the framework, a few similarity measuring methods for heterogeneous data have been proposed. Generalized Minkowski metric-based methods [86, 87] consider the dissimilarity between two values of an attribute as a combination of three factors:

possition, *span* and *content*. Subsequently, the dissimilarity between two data objects is assigned by adding linearly dissimilarities of their attribute value pairs. Besides, generalized Minkowski metric approach, there is another approach which bases on two Cartesian operators *meet* ($\otimes$) and *joint* ($\oplus$) to measure the dissimilarity of one value pair [92, 89, 90]. Dissimilarities of attribute value pairs are integrated using Minkowski distance into similarities between data objects. Unlike the above algebra-based approaches, in [76] Le and Ho addressed the similarity measuring problem for heterogeneous data by a probability-based approach. They defined the similarity of one value pair as the probability of picking up randomly a value pair that is less similar than or equally similar in terms of order relations defined appropriately for data types. Similarities of attribute value pairs of two objects are then integrated using a statistical method to assign the similarity between them.

3.2 Gowda and Diday methods

Let x and y be two objects are written in the Cartesian product of m attributes $A_1, \ldots A_m$ as:

$$\mathbf{x} = x_1 \times \ldots \times x_m$$

$$\mathbf{y} = y_1 \times \ldots \times y_m$$

where x_i and y_i are values of attribute A_i.

3.2.1 Similarity measure for a single attribute

For the k^{th} attribute, the similarity between x_k and y_k, denoted $S(x_k, y_k)$, is defined using following three components:

1. $S_p(x_i, y_i)$ due to *position* p
2. $S_s(x_i, y_i)$ due to *span* s
3. $S_c(x_i, y_i)$ due to *content* c.

The similarity components due to *position* arises only when the attribute type is quantitative. It indicates the relative positions of x_i and y_i on the real axis. The similarity component due to *span* indicates the relative sizes of the attribute values without referring to common parts between them. The similarity component due to *content* is a measure of the common parts between two attribute values. These components are defined so that their values are normalized between 0 and 1. Obviously, the similarity

scores for any attribute values have the same meaning by applying the same measure scheme to all data types.

The following are the descriptions of *position*, *span*, and *content* components for different data types.

- Quantitative interval

  Let

  - a_l be the lower limit of interval x_i
  - a_u be the upper limit of interval x_i
  - b_l be the lower limit of interval y_i
  - b_u be the upper limit of interval y_i
  - $inters$ be the length of intersection between x_i and y_i
  - $l_s = |max(a_u, b_u) - min(a_l, b_l)|$ is span length of x_i and y_i,

  The three similarity components are defined as follows

  - Similarity component due to position is

    $$S_p(x_i, y_i) = 1 - \frac{a_l - b_l}{|A_i|}$$

    where $|A_i|$ denotes the length of the maximum interval of attribute A_i) (max $(a_u - a_l)$).
  - Similarity component due to span is

    $$S_s(x_i, y_i) = \frac{l_a + l_b}{2l_s}$$

    where $l_a = a_u - a_l$ and $l_b = b_u - b_l$
  - Similarity component due to content is

    $$S_c(x_i, y_i) = \frac{inters}{l_s}$$

- Quantitative ratio/Absolute type

  Quantitative ratio and absolute type of features are special cases of interval type having the following properties:

  $$a_l = a_u, b_l = b_u, l_a = l_b = inters = 0.$$

- Qualitative type

  For qualitative type of features, the similarity component due to position is absent. The two components that contribute to similarity are span and content.

  Let l_a and l_b be the lengths of x_i and y_i or number of elements in x_i and y_i. $inters$ is the number of elements common to x_i and y_i, l_s be span length of x_i and y_i combined, $l_a + l_b - inters$. The components due to span and content are defined as

  - $S_s(x_i, y_i) = \dfrac{l_a + l_b}{2l_s}$
  - $S_p(x_i, y_i) = \dfrac{inters}{2l_s}$.

3.2.2 Integration

Similarity between two objects $\mathbf{x}$ and $\mathbf{y}$ is defined as the total similarity between their attribute value pairs.

$$S(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} S(x_i, y_i) \tag{3.1}$$

$$= \sum_{i=1}^{m} S_p(x_i, y_i) + S_s(x_i, y_i) + S_p(x_i, y_i) \tag{3.2}$$

Discussion

This method uses three basic aspects to estimate the similarity between data objects: *position*, *span*, and *content*. The similarity scores that have the same meaning as the similarity between two values of any data types are estimated by the same three factors. Thus they can be easily integrated. The best advantage of this measure is its comprehension as is easy to explain the similarity score between objects.

The problem, however, lies in the determination of the three factors *possition*, *span* and *content*. It is easy to see that these three factors are not always suitable for all data types. For example, *position* cannot be applied to noncontinuous data. It means that the similarity between two values of different data types may be estimated differently. In addition, different data types require different definitions for these three factors. Thus, even using the same name with the same strategy, similarity scores between values of different data types may have different meaning. This leads to unreasonableness when integrating the similarity scores of attribute value pairs into the similarity score between objects.

3.3 Minkowski metrics

3.3.1 *Joint* and *Meet* operators

Denote $U^{(d)}$ the domain of attribute spaces

$$U^{(d)} = |A_1| \times \ldots |A_m|$$

Let $\mathbf{x} = x_1 \times \ldots \times x_m$ and $\mathbf{y} = y_1 \times \ldots \times y_m$ be two objects. The *joint* operator is defined as following:

Definition 14 (Cartesian *joint* operator) *The Cartesian joint, denoted* $\mathbf{x} \oplus \mathbf{y}$*, is defined by a cartesian product set*

$$\mathbf{x} \oplus \mathbf{y} = (x_1 \oplus y_1) \times \ldots \times (x_m \oplus y_m)$$

where $x_i \oplus y_i$ *is the Cartesian joint of* x_i *and* y_i*.*

The *joint* operator between two values x_i and y_i is defined based on the attribute type of A_i. Following are *joint* operators for common data types:

- if A_i is quantitative or ordinal qualitative, $x_i \oplus y_i$ becomes a closed interval

  $$x_i \oplus y_i = [\min(x_{il}, y_{i_l}), \max(x_{iu}, y_{iu})]$$

  where x_{il} and y_{il} are the lower bound of x_i and y_i, and x_{iu} and y_{iu} are the upper bound of x_i and y_i.

- if A_i is a nominal qualitative attribute, $x_i \oplus y_i$ becomes the union of x_i and y_i:

  $$x_i \oplus y_i = x_i \cup y_i$$

- if x_i is a tree structure, let $N(x_i)$ denote the nearest parent node common to all terminal values in x_i. Then, if $N(x_i) = N(y_i)$, let

  $$x_i \oplus y_i = x_i \cup y_i$$

  and if $N(x_i) \neq N(y_i)$, let

  $$x_i \oplus y_i = \{\text{all terminal values branched from node } N(x_i \cup y_i)\}$$

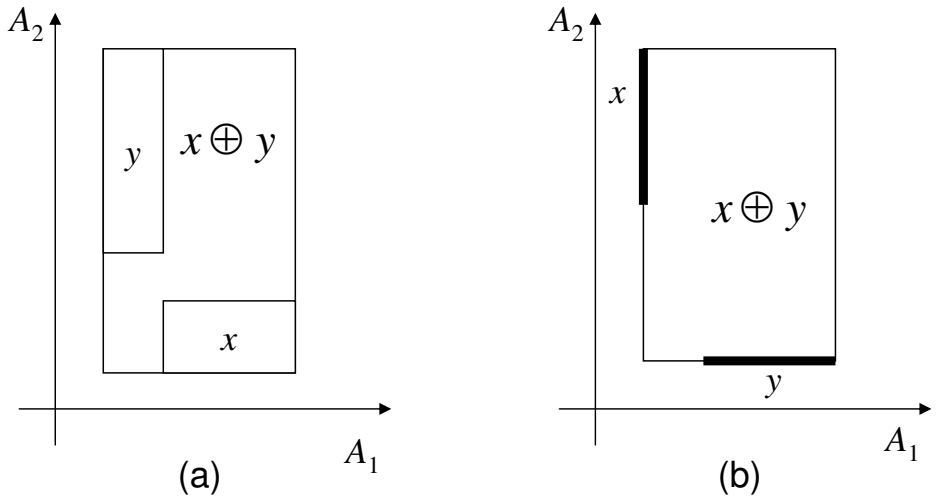


Figure 3.1: Illustration of the Cartesian *joint* in the Euclidean plane

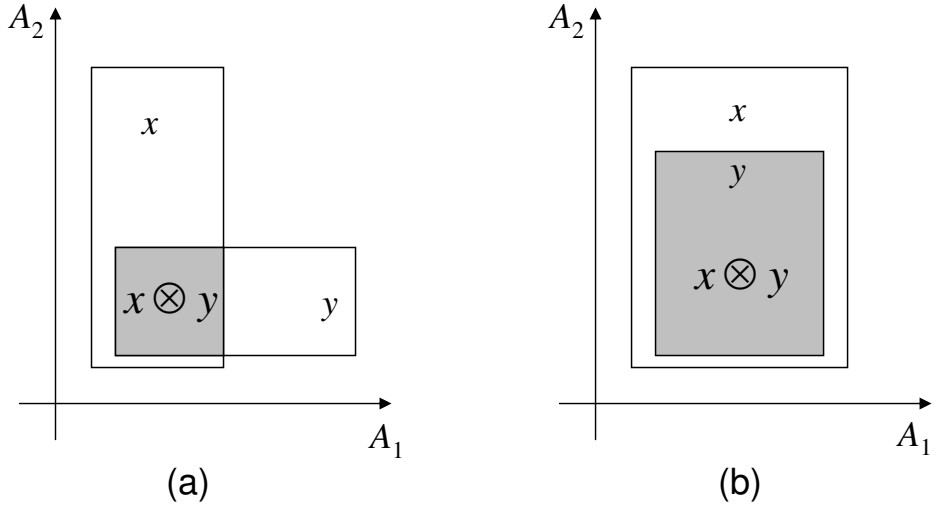


Figure 3.2: Illustration of the Cartesian *meet* in the Euclidean plane

Definition 15 (Cartesian *meet* operator) *The Cartesian meet of a pair of objects* $\mathbf{x} = x_1 \times \ldots \times x_m$ *and* $\mathbf{y} = y_1 \times \ldots \times y_m$*, denoted* $\mathbf{x} \otimes \mathbf{y}$ *is defined by a Cartesian product set:*

$$\mathbf{x} \otimes \mathbf{y} = (x_1 \otimes y_1) \times \ldots \times (x_m \otimes y_m)$$

where $x_i \otimes y_i$ *is the Cartesian meet of the* i^th *attribute and defined by the intersection of* x_i *and* y_i*:*

$$x_i \otimes y_i = x_i \cap y_i$$

Definition 16 *The distance between* x_i *and* y_i *is defined as*

$$\phi(x_i, y_i) = |x_i \oplus y_i| - |x_i \otimes y_i| + \gamma(2|x_i \oplus y_i| - |x_i| - |y_i|)$$

where $0 \leq \gamma \leq 0.5$ *and* $|x_i|$ *denotes the length of the interval* x_i *if* A_i *is continuous quantitative, and is the number of possible values included in* x_i *if* A_i *is discrete quantitative, qualitative, and structural.*

Theorem 8 *For any objects* $\mathbf{x}, \mathbf{y}$*, and* $\mathbf{z}$ *of* $U^{(d)}$*,* $\phi(.,.)$ *satisfies the following axioms for metrics:*

- $\phi(x_i, y_i) \geq 0$ *and* $\phi(x_i, y_i) = 0$ *iff* $x_i = y_i$
- $\phi(x_i, y_i) = \phi(y_i, x_i)$
- $\phi(x_i, z_i) \leq \phi(x_i, y_i) + \phi(y_i, z_i)$

The proof is given in [92].

3.3.2 Integration

There are two problems pointed out by Anderberg [101]

1. Different measure units lead to different similarity score values. For example, the value $\phi(x_i, y_i)$ is much different when attribute A_i is expressed in feet from that when A_i is expressed in inch. Likely, $\phi(x_j, y_j)$ is different when A_j is expressed in pounds and in ounces. Thus, effect of use of different units to calculate the distance between an attribute value pair must be taken into account.

2. The units for the different attribute values are combined to achieve a single measure of distance which implies a composite of the unit. Thus it is possible to interpret the sum of feet and ounces. In other words, it does not make sense to combine distance measured by different units.

To solve these problems, the attribute variables should be equalized to remove the artifact of measurement units and anchor each attribute variable to some common numerical property as follows:

$$\psi(x_i, y_i) = \frac{\phi(x_i, y_i)}{|A_i|}$$

Then for each attribute, the function $\psi(.,.)$ becomes a dimensionless quantity and

$$0 \leq \psi(x_i, y_i) \leq 1$$

Since there may be knowledge about relative importance of attributes beforehand, weight factors for attributes should be considered.

Definition 17 *The distance between two objects* $\mathbf{x}$ *and* $\mathbf{y}$ *are defined following Minkowski distance of order* $p(\geq 1)$ *as*

$$d_p(\mathbf{x}, \mathbf{y}) = \left[\sum_{i=1}^{m} c_i \psi(x_i, y_i)^p\right]^{1/p}$$

where c_i *is the weight factor of attribute* A_i *such that*

$$c_i \geq 0 \text{ and } \sum_{i=1}^{m} c_i = 1$$

By using the well-known Minkowski inequality and Theorem 8, we can prove the following proposition.

Propostion 4 *The distance between with order* $p(\geq 1)$ *satisfies all axioms for a metric.*

The proof is given in [92].

Discussion

The Minkowski metrics produce a framework for measuring the dissimilarity between heterogeneous data based on *joint* and *meet* operators. Dissimilarity measures of this frameworks satisfy axioms for metrics.

The disadvantage of this frameworks lies in determining proper *joint* and *meet* operators for different data types, the first task of building a similarity measures for heterogenous data. Determining the proper *joint* and *meet* operators for all data types is hard, somehow impossible. For example, it is very difficult to properly define *joint* and *meet* operators for graph data or sequence data. Besides, the *joint* (the intersection operator) is not always properly applicable to all data types, i.e. *joint*

operator is not suitable for continuous data. This leads to the fact that this framework is only suitable for databases containing data types that are all suitable for the *meet* and *joint* operators. That limits applications of this framework for real-life databases, especially for heterogeneous data.

3.4 Ordered probability-based similarity measure

Unlike the above algebra-based approaches, in [76] Le and Ho addressed the similarity measuring problem for heterogeneous data in probability-based approach.

For each attribute A_i, denote $\preceq_i$ an order relation on A_i^2 where $(x_i', y_i') \preceq_i (x_i, y_i)$ implies that value pair (x_i', y_i') is less similar than or as similar as value pair (x_i, y_i). For example, when A_i is continuous, $\preceq_i$ on A_i^2 can be defined as

$$(x_i, y_i) \preceq_i (x_i', y_i') \Leftrightarrow |x_i - y_i| \geq |x_i' - y_i'| \tag{3.3}$$

3.4.1 Ordered probability-based similarity measure

The first task of measuring similarity for heterogeneous data is to determine similarity measures for value pairs of each attribute. In [76], Le and Ho defined the ordered probability-based similarity for value pair (x_i, y_i) of attribute A_i as follows:

Definition 18 (Order similarity measures) *The ordered probability-based similarity between two values x_i and y_i of attribute A_i with respect to order relation $\preceq_i$, denoted by $S_{\preceq_i}(x_i, y_i)$, is the probability of picking randomly a value pair of A_i that is less similar than or as similar as (x_i, y_i)*

$$S_{\preceq_i}(x_i, y_i) = \sum_{(x_i', y_i') \preceq_i (x_i, y_i)} p(x_i', y_i')$$

where $p(x_i', y_i')$ is the probability of picking value pair (x_i', y_i') of A_i.

Definition 18 implies that the similarity of one value pair depends on both the number of value pairs that are less similar than or as similar and probabilities of picking them. Intuitively, the more number the of pairs that less similar than or as similar as one value pair, the more similarity the value pair is.

For example, consider the attribute *age* in Table 3.3 and the order relation defined in Equation 3.3. The set T of value pairs that are less similar than or as similar as (26, 55) is: $\{(23, 55), (55, 23), (25, 55), (55, 25), \ldots, (57, 26))\}$. The similarity between

26 and 55 is then estimated as:

$$\begin{aligned} S_{age}(26,55) &= p(23,55)+p(55,23)+p(25,55)+p(55,25)+\ldots+p(57,26) \\ &= \frac{1\times 1}{10^2}+\frac{1\times 1}{10^2}+\frac{1\times 1}{10^2}+\frac{1\times 1}{10^2}+\ldots+\frac{1\times 1}{10^2} \\ &= 0.18 \end{aligned}$$

As it can be induced from Definition 18, similarities of value pairs do not depend on data types. They are based only on order relations and probability distributions of value pairs. Hence, similarities of value pairs have the same meaning regardless of their data types. It means that the measure satisfies the requirement of the first task where similarity measures for different data types have the same essential meaning. Besides the measure is defined based on order relations of data types that are particularly suitable for data types. In other words, the measure is applicable to all data types and also employs particular properties of data types.

3.4.2 Order relations for real data

In the following parts, we summarize order relations of some common data types, e.g. continuous data, interval data, ordinal data, categorical data, and item set data.

It is not difficult to define order relations on A_i^2. The order relations are often induced from similarity measures on A_i^2. For example, if there is a similarity measure S_i on A_i, the order relation on A_i^2 can be induced as

$$(x_i, y_i) \preceq (x_i', y_i') \Leftrightarrow S_i(x_i, y_i) \leq S_i(x_i', y_i')$$

- **Continuous data:** A value pair is less similar or as similar as another value pair if and only if the absolute difference of the first pair is greater than or equal to that of the second pair.

$$(x', y') \preceq (x, y) \Leftrightarrow |x' - y'| \geq |x - y|$$

- **Interval data:** A value pair is less similar than or as similar as another value pair if and only if the proportion between the intersection interval and the union interval of the first pair is smaller than or equal to that of the second pair.

$$(x', y') \preceq (x, y) \Leftrightarrow \frac{|x' \cap y'|}{|x' \cup y'|} \leq \frac{|x \cap y|}{|x \cup y|}$$

- **Ordinal data:** A value pair is less similar than or as similar as to another value pair if and only if the interval between two values of the first pair contains that of the second pair:

$$(x', y') \preceq (x, y) \Leftrightarrow [x'..y'] \supseteq [x..y]$$

- **Categorical data:** A value pair is less similar than or as similar as another value pair if and only if either they are identical or values of the first pair are not identical meanwhile those of the second pair are:

$$(x', y') \preceq (x, y) \Leftrightarrow \begin{cases} x' = x, y' = y \\ x' \neq y', x = y \end{cases}$$

- **Item set data:** Following the idea of Leischner [102], the order relation for item set value pairs that come from item set M is defined as follows:

$$(X, Y), (X', Y') \in M^2 : (X', Y') \preceq (X, Y) \Leftrightarrow \begin{cases} X' \cap Y' \subseteq X \cap Y \\ \overline{X}' \cap \overline{Y}' \subseteq \overline{X} \cap \overline{Y} \\ X' \cap \overline{Y}' \supseteq X \cap \overline{Y} \\ \overline{X}' \cap Y' \supseteq \overline{X} \cap Y \end{cases}$$

Obviously, these order relations are transitive.

3.4.3 Probability approximation

Assuming that values of each attribute are independent, the probability of picking up a value pair (x_i, y_i) of A_i is approximately estimated as:

$$p(x_i, y_i) = \frac{\delta(x_i)\delta(y_i)}{n^2}$$

where $\delta(x_i)$ and $\delta(y_i)$ are the numbers of objects that have attribute value x_i, y_i respectively, and n is the number of data objects.

3.4.4 Integration methods

The similarity between two data objects consisting of m attributes is measured by a combination of m similarities of their attribute value pairs. Taking advantage of measuring similarities of attribute value pairs in terms of probability, integrating similarities of m attribute value pairs becomes the problem of integrating m probabilities.

Denote $S(\mathbf{x}, \mathbf{y}) = f(S_1, \ldots, S_m)$ the similarity between two data objects $\mathbf{x}$ and $\mathbf{y}$ where S_i is the similarity between values x_i and y_i of attribute A_i, and $f(.)$ is a function for integrating m probabilities $S_1, \ldots, S_m$.

Many measures for integrating probabilities have been proposed [103, 104, 105]. The most popular method is due to Fisher's transformation [103], which uses the test statistic

$$T_F = -2 \sum_{i=1}^{m} \ln S_i$$

and compares this to the χ^2 distribution with $2m$ degrees of freedom.

In [104], Stouffer et al. defined

$$T_s = \sum_{i=1}^{m} \frac{\Phi^{-1}(1 - S_i)}{\sqrt{m}}$$

where Φ^{-1} is the inverse normal cumulative distribution function. The value T_s is compared to the standard normal distribution.

Another P-value method was proposed by Mudholkar and Geore [105]

$$T_M = -c \sum_{i=1}^{m} \log \frac{S_i}{1 - S_i}$$

where

$$c = \sqrt{\frac{3(5m + 4)}{m\pi^2(5m + 2)}}$$

The combination value of $S_1, \ldots, S_m$ is referenced to the t distribution with $5m + 4$ degrees of freedom.

In practice, probability integrating functions are often non-decreasing functions. It implies that the greater similarity scores between attribute value pairs $S_1, \ldots, S_m$, the greater similarity between x and y, $S(\mathbf{x}, \mathbf{y})$.

Clearly these probability integrating functions are non-decreasing functions.

3.4.5 Example

To illustrate how the similarity between two data objects is measured, consider the simple data set given in Table 3.3 that was obtained from a user internet survey. This data set contains 10 data objects comprising 3 different attributes, e.g. age (continuous data), connecting speed (ordinal data), and time on internet (interval data). Consider the first data object ($\{26, 128k, [6..10]\}$ and the second one $\{55, 56k, [7..15]\}$, the similarity between them is measured as follows:

$$\begin{aligned} S_{age}(26, 55) &= p(23, 55) + p(55, 23) + p(25, 55) + p(55, 25) + \ldots + p(57, 26) \\ &= \frac{1 \times 1}{10^2} + \frac{1 \times 1}{10^2} + \frac{1 \times 1}{10^2} + \frac{1 \times 1}{10^2} + \ldots + \frac{1 \times 1}{10^2} \\ &= 0.18 \end{aligned}$$

$$
\begin{aligned}
S_{speed}(128k, 56k) &= p(14k, 128k) + p(128k, 14k) + p(28k, 128k) \\
&\quad + p(128k, 28k) + \ldots + p(56k, > 128k) + p(> 128k, 56k) \\
&= \frac{2 \times 1}{10^2} + \frac{2 \times 2}{10^2} + \frac{2 \times 1}{10^2} + \frac{2 \times 2}{10^2} + \ldots + \frac{2 \times 1}{10^2} + \frac{2 \times 2}{10^2} \\
&= 0.42
\end{aligned}
$$

$$
\begin{aligned}
S_{time}([6..10], [7..15]) &= p([5..10], [20..30]) + p([20..30], [5..10]) + p([5..10], [12..20]) \\
&\quad + p([12..20], [5..10]) + \ldots + p([3..7], [5..12]) \\
&= \frac{1 \times 1}{10^2} + \frac{1 \times 1}{10^2} + \frac{1 \times 1}{10^2} + \frac{1 \times 1}{10^2} + \ldots + \frac{1 \times 1}{10^2} \\
&= 0.76
\end{aligned}
$$

Now using Fisher's transformation test statistic [103] to integrate S_{age}, S_{speed} and S_{time} :

$$
\begin{aligned}
T_F &= -2(\ln S_{age} + \ln S_{speed} + \ln S_{time}) \\
&= -2(\ln(0.18) + \ln(0.42) + \ln(0.76)) \\
&= 5.71
\end{aligned}
$$

The value of the χ^2 distribution with 6 degrees of freedom at point 5.71 is 0.456. Thus, the similarity between the first and the second objects, $S(\{26, 128k, [6..10]\}, \{55, 56k, [7..15]\})$, is 0.456.

3.4.6 Characteristics

In this section, we investigate the characteristics and properties of the order-probability based similarity measure method. For convenience, let us recall an important required property of similarity measures that was proposed by Geist et al. [102].

Definition 19 *Similarity measure $\rho : \Gamma^2 \rightarrow R^+$ is called an order-preserving similarity measure with respect to order relation $\preceq$ if and only if it holds true for:*

$$\forall(\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}') \in \Gamma^2, (\mathbf{x}', \mathbf{y}') \preceq (\mathbf{x}, \mathbf{y}) \Rightarrow \rho(\mathbf{x}', \mathbf{y}') \leq \rho(\mathbf{x}, \mathbf{y})$$

Since order-preserving measures play important roles in practice, most common similarity measures (e.g. Euclidean, Hamming, Russel and Rao, Jaccard and Needham) possess this property with respect to reasonable order relations.

Table 3.3: A data set obtained from an user internet survey includes 10 data objects, comprising 3 different attributes e.g. age (continuous data), connecting speed (ordinal data) and time on internet (interval data)

No.	Age (year)	Connecting Speed (k)	Time on Internet (hour)
1	26	128	[6..10]
2	55	56	[7..15]
3	23	14	[5..10]
4	25	36	[20..30]
5	56	> 128	[12..20]
6	45	56	[15..18]
7	34	28	[3..4]
8	57	28	[3..7]
9	48	14	[8..12]
10	34	> 128	[5..10]

Theorem 9 *Similarity measure $S_{\preceq_i} : A_i^2 \to R^+$ is an order-preserving similarity measure with respect to order relation $\preceq_i$ if order relation $\preceq_i$ is transitive.*

Proof:

Denote $\Lambda(x_i, y_i)$ the set of pairs which are smaller than or equal to (x_i, y_i)

$$\Lambda(x_i, y_i) = \{(x_i', y_i') : (x_i', y_i') \preceq_i (x_i, y_i)\}$$

Since $\preceq_i$ is a transitive relation, for any two value pairs (x_{i_1}, y_{i_1}) and (x_{i_2}, y_{i_2}), when $(x_{i_1}, y_{i_1}) \preceq_i (x_{i_2}, y_{i_2})$ we have $\forall (x_i, y_i) \in \Lambda(x_{i_1}, y_{i_1}) : (x_i, y_i) \preceq (x_{i_1}, y_{i_1})$ implies $(x_i, y_i) \preceq_i (x_{i_2}, y_{i_2})$. This means $(x_i, y_i) \in \Lambda(x_{i_2}, y_{i_2})$, and thus

$$\Lambda(x_{i_1}, y_{i_1}) \subseteq \Lambda(x_{i_2}, y_{i_2}) \tag{3.4}$$

On other hand, we have

$$S_{\preceq_i}(x_i, y_i) \;\; = \sum_{(x_i', y_i') \preceq_i (x_i, y_i)} p(x_i', y_i') = \sum_{(x_i', y_i') \in \Lambda(x_i, y_i)} p(x_i', y_i') \tag{3.5}$$

From (3.4) and (3.5),

$$S_{\preceq_i}(x_{i_1}, y_{i_1}) = \sum_{(x_i, y_i) \in \Lambda(x_{i_1}, y_{i_1})} p(x_i, y_i) \leq \sum_{(x_i, y_i) \in \Lambda(x_{i_2}, y_{i_2})} p(x_i, y_i) = S_{\preceq_i}(x_{i_1}, y_{i_1})$$

Thus, $S_{\preceq_i}(.,.)$ is an order-preserving measure. ■

In practice, order relation $\preceq_i$ are often transitive. Thus, the ordered probability-based similarity measures for attributes are also order-preserving similarity measures.

Denote $\mathbb{A} = A_1 \times \ldots \times A_m$ the product space of m attributes $A_1, \ldots, A_m$. The product of order relation $\preceq_1, \ldots, \preceq_m$ is defined as follows:

Definition 20 *The product of order relations $\preceq_1, \ldots, \preceq_m$, denoted by $\prod_{i=1}^m \preceq_i$, is an order relation $\preceq$ on $\mathbb{A}^2$, for which one data object pair is said to be less similar than or as similar as another data object pair with respect to $\prod_{i=1}^m \preceq_i$ if and only if attribute value pairs of the first data object pair are less similar than or as similar as those of the second data object pair*

$$\forall(\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}') \in \mathbb{A}^2 : (\mathbf{x}', \mathbf{y}') \preceq (\mathbf{x}, \mathbf{y}) \Leftrightarrow (x_i', y_i') \preceq_i (x_i, y_i), i = 1, \ldots, m$$

Propostion 5 *The product of order relations $\preceq_1, \ldots, \preceq_m$ is transitive when order relations $\preceq_1, \ldots, \preceq_m$ are transitive.*

Proof:
Denote $\preceq = \prod_i^m \preceq_i$. For any triple data object pairs $(\mathbf{x_1}, \mathbf{y_1}), (\mathbf{x_2}, \mathbf{y_2})$, and $(\mathbf{x_3}, \mathbf{y_3})$. if $(\mathbf{x_1}, \mathbf{y_1}) \preceq (\mathbf{x_2}, \mathbf{y_2})$, and $(\mathbf{x_2}, \mathbf{y_2}) \preceq (\mathbf{x_3}, \mathbf{y_3})$, we have

$$(\mathbf{x_1}, \mathbf{y_1}) \preceq (\mathbf{x_2}, \mathbf{y_2}) \Leftrightarrow (x_{i_1}, y_{i_1}) \preceq_i (x_{i_2}, y_{i_2}) \ \ \forall i = 1 \ldots m$$
$$(\mathbf{x_2}, \mathbf{y_2}) \preceq (\mathbf{x_3}, \mathbf{y_3}) \Leftrightarrow (x_{i_2}, y_{i_2}) \preceq_i (x_{i_3}, y_{i_3}) \ \ \forall i = 1 \ldots m$$

Since $\preceq_i$ is transitive for $i = 1 \ldots m$, $(x_{i_1}, y_{i_1}) \preceq_i (x_{i_2}, y_{i_2})$ and $(x_{i_2}, y_{i_2}) \preceq_i (x_{i_3}, y_{i_3})$ implies $(x_{i_1}, y_{i_1}) \preceq_i (x_{i_3}, y_{i_3})$. Hence $(\mathbf{x_1}, \mathbf{y_1}) \preceq (\mathbf{x_3}, \mathbf{y_3})$.

Thus, $\prod_i^m \preceq_i$ is transitive. ■

Theorem 10 *Similarity measure $S : \mathbb{A}^2 \to R^+$ is an order-preserving similarity measure with respect to $\prod_{i=1}^m \preceq_i$ when order relations $\preceq_1, \ldots, \preceq_m$ are transitive and probability integrating function f is non-decreasing.*

Proof:
Denote $\preceq = \prod_i^m \preceq_i$ and let $(\mathbf{x}', \mathbf{y}')$ and $(\mathbf{x_2}, \mathbf{y_2})$ be two data object pairs. We have

$$(\mathbf{x}', \mathbf{y}') \preceq (\mathbf{x}, \mathbf{y}) \Leftrightarrow (x_i', y_i') \preceq (x_i, y_i) \ \forall i = 1, \ldots, m;$$

Since $\preceq_i$ is transitive for $i = 1, \ldots, m$, following Theorem 9,

$$S_i' = S_{\preceq_i}(x_i', y_i') \leq S_{\preceq_i}(x_i, y_i) = S_i \ \forall i = 1, \ldots, m$$

Since f is a non-decreasing function,

$$S(\mathbf{x}', \mathbf{y}') = f(S'_1, \ldots, S'_m) \leq f(S_1, \ldots, S_m) = S(\mathbf{x}, \mathbf{y})$$

Since $(\mathbf{x}', \mathbf{y}') \preceq (\mathbf{x}, \mathbf{y}) \Rightarrow S(\mathbf{x}', \mathbf{y}') \leq S(\mathbf{x}, \mathbf{y})$, $S(.,.)$ is an order-preserving similarity measure with respect to $\prod_i^m \preceq_i$. ■

Theorem 10 says that if attribute value pairs of an object pair are less similar than or equal to those of another object pair, the similarity of the first object pair is smaller than or equal to the similarity of the second object pair under conditions that order-relations $\preceq_1, \ldots, \preceq_m$ are transitive and probability integrating function f is non-decreasing.

Discussion

The main advantage of this framework is the ability to avoid using common operators for all data types that algebra-based approaches use. Using order relations overcomes natural differences between data types and helps to treat homogeneously different data types. Moreover, it allows to take particular properties of data types to form the order relations, or in other words, to indirectly estimate the similarity between objects. Besides, a similarity measure of this framework contains the most important properties of similarity measures.

However, the disadvantage of the framework is running time. Since to estimate the similarity between two values may cost $O(n^2)$, it limits applications of this framework to large databases. However, for each data type there may be an algorithm for estimating the similarity between two values with reasonable times. For example, we can estimate the similarity between two values of continuous attribute in $O(n)$. Reducing running time to estimate the similarity between values is an opening problem and should be paid much attention.

3.5 Applications to real data

In the following parts, we analyze real data sets using ordered probability-based similarity measure in conjunction with clustering methods.

3.5.1 Data set

The Cultural Issues in Web Design data set was obtained from the GVU's 8th WWW User Survey (http://www.cc.gatech.edu/gvu/user_surveys/survey-1997-10/). The data

set is a collection of users's opinions on influences of languages, colors, culture, etc. on web designs. The data set includes 1097 respondents, which are described by 3 item set attributes, 10 categorical attributes, and 41 ordinal attributes.

3.5.2 Methodology

Similarity measure method

We apply the ordered probability-based similarity measure to measure similarities between respondents of the Cultural Issues in Web Design data set. The Fisher's transformation is chosen to integrate similarities of attribute value pairs.

Clustering methods

Most of clustering methods belong to either the partitioning approach or hierarchical approach. Most of partitioning methods such as K-means [34] and Kmedoid [35] requires three conditions:

1. continuation of data for convergence problems.
2. A method for detecting representatives for clusters.
3. The triangle inequality property to guarantee qualities of clustering results.

Since these conditions are rarely satisfied when applying for heterogenous data, partitioning methods are not suitable in this case.

Hierarchical clustering methods are summarized in tree diagrams. Initially, there are N clusters, each contains a single object. The number of clusters reduces by one at each step of algorithm, by amalgamating two most similar pair of existing clusters into a new one. The different ways of defining the dissimilarity between two clusters of objects lead to different clustering strategies. If two clusters C_i and C_j are amalgamated, a general relation for evaluating the dissimilarity between $C_i \cup C_j$ with some other cluster C_k, is

$$\begin{aligned} d(C_i \cup C_j, C_k) &= \alpha_i d(C_i, C_k) + \alpha_j d(C_j, C_k) + \beta d(C_i, C_j) + \\ &\quad \gamma |d(C_i, C_k) - d(C_j, C_k)| + \delta_i h(C_i) + \delta_j h(C_j) + \epsilon h(C_k) \end{aligned}$$

where $h(C_i)$ denotes the height in the dendrogram of C_i. The formula without terms δ_i, δ_j, and ϵ was proposed by Lance and Williams [106, 107] and the complete formula was proposed by Jambu [36]. The recursion is started by defining $d(C_i, C_j) = d(o_i, o_j)$ for singleton cluster $C_i = \{o_i\}$ and $C_j = \{o_j\}$. When clusters C_i and C_j are chosen for

amalgamation, the height in the dendrogram of their union $C_i \cup C_j$, $h(C_i \cup C_j)$, is given by $d(C_i, C_j)$. Different choices of the parameters $\{\alpha, \beta, \gamma, \delta\}$ define different clustering strategies. The most common ones are summarized in Table 3.4.

In the experiment, the hierarchical clustering with the mean similarity criteria was chosen. Each step, two clusters with the maximum average similarity between data objects are chosen to be merged. The loop continues until the required number of clusters is reached.

3.5.3 Clustering results

The Cultural Issues in Web Design data set was clustered into 10 clusters. However, characteristics of only three clusters were presented due to space limitation (see Table 3.5). A characteristic of a cluster is presented as an attribute value that majority of respondents of the cluster answered. For example, value *can't write* of attribute *Unfamiliar site* is considered as a characteristic of the first cluster because 92% respondents of this cluster answered the value.

3.5.4 Remarks from experiment results

As it can be seen from Table 3.5, the clusters have many characteristics, e.g. the first and second clusters have 13 characteristics. Moreover, the clusters' characteristics differ from cluster to cluster. In particular, when visiting an *unfamiliar site*, the problem of 92% respondents of the first cluster is *cannot write*, while 81% respondents of the second cluster is *cannot translate*, and 84% respondents of the third cluster is *cannot read*. Moreover, answers of respondents in the same clusters are somehow similar. For example, all respondents of the first cluster can neither read *Rabic* and *Hebrew* nor speak *Bengari* and *Hebrew*. In short, almost all respondents in the same cluster have the same answers but they are different from answers of respondents form different clusters. The analysis of characteristics from these clusters shows that our similarity measuring method in combination with the agglomeration hierarchical average linkage clustering method discovers valuable clusters of real data sets.

3.6 Conclusions

In this chapter, we discussed similarity measures and dissimilarity measures for heterogeneous data. The introduced measures follow the same frameworks including: determining the same essential similarity measures for different data types and inte-

Table 3.4: Clustering strategies obtainable from the general recurrence relation of Jambu (1978)

	Name (reference)	δ_i	β	γ	δ	ϵ
1	Single link [37]	$\frac{1}{2}$	0	$\frac{-1}{2}$	0	0
2	Complete link [40]	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
3	Group average link [38, 39]	$\frac{n_i}{n_i+n_j}$	0	0		
4	Weight average link [108, 39]	$\frac{1}{2}$	0	0	0	0
5	Mean dissimilarity [109]	$\frac{\mathbf{C}_2^{n_i+n_k}}{\mathbf{C}_2^{n_+}}$	$\frac{\mathbf{C}_2^{n_i+n_j}}{\mathbf{C}_2^{n_+}}$	0	$\frac{-\mathbf{C}_2^{n_i}}{\mathbf{C}_2^{n_+}}$	$\frac{-\mathbf{C}_2^{n_k}}{\mathbf{C}_2^{n_+}}$
6	Sum of squares [109]	$\frac{n_i+n_k}{n_+}$	$\frac{n_i+n_j}{n_++}$	0	$\frac{-n_i}{n_+}$	$\frac{-n_k}{n_+}$
7	Incremental sum square [110, 111]	$\frac{n_i+n_k}{n_+}$	$\frac{-n_k}{n_+}$	0	0	0
8	Centroid [38, 112]	$\frac{n_i}{n_i+n_j}$	$\frac{-n_in_j}{(n_i+n_j)^2}$	0	0	0
9	Median [106, 112]	$\frac{1}{2}$	$-\frac{1}{4}$	0	0	0
10	Flexible [106]	$\frac{1}{2}(1-\beta)$	β	0	0	0

n_i denotes the number of objects in the cluster C_i; $n_+ \equiv n_i + n_j + n_k$

Table 3.5: Characteristics of three discovered clusters

Cluster 1

No.	Att. Names	Value	P_a	Value	P_a	Value	P_a	Value	P_a
1	Unfamiliar sites	Can't write	92					Other	8
2	Read Arabic	None	100						
3	Read Hebrew	None	100						
4	Speak Bengali	None	100						
5	Speak Hebrew	None	100						
6	Primary same as Native	Yes	98	No	2				
7	Important problem	Can't write	92	None	2			Other	6
8	American images	None	79					Other	21
9	Native Language	English	79	Chinese	4	German	4	Other	12
10	Read German	None	71	Basic phrases	19	Native	8	Other	2
11	Software	Yes both	73	Yes get	25	No	2		
12	Speak English	Native	69	Conver.	17	None	14		
13	Provide native sites	Agree strongly	69	Agree somewhat	23	Disag. somewhat	4	Other	4

Cluster 2

No.	Att. Names	Value	P_a	Value	P_a	Value	P_a	Value	P_a
1	Unfamiliar sites	Can't translate	81					Other	19
2	Read Chinese	None	100						
3	Read Hindi	None	100						
4	Read Japanese	None	100						
5	Speak Hindi	None	100						
6	Due to culture	No	93	Yes-both	8				
7	Sites in non-fluent	Few	89	None	9	Most	2		
8	Non-English sites	Few	89	None	8	Half	4		
9	Translations	Yes-useful	87					Other	13
10	Read German	None	83	Basic phrases	9	Literate	8		
11	Native Language	English	81	Spanish	8	Arabic	2	Other	9
12	Speak German	None	81	Basic phrases	11	Conver.	8		
13	Designed culture	Yes	70	No	28	Don't know	2		

Cluster 3

No.	Att. Names	Value	P_a	Value	P_a	Value	P_a	Value	P_a
1	Unfamiliar sites	Can't read	84					Other	16
2	Read Arabic	None	100						
3	Read Chinese	None	100						
4	Read Hindi	None	100						
5	Speak Arabic	None	100						
6	Speak Bengali	None	100						
7	Speak Hindi	None	100						
8	Read Italian	None	93	Basic phrases	4	Native	2	Other	2
9	Speak Italian	None	93	Basic phrases	7				
10	Speak Spanish	None	84	Basic phrases	14	Conver.	2		
11	Read Spanish	None	82	Basic phrases	18				
12	Sites designed for culture	Yes	68	No	29	Dontknow	4		
13	Sites in non-fluent	Few	77	All	11	None	7	Other	5
14	Software	Yes get	77	Yesboth	18	No	5		
15	Non-English sites	Few	68	None	21	Half	9	Other	2

grating properly similarity scores of attribute value pairs into similarity scores between objects.

The algebra-based approach uses the same common factors such as position of values or common parts to estimate the similarity between values. However, this ideas faces with problems of determining factors/operators that are suitable for all data types.

The framework to estimate the similarity between two values by ordered probability overcomes the problem algebra-based methods face since it bases on order relations that are particularly built for data types. However, time consumption is the main drawback of this framework. Since the algebra methods require $O(m)$ to estimate the similarity between two objects, this framework requires at least $O(nm)$. This limits applicability

of this framework to large databases.

All of the measures mentioned in this chapter are designed for data sets whose data objects have the same number of attribute values. However, in real databases, the number of attributes of data objects may be different from objects to objects. Thus, there is a need of an investigation on how to adapt these measures to databases whose objects may have different numbers of attributes.

Chapter 4

Similarity measures for graph data

Recently, there has been an increased interest in graph data such as intrusion semantic web, behavioral modeling, or image processing. Due to its complex structures, to measure similarity for graph data is a challenging task. This chapter shortly summarizes popular similarity measures including the ϕ distance similarity measure, the measure of papadopoulos and manolopoulos, similarity based on the maximal common subgraph, the edit distance for graphs, and introduces a nonoverlap connected subgraph-based measure.

4.1 Introduction

In recent years, there has been an increased interest in developing data mining algorithms that operate on graphs. Such graphs arise naturally in a number of different application domains including network intrusion semantic web [113], behavioral modeling [114, 115] VLSI reverse engineering [116], link analysis [117, 118, 119] and chemical compound classification [120, 121, 122, 123, 124, 125], image processing [23, 10, 24]. Moreover, they can be used to effectively model the structural and relational characteristics of a variety of datasets arising in other areas such as physical sciences (e.g., chemistry, fluid dynamics, astronomy, structural mechanics, and ecosystem modeling), life sciences (e.g. genomics, proteomics, pharmacogenomics, and health informatics), and home-land defense (e.g. information assurance, network intrusion, infrastructure protection, and terrorist-threat prediction/identification). Examples of such data objects are molecules, images or audio data. Those data objects have a complex internal structure, e.g. atoms in a molecule (see Figure 4.1), protein structures (see Figure 4.2) or objects in an image (see Figure 4.3).

A graph object is defined by a labelled graph as

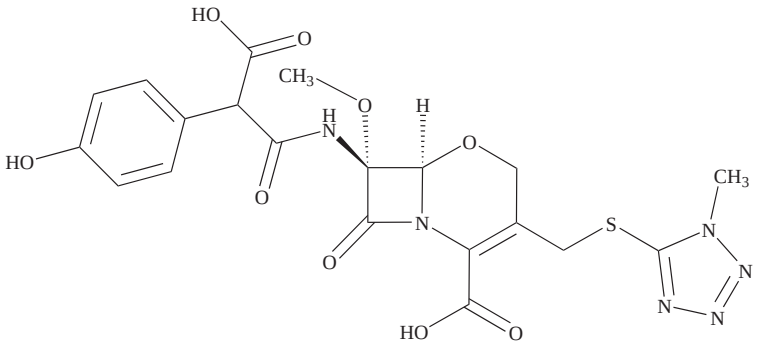


Figure 4.1: Molecular structure: Moxalactam Latamoxef

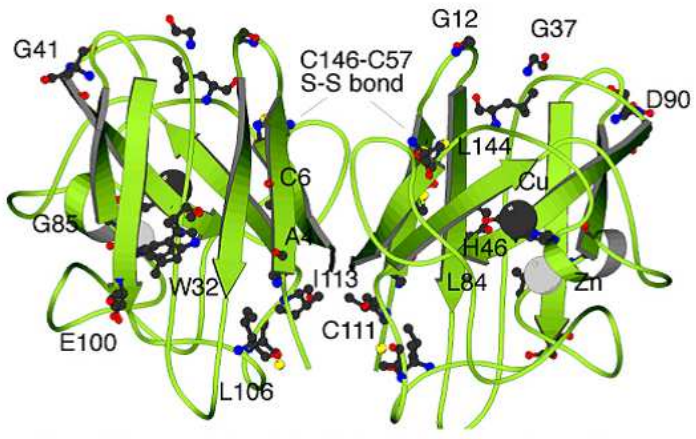


Figure 4.2: Protein Structure

Definition 21 (Labelled graph) *A labelled graph is a 4-tuple* $G =< V, E, \alpha, \beta >$ *where*

- V *is the the finite sets of vertices.*
- $E \subseteq V \times V$ *is the set of edges.*
- $\alpha : V \rightarrow L_V$ *is a function assigning labels to the vertices.*
- $\beta : E \rightarrow L_E$ *is a function assigning labels to the edges.*

4.1.1 Similarity measures for graph data

There exist several similarity measures for graphs. They differ in the types of graphs for which they are defined and whether they take attribute information into account or not. But most of the measures have one thing in common, which is that they are based on some sort of edit operations. The basic idea of all those measures is to define

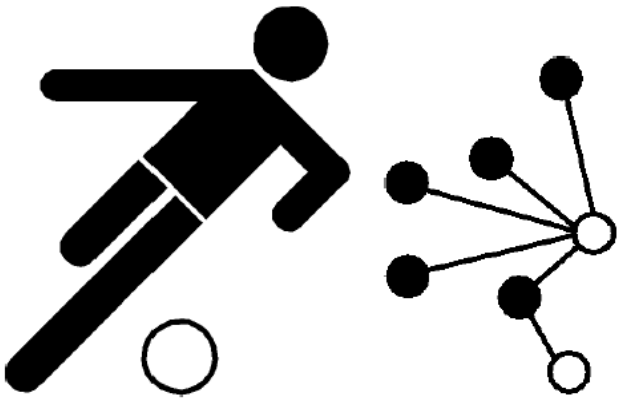


Figure 4.3: An image and the extracted graph.

the similarity of graphs based on the effort needed to make the graphs identical. This effort is measured in number of primitive operations which are needed to make the graphs identical. The following sections presents the similarity measures for graphs from the literature and discuss, how different approaches define the identity of graphs and effort to achieve it.

4.1.2 Basic notation

Definition 22 (order and degree) *Given a graph* $G =< V, E, \alpha, \beta >$.

- *The number of vertices of* G*, denoted as* $|V|$*, is called the order (size) of* G.
- *The number of edges incident to a vertex* v *is called the degree of* v*, denoted by* $degree(v)$.
- *An edge* $e = (u, v)$ *is called incident to the vertices* u *and* v*. Two vertices are said to be adjacent if there exists an edge that is incident to both of them.*

Definition 23 *A path from vertex* u *to vertex* v *in a graph is an alternating sequence* $< u = u_0, u_1, \ldots, u_k = v >$ *where* u_i *and* u_{i+1} *are adjacent. A graph* G *is said to be connected, if* G *contains a path between each pair of vertices* u *and* v.

Definition 24 *Let* $G =< V, E, \alpha, \beta >$ *and* $G' =< V', E', \alpha', \beta' >$ *be graphs.* G' *is a subgraph of* G *if and only if*

- $V' \subseteq V$
- $\alpha'(v) = \alpha(v)$ *for all* $v \in V'$

- $E' = E \cap (V' \times V')$
- $\beta'(e) = \beta(e)$ *for all* $e \in E'$

From Definition 24 it follows that given a graph $G =< V, E, \alpha, \beta >$, any subset $V' \subseteq V$ of its vertices uniquely defines a subgraph. This subgraph is called the subgraph *induced* by V'.

Definition 25 *Let* $G =< V, E, \alpha, \beta >$ *and* $G' =< V', E', \alpha', \beta' >$ *be graphs. A graph isomorphism between* G *and* G' *is a bijective map* $f : V \mapsto V'$ *such that*

- $\alpha(v) = \alpha'(f(v))$ *for all* $v \in V$
- *For any edge* $e = (u, v) \in E$*, there exists an edge* $e' = (f(u), f(v)) \in E'$ *such that* $\beta(e) = \beta(e')$*, and for any edge* $e' = (u', v') \in E'$*, there exists an edge* $e = (f^{-1}(u), f^{-1}(v)) \in E$ *such that* $\beta(e') = \beta(e)$.

G *is called a isomorphism graph of* G'*, denoted* $G' \cong G$.

If $f : V \mapsto V'$ is a graph isomorphism between G and G', and G' is a subgraph of another graph G'', $G' \subseteq G''$, then f is called a subgraph isomorphism from G to G''. G' is called the responding subgraph of G in G''.

Definition 26 $\bar{G} =< \bar{V}, \bar{E} >$ *is called a subgraph of* $G =< V, E >$ *if and only if* $\bar{V} \subseteq V$ *and* $\bar{E} = \bar{V} \times \bar{V} \cap E$

Definition 27 *Let* $G_1 =< V_1, E_1, \alpha_1, \beta_1 >$ *and* $G_2 =< V_2, E_2, \alpha_2, \beta_2 >$ *be graphs. A common subgraph of* G_1 *and* G_2*, denoted* $cs(G_1, G_2)$*, is a graph* $G = (V, E)$ *such that there exist subgraph isomorphisms from* G *to* G_1 *and from* G *to* G_2.

There are several figures used to describe graphs, e.g. chromatic number or girth. The most important for the discussions in the following chapters are the order and size of a graph and the degree of a vertex.

4.2 The ϕ distance similarity measure

In [126] Chartrand et al. proposed a similarity measure for graphs. This measure is based on mappings between the vertex sets of the graphs, which are compared, and is defined for connected graphs of the same order. Before defining this similarity measure, the ϕ-distance is introduced.

Definition 28 (ϕ-distance) *Given two connected graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ of the same order n and a one-to-one mapping $\phi : V_1 \mapsto V_2$. The ϕ distance between G_1 and G_2 is defined as*

$$dist_\phi(G_1, G_2) = \sum |l_p(u, v) - l_p(\phi(u), \phi(v))|$$

where $l_p(u, v)$ is the length of the shortest path between u and v in G_1 and G_2.

The ϕ-distance similarity measure is defined as follows:

Definition 29 (ϕ-distance similarity measure) *The ϕ-distance similarity measure between two connected graphs G_1 and G_2 of the same order is defined as:*

$$d_\phi(G_1, G_2) = \min\{dist_\phi(G_1, G_2) : \phi : V_1 \mapsto V_2 \text{ is a one to one mapping}\}$$

In [126], Chartrand et al. proved that the ϕ-distance similarity measure is a metric. Besides, for any two connected graphs G_1 and G_2 of the same order, the following holds:

$$|td(G_1) - td(G_2)| \leq d_\phi(G_1, G_2)$$

where

$$td(G) = \sum l_p(u, v)$$

If G_1 is a connected subgraph with $n - 1$ edges (spanning tree) of G_2, it can even be shown that

$$d_\phi(G_1, G_2) = td(G_1) - td(G_2)$$

Theorem 11 *When G_1 and G_2 are two graphs with the same order, to determine $d_\phi(G_1, G_2)$ is NP-hard.*

Proof
We prove that $d_\phi(G_1, G_2) = 0$ when and only when G_1 and G_2 are isomorphic.

- $d_\phi(G_1, G_2) = 0 \rightarrow G_1, G_2$ are isomorphic.

  Since $d_\phi(G_1, G_2) = 0$, $l_p(u, v) = l_p(\phi(u), \phi(v)) \forall u, v$. Thus, there is an edge between u and v if and only if there is an edge between $\phi(u)$ and $\phi(v)$. That means G_1 and G_2 are isomorphic.

- If G_1 and G_2 are isomorphic, then $d_\phi(G_1, G_2) = 0$

  Let $\phi : V_1 \mapsto V_2$ be the isomorphic map. It is obvious that if $l_p(u, v) = 1$ then $l_p(\phi(u), \phi(v)) = 1$. Assuming $l_p(u, v) = l_p(\phi(u), \phi(v))$ when $l_p(u, v) \leq k \ \ \forall u, v$. We prove that $l_p(u, v) = l_p(\phi(u), \phi(v))$ when $l_p(u, v) = k + 1$.

Since $l_p(u,v) = k+1$, there exists u' such that $l_p(u,u') = 1$ and $l_p(u',v) = k$. It is clear that $l_p(\phi(u),\phi(u')) = 1$ and $l_p(\phi(u'),\phi(v)) = k$. Thus, $l_p(\phi(u),\phi(v)) = k+1$. Due to $l_p(u,v) = l_p(\phi(u),\phi(v))$, $d_\phi(G_1,G_2) = 0$.

Since the problem of determining whether G_1 and G_2 are isomorphic is an NP-hard, to determine $d_\phi(G_1,G_2)$ is an NP-hard. ■

Discussion

The ϕ-distance similarity measure is defined only for connected graphs of the same order and does not take attribute information into account. The integration of attribute information would be possible by using a distance function which takes attribute information into account instead of the lengths of paths between the pairs of vertices. The choice of this distance function would have to be done carefully in order to preserve the metric property of the ϕ-distance similarity measure. Nevertheless, the limitation to connected graphs of the same order remains, which limits the applicability of the ϕ-distance similarity measure to special cases where the requirements are fulfilled.

Additionally, the measure has no parameter and, therefore, is not adaptive to application requirements and user needs. Just like integrating attribute information, adaptability could be achieved by introducing another distance function for vertex pairs. Again, the choice of this function would have to be done with special care to preserve the metric properties.

Finally, the time complexity of the measure remains an open issue. However, since there is no algorithm with polynomial time complexity known, which calculates the ϕ-distance similarity measure, a moderate time complexity of this measure cannot be approved.

4.3 Similarity Based on the Maximal Common Subgraph

Based on the maximum common subgraphs, many similarity measures have been proposed [127, 128] (see Table 4.1). These measures consider the similarity between two graphs with the intuition that the larger their common parts in comparing with the size of two graphs, the more similar these two graphs. In fact, these measures are defined after coefficients similarity measure for binary vectors introduced in Chapter 2.

In [130], Bunke and Shearer introduced a distances for graph structures named maximal common subgraph similarity distance.

Table 4.1: Cost-based similarity coefficients

Reference	$d(G_1, G_2)$	Range
Wallis et al.[127]	$\frac{\lvert G_{12}\rvert}{\lvert G_1\rvert + \lvert G_2\rvert}$	0 to 1
Asymmetric [128]	$\frac{\lvert G_{12}\rvert}{\min(\lvert G_1\rvert + \lvert G_2\rvert)}$	0 to 1
Sokal and Sneath [128]	$\frac{\lvert G_{12}\rvert}{2\lvert G_1\rvert + 2\lvert G_2\rvert - 3\lvert G_{12}\rvert}$	0 to 1
Kulczyski [128]	$\frac{\lvert G_{12}\rvert(\lvert G_1\rvert + \lvert G_2\rvert)}{2\lvert G_1\rvert\lvert G_2\rvert}$	0 to 1
McCpmmaughey [128]	$\frac{\lvert G_{12}\rvert(\lvert G_1\rvert + \lvert G_2\rvert)}{\lvert G_1\rvert\lvert G_2\rvert} - 1$	-1 to 1
Tanimoto [129]	$\frac{\lvert G_{12}\rvert}{\lvert G_1\rvert + \lvert G_2\rvert - \lvert G_{12}\rvert}$	0 to 1

$|G_{12}|$ is the size of the MCS or MCES between two graphs of sizes of $|G_1|$ and of $|G_2|$.

Definition 30 (maximal common subgraph similarity distance) *The maximal common subgraph distance between two non-empty graphs G_1 and G_2 is defined as*

$$d_{mcs}(G_1, G_2) = \frac{1 - |mcs(G_1, G_2)|}{max\{|G_1|, |G_2|\}}$$

They proved that this distance is metric.

Optimal algorithms to find a maximum common subgraph of two graphs often based on max clique detection [131] or backtracking [132] (see [133] for more detail). Since these algorithms require exhausted search, it is unsuitable to apply to real-life data. Thus, approximated algorithms for finding clique are often used [134, 135, 136]. Figure 6.1 presents k-opt algorithms introduced by Kengo et al. [135].

Discussion

Different from the similarity measure of Papadopoulos and Manolopoulos and the ϕ-distance similarity measure, the maximal common subgraph similarity distance is defined for graph data and is not restricted to certain graph types.

In [130], it is stated that a design goal for the development of the measure is to avoid the need for a cost function within the similarity measure. As a reason for this, the complexity of choosing the best cost function for edit distance based similarity measure is mentioned. But because of the lack of a cost function, the maximal common subgraph similarity distance is not adaptive to specific applications and user needs. This fact greatly limits the usability of the measure for many applications.

An explanation for the similarity distance could be provided by presenting the maximal common subgraph determined during the calculation. But obviously, this is no longer an important requirement, since the measure has no parameters which are adaptive.

While the measure fulfills the metric properties, it can only be calculated with exponential time complexity. Therefore, it does not meet the requirement of moderate computational complexity.

4.4 The Edit Distance for Graphs

The edit distance for graphs is an extension of the well known edit distance for strings [137, 138] to graphs. Sanfeliu and Fu first introduced the edit distance for attributed graphs in [139]. The edit distance between two graphs is the minimum number of edit operations which are necessary to transform the graphs into each other. Edit operations may be the deletion or insertion of vertices or edges or the change of

vertex or edge attributes. There exist many variants of the edit distance for graphs which differ in the edit operations that are allowed or whether attributes are considered or not.

The edit distance for graph is the most common similarity measure for graphs for several reasons. First, it is a very intuitive measure, with which users can easily understand how the distance between two objects comes about. As a consequence, the users can set parameters systematically if the results of a similarity search are satisfying. This allows us to apply the edit distance in broad range applications and strengthens the trust of users in the results. Furthermore, the calculation of the edit distance also produces a mapping between the vertices of the two compared graphs, which can be visualized for users. This supports users in the often explorative similarity search process and again, in adapting necessary parameters. Another property of the edit distance which also increases its adaptability for different applications and users is the fact that variants of the edit distance are available. Those variants are on different weights for the edit operations, a restriction of the allowed edit operations, or on a combination of those two techniques.

Edit distance has been used successfully in many application domains such as face recognition [140] or object recognition [141].

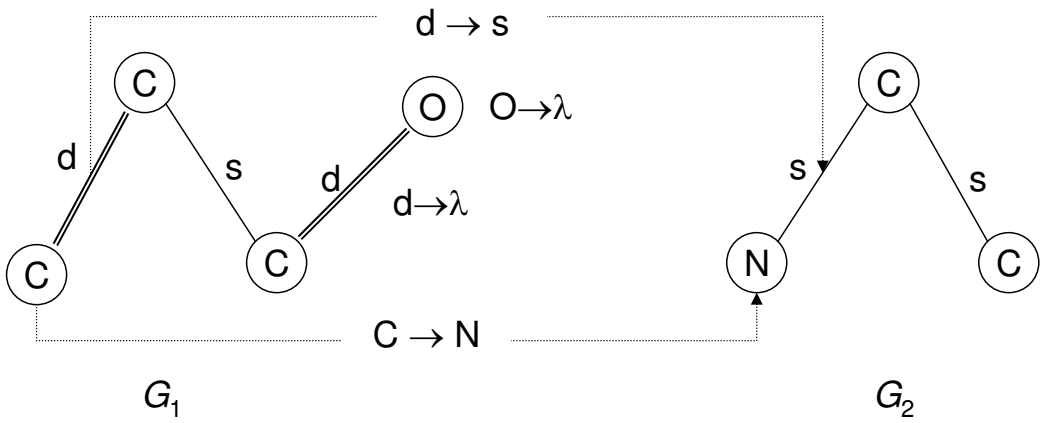
Definition 31 (Edit operation, edit sequence) *Let $G = (V, E, \alpha, \beta)$ be an attributed graph. An edit operation is the insertion, the deletion or the change of a label (relabelling) of a vertex or edge in G. The insertion of a vertex or edge x is denoted by $(\lambda \rightarrow x)$, the deletion of x is denoted by $(x \rightarrow \lambda)$ and the relabelling of x to y is denoted by $(x \rightarrow y)$. An edit sequence S is a sequence of edit operations, $S =< e_1, \ldots, e_m >$ which can be applied to G. The result of the application of an edit sequence S to a graph G, $S(G)$, is an edited graph G'.*

Definition 32 (Edit cost function) *Each edit operation e is assigned a non-negative cost $c(e)$. The cost of a sequence of edit operations $S =< e_1, \ldots, e_m >$ is defined as the sum of the cost of each edit operation in S.*

$$c(S) = \sum_{i=1}^{m} c(e_i)$$

Definition 33 (Edit distance) *The edit distance between two attributed graphs G_1 and G_2, $d_{edit}(G_1, G_2)$, is the minimum cost of all edit sequences S that make G_1 and G_2 isomorphic:*

$$d_{edit}(G_1, G_2) = \min\{c(S) | S(G_1) \cong G_2\}$$



$$d_{edit}(G_1, G_2) = 4$$

Figure 4.4: Simple edit distance between two graphs. The distance is calculated with unit cost for all edit operations.

The simplest and most common variant of the edit distance is the weighted edit distance. It differs from the simple edit distance in the cost function for edit operations. While for the simple edit distance, each edit operation is assigned the same cost, in the weighted case, the cost for insertion, deletion and relabeling operations can differ. It is even possible that the cost for an edit operation depends on the individual objects involved in the edit operation. The cost for a relabeling, for example, may be proportional to how much the values of the labels are changed.

The similarity measure for graphs from Papadopoulos and Manolopoulos [142], as already described in Section **??**, also represents a special form of edit distance. But in contrast to the measures presented in the previous sections, they do not define an insert or delete operation for edges, but introduce a vertex update operation. Consequently, the deletion of a single edge takes two edit operations which are the update operations for the two incident vertices. Consequently, graphs with different size are considered less similar with Papadopoulos and Manolopoulos measure than with the normal edit distance. Additionally, the measure is only defined for non-attributed graphs. While this problem could be solved by introducing an appropriate relabeling operation, the resulting measure would be incompatible with the efficient search methods presented in [142].

Time Complexity of the Edit Distance

In [143], Zhang et al. showed that computing the edit distance between unordered trees is an NP-hard which means that it has no polynomial approximation scheme unless P = NP. Consequently, techniques that reduce the complexity of the query processing become indispensable when using the edit distance.

Discussion

The edit distance for attributed graphs meets almost all of the requirements of a similarity measure. With the suitable edit cost, we can have a metric distance. Since these measures are comprehensible for human, they are widely used, especially in case there is a need of explanations about the similarity scores between graphs. Besides, users can interfere into similarity measures by using weighed edit operators.

The first weak point of the edit distance is that since this distance needs predefined weights of edit operators, it requires users to have advance knowledge of data. It is difficult for users, especially when facing with large or unknown databases. The second one is the complexity. As other similarity measures do, the time complexity of the edit distance and its variants is extremely high. This limits applicability of these measures to large databases.

4.5 Nonoverlap connected subgraph-based measure

In [77], Le et al. presented an idea of using connectivity and closeness of common subgraphs to estimate the similarity between two graphs.

4.5.1 Similarity measure

Definition 34 *Two subgraphs G' and G'' of G are called nonoverlaped if their induced vertex sets are nonoverlapped.*

Denote $\mathbf{\Gamma} = \{G_i : i = 1..m\}$ a set of common nonoverlap subgraphs of G and G'. Given subgraph G_i, its closeness with respect to G is estimated as a combination of the common edges and vertices in both G_i and G. To this end, each vertex $v \in V_i$ is weighted as the ratio between the number of edges starting from v in E_i and that E, $\tau(v, G_i, G)$:

$$\tau(v, G_i, G) = \frac{|\{(v, v') \in E_i\}|}{|\{(v, v') \in E\}|} \tag{4.1}$$

It is easy to see the more common edges between G_i and G with respect to the vertex v, the greater the weight $\tau(v, G_i, G)$. In fact, since the number of edges from v in G_i is always less than or equal to that in G, the greater number of edges from v in G_i means the closer structure at node v in G_i with respect to G. Thus, $\tau(v, G_i, G)$ presents how similar the structure at node v in G_i in comparing with that of the responding node in G. The maximum value of $\tau(.)$ reaches when these structures are identical.

Then, the closeness $\rho(G_i, G)$ of G_i in G is estimated as all weights of vertices $v \in V_i$

$$\rho(G_i, G) = \sum_{v \in V_i} \tau(v, G_i, G) \tag{4.2}$$

Following is an important theorem that ensures the less separated subgraph means the higher closeness.

Theorem 12 *For any set of nonoverlap subgraphs of a subgraph G', $\mathbf{\Gamma} = \{G_i =< V_i, E_i, \alpha_i, \beta_i >: i = 1.\ldots.\}$, it holds true that*

$$\sum_{G_i} \rho(G_i, G) \leq \rho(G', G)$$

The equality occurs when and only when $\mathbf{\Gamma} = \{G'\}$

Proof:

- $\sum_{G_i} \rho(G_i, G) \leq \rho(G_i, G)$ Since G_is are subgraphs of G', $\tau(v, G_i, G) \leq \tau(v, G', G)$ when $v \in V_j$.

  Since V_is are not overlapped,

  $$\sum_{G_i} \sum_{v \in V_i} \tau(v, G_i, G)) \leq \sum_{G_i} \sum_{v \in V_i} \tau(v, G', G)) \leq \sum_{v \in V'} \tau(v, G', G)$$
  $$\Rightarrow \sum_{G_i} \rho(G_i, G) \leq \rho(G', G).$$

- $\sum_{G_i} \rho(G_i, G) = \rho(G_i, G) \Leftrightarrow \mathbf{\Gamma} = \{G'\}$

  – " $\Rightarrow$ " Since

  $$\sum_{G_i} \sum_{v_i \in V_i} \tau(v_i, G_i, G) \leq \sum_{G_i} \sum_{v_i \in V_i} \tau(v_i, G', G) \leq \sum_{v' \in V'} \tau(v', G', G),$$

  $$\bigcup_{V_j} = V_i, \text{ and } \tau(v_j, G_j, G) = \tau(v_j, G_i, G) \ \ \forall v_j \in V_i.$$

  Thus $G_j \equiv G_i$ or $\mathbf{\Gamma} = \{G_i\}$

– " $\Leftarrow$ " is obvious. ■

Corollary 1 *For any subgraphs G_i and G_j. If G_j is a subgraph of G_i, then $\rho(G_j, G) \leq \rho(G_i, G)$*

Equation 4.1 and Corollary 1 say the larger the common parts and more similar to the compound structures, the higher similarity score.

Further, the closeness $\delta(G_i, G, G')$ of the common subgraph G_i with respect to two graphs G and G' is estimated as follows

$$\delta(G_i, G, G') = \rho(G_i, G) \times \rho(G_i, G') \tag{4.3}$$

Here, the closeness $\delta(G_i, G, G')$ is considered as an estimator of the similarity between two graphs G and G' when they share common subgraph G_i.

Finally, the similarity between two graphs G and G' is defined as the closeness of all $G_i \in \mathbf{\Gamma}$. However, since the size of graph are different, it is normalized this sum by the size of G and G'

$$\delta(\mathbf{\Gamma}, G, G') = \frac{\sum_{G_i \in \mathbf{\Gamma}} (\rho(G_i, G) \times \rho(G_i, G'))}{|V| \times |V'|} \tag{4.4}$$

Theorem 13 *Let $\mathbf{\Gamma} = \{G_1, \ldots, G_k\}$ and $\mathbf{\Gamma}' = \{G'_1, \ldots, G'_{k'}\}$ be two sets of nonoverlapped connected common subgraphs of G and G'. If $G_i \in \mathbf{\Gamma}$ for $i = 1 \ldots k$ is a subgraph of $G'_j \in \mathbf{\Gamma}'$, then $\delta(\mathbf{\Gamma}) \leq \delta(\mathbf{\Gamma}')$.*

Proof:
According to 12, it is clear that for G_{i_k}(k 1..) in $\mathbf{\Gamma}$ being subgraphs of G'_j in $\mathbf{\Gamma}'$,

$$\sum_k \rho(G_{i_k}, G) \leq \rho(G'_j, G), \quad \sum_k \rho(G_{i_k}, G') \leq \rho(G'_j, G')$$

On the other hand, we have

$$\sum_k \rho(G_{i_k}, G)\rho(G_{i_k}, G') \leq \sum_k \rho(G_{i_k}, G) \sum_k \rho(G_{i_k}, G') \leq \rho(G'_j, G)\rho(G'_j, G')$$

Consequently,

$$\delta(\mathbf{\Gamma}) = \frac{\sum_{G_i} (\rho(G_i, G)\ \rho(G_i, G'))}{\rho(G, G)\ \rho(G', G')} \leq \frac{\sum_{G'_j} (\rho(G'_j, G)\ \rho(G'_j, G'))}{\rho(G, G)\ \rho(G', G')} = \delta(\mathbf{\Gamma}').$$ ■

It can be induced from Theorem 13 that the larger subgraphs G_is are, the greater is the weight of $\mathbf{\Gamma}$.

Definition 35 (Maximum set of nonoverlap common connected subgraphs) *The maximum nonoverlap common connected subgraphs set between two graphs G and G', stand for $\mathbf{\Gamma}^*$, is defined as*

$$\mathbf{\Gamma}^* = \arg\max_{\mathbf{\Gamma}} \{\delta(\mathbf{\Gamma}, G, G')\}$$

Finally, the similarity between two graph G and G' is defined as:

Definition 36 *The similarity between two graphs G and G' is defined as the weight of the maximum nonoverlap common connected subgraphs:*

$$\psi(G, G') = \delta(\mathbf{\Gamma}^*, G, G') \tag{4.5}$$

4.5.2 Properties

Propostion 6 *For any pair of graphs (G, G'), the following properties hold true:*

1. $0 \leq \psi(G, G') \leq 1$
2. $\psi(G, G') = \psi(G', G)$
3. $\psi(G, G') = 1$ *if and only if G and G' are isomorphic graphs.*
4. $\psi(G, G') = 0$ *if and only if G and G' have no common connected subgraphs of the size larger than 1.*

Proof:

1. From the definition in Equation 4.1, $\rho(G_i, G) \leq |V_i|$.

   Thus, for any common subgraph set $\mathbf{\Gamma}$ of (G, G'),

   $$\delta(\mathbf{\Gamma}) \leq \frac{\sum_i (|V_i|)^2}{|V|\,|V'|}$$

   Meanwhile, since G_is are disjoint common subgraphs of (G, G'),

   $$\sum_i |V_i| \leq \min(|V|, |V'|).$$

   Hence,

   $$\sum_i |V_i|^2 \leq \left(\sum_i |V_i|\right)^2 \leq \min(|V|, |V'|)^2 \leq |V||V'| \tag{4.6}$$

   This leads to $\psi(G, G') \leq 1$.

   The left part of Property 1 can be obviously seen.

2. It is apparent that $\delta(\mathbf{\Gamma})$ is the same no matter (G, G') or (G', G). Thus, Property 2 is true.

3. $\psi(G, G') = 1 \Leftrightarrow$ the equality in inequality (4.6) happens. This is equivalent to $|V| = |V'|, |\mathbf{\Gamma}| = 1$, and $\rho(G_1, G) = \rho(G_1, G') = |V_1||V|$, which means G and G' are isomorphic.

4. $\psi(G, G') = 0$ is equivalent to $\rho(G_i, G) = 0$ and $\rho(G_i, G') = 0 \quad \forall G_i \in \mathbf{\Gamma}$. That means $|V_i| = 1$ for all G_i, or G and G' have no connected common subgraphs of the size larger than 1. ■

Theorem 14 *Finding $\psi(G, G')$ is an NP-hard problem.*

Proof:

Due to Proposition 2 $\psi(G, G') = 1$ if and only if G and G' is isomorphic. So detecting $\psi(G, G')$ implies detecting G and G' are isomorphic or not. Since the isomorphic problem is an NP-hard problem, detecting $\psi(G, G')$ is an NP-hard problem. ■

We can learn from Theorem 14 that determining exact similarity between two graphs requires a huge amount of time. It is impossible when applying to large databases or databases of large graphs. Thus, there is a need of heuristic algorithms to approximate the similarity between graphs in reasonable running time.

4.5.3 Approximation algorithm

Proposition 1, Theorem 12 and 13 show the larger connected common subgraph, the greater its weight is. This leads to a naturally approximated method which sequentially finds out the largest commonly connected subgraph (see Algorithm 2). Although finding out the largest commonly connected subgraph is also an NP-hard problem, it can be solved efficiently by applying the backtracking algorithm.

Discussion

This similarity measure has no parameter and, therefore, does not need any priory knowledge of users. However, users can interfere in weighing the closeness scores to make the measure suitable for their purposes. Similar to similarity measures based on the maximal common subgraph, this measure is not restricted to certain graph types.

An explanation for the similarity dissimilarity could be provided by presenting nonoverlap common connected subgraphs determined during calculating. In addition,

Algorithm 2 Algorithm for determining the similarity between two graphs

1: $\mathbf{\Gamma} = \emptyset$
2: **repeat**
3: cG = the largest connected common of (G, G')
4: $\mathbf{\Gamma} = \mathbf{\Gamma} + cG$
5: $G = G - cG, G' = G' - cG$
6: **until** $cG = \emptyset$
7: return $\delta(\mathbf{\Gamma})$

the calculation is intuitive and comprehensive to human. Moreover, it is no longer an important requirement as the measure has no parameters which need to be adapted.

The time complexity of this measure remains an open issue. Since there is no algorithm with polynomial time complexity known, which calculates the measure, a moderate time complexity of this measure cannot be approved. However, the proposed approximation algorithm can help to estimate reasonably this similarity measure.

4.6 Conclusion

This chapter mentioned similarity measures for graph data. The structure complexity leads to difficulty in estimating the similarity for graph data. Strategies of similarity measures for standard data cannot be applied to this kind of data.

The main common points of similarity measures for graph data are that they based on common parts (corresponding parts) to estimate their similarity measures. Thus, the first problem of each measure is to determining their common parts. The $\phi()$ distance and the measure of Papadopoulos and Manolopoulos are to find a map between two graphs and determine the similarity score based on this map. Similarity measure based on the maximum common subgraph, edit distance, and the nonoverlap connected subgraph-based measure search the particular common parts between two graphs to form their similarity score.

The main limitations of the similarity measures are their complexity. The problem of detecting common parts (corresponding parts) of the similarity measures are often an NP-Hard. This leads to huge and unrealistic time consumption estimating exactly the similarity between two graph objects when applying to large graph objects or large databases. However, each of the measure has heuristic algorithms for approximating the similarity in reasonable times.

Chapter 5

Applications of graph similarity measures for 2D chemical structures

This chapter reports experiments when applying similarity measures for graph data to 2D chemical structure data. The first experiments are to compare similarity measures based on the maximal common subgraph and the nonoverlap connected subgraph-based measure in classification nearest neighbor. The other experiments are to apply clustering with the nonoverlap connected subgraph-based measure to real-life databases. These experiments help to discover interesting relations between structures and chemical properties and that between clusters of structures and clusters of enzymes in pathways.

5.1 Introduction

Measuring the similarity between chemical compounds (molecules) is one of the primary tasks in chemistry and biology. Many applications from different areas such as classification, clustering, database searching, protein-ligand docking, reaction site modeling, and biological prediction are mainly based on similarity measures to product results. For the purpose of drug discovery, structures are considered similar if they carry similar biological activity. Thus, the structure representation has to consider those properties of a chemical structure that are deemed to be responsible for the biological activity under investigation. The similar property principle states that structurally-similar molecules exhibit similar properties. Searching is also an area where similarity measure play an important roles. For example, finding similar or relevant compounds to a given

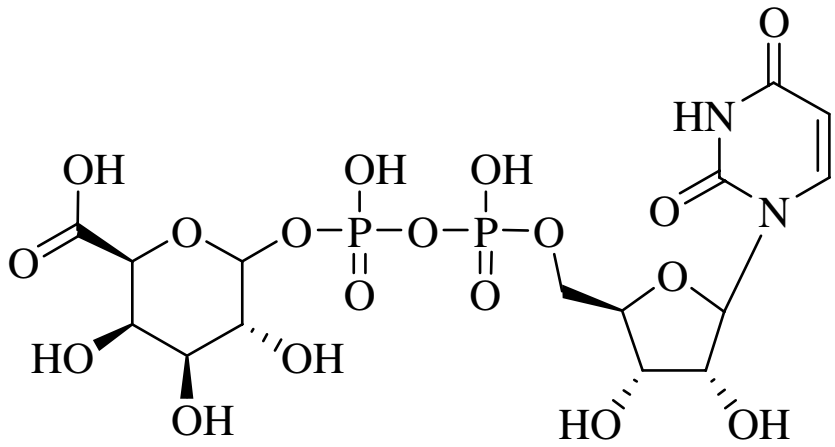


Figure 5.1: (R)-AMAA, (R)-2-Amino-2-(3-hydroxy-5-methyl-4-isoxazolyl)acetic acid

compounds needs similarity measures to determine how similar between compounds in databases and given compounds. The more proper the similarity measure, the more relevant output we obtain. The use of similarity searching in chemical databases is given in [144].

The similarity between compounds is often measured by comparing their 2D structures where a compound is presented by a graph in which vertices and edges present compound's atoms and bonds. The main reasons for using 2D structures are its adequacy for most real-life purposes [122, 123] and easy detection. Besides, 2D visualization is comprehensive for human.

5.2 2D Chemical structure

The 2D presentation of a chemical compound is a graph $G =< V, E, \alpha, \beta >$ where

- The node set V is the set of atoms
- The edge set E is the set of links
- Map α maps atom names to node labels
- Map β maps link types to edge labels.

For example, Figure 5.1 and 5.2 show compound "(R)-AMAA (R)-2-Amino-2- (3-hydroxy-5-methyl-4-isoxazolyl) acetic acid" and its graph structure.

Two popular formats to present chemical structures are SMILES and CTfile formats.

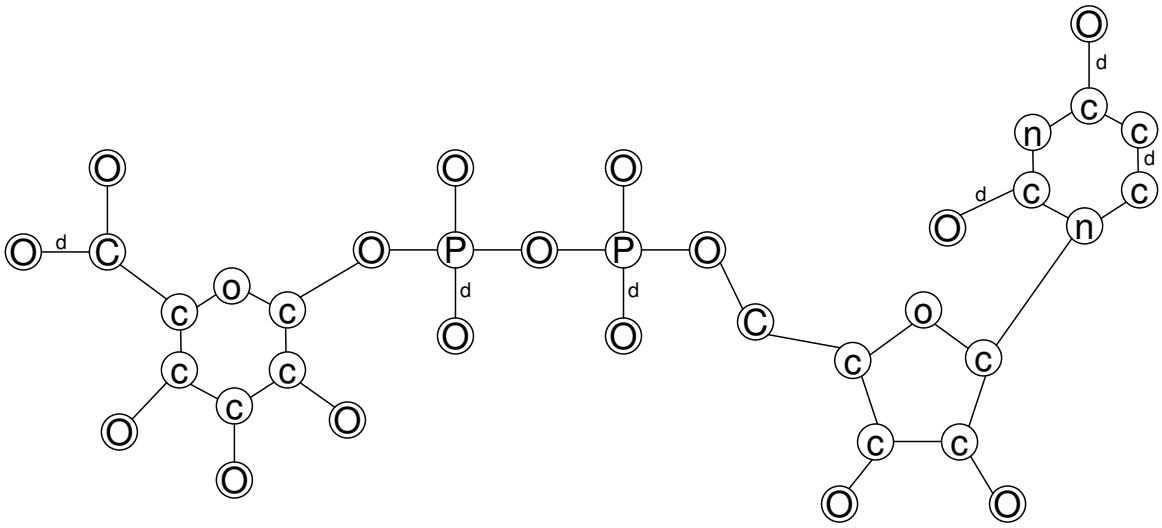


Figure 5.2: the graph representation of (R)-AMAA, (R)-2-Amino-2-(3-hydroxy-5-methyl-4-isoxazolyl)acetic acid

- SMILES [145], (Simplified Molecular Input Line Entry System) is a linear string representation language for chemical molecules. The SMILES language is commonly used in computational chemistry and is supported by most major software tools in the field, like the commercial Daylight toolkit or the Open-Source Open-Babel library. The SMILES notations of chemical compounds are comprised of atoms, bonds, parenthesis, and numbers:

  - **Atoms**: Atoms are represented using their atomic symbols, i.e. C for carbon, N for nitrogen, or S for sulfur. For aromatic atoms, lower case letters are used, and upper case letters otherwise. Atoms with two letter symbols, like chlorine (Cl) or bromine (Br), are always written with the first letter in upper case and the second letter can be written either with upper or lower case. With a rare few exceptions, hydrogen atoms are not included in the string representation of a molecule.
  - **Bonds**: Four basic bond types are used in the SMILES language: single, double, triple, and aromatic bonds, represented by the symbols: "-", "=", "#", and ":" respectively. Single and aromatic bonds are usually omitted from SMILES strings. Not belonging to the four basic bonds are ionic bonds, or disconnections, represented by a '.'.
  - **Branches**: Branches are specified by enclosing brackets, "(" and ")", and indicate side-structures. A branch can, and often does, contain other branches.

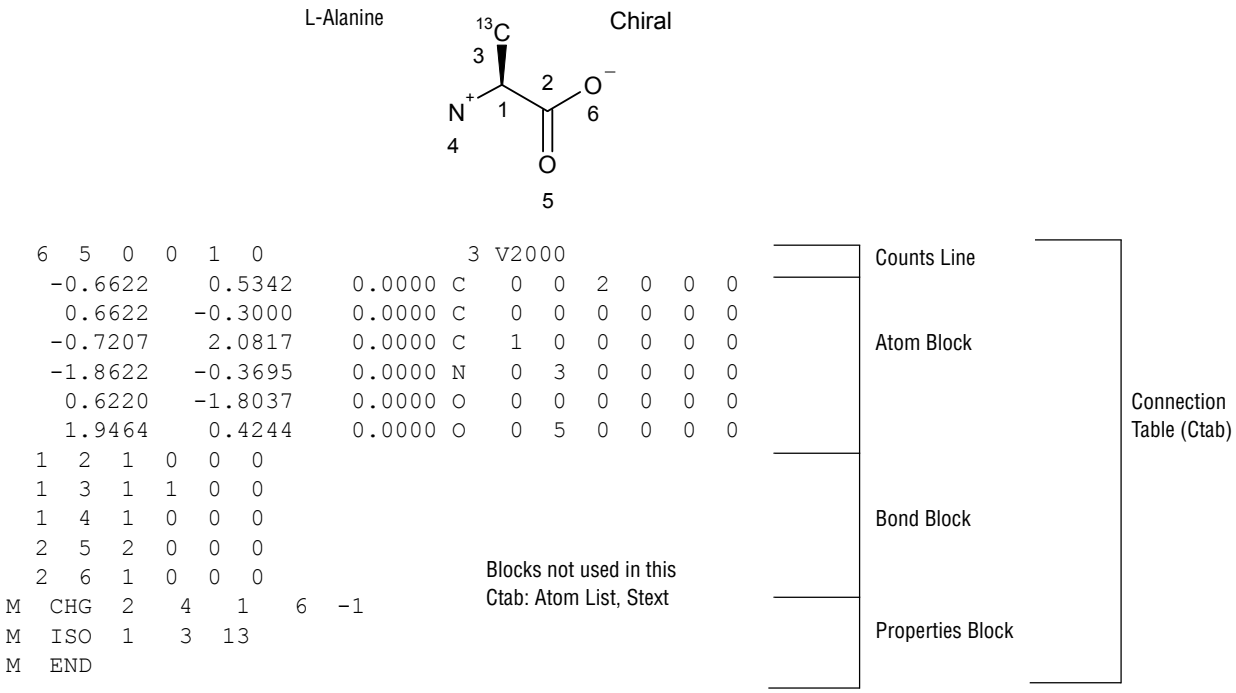


Figure 5.3: Mol format

- **Cycles**: Cyclic structures are represented by breaking one bond in each ring. The atoms adjacent to the bond obtain the same number. For example, "cccccc" denotes a (linear) sequence of six aromatic carbons and "c1ccccc1" denotes a ring of six carbons. Here, we refer to the numbers indicating the cyclic structures of a compound as cyclic link numbers. These cyclic link numbers are not unique within the a SMILES representation of a molecule. For instance, "c1ccccc1Oc2ccccc2" or "c1ccccc1Oc1ccccc1" are valid notations for the same molecule.

For example, the smile presentation of compound (R)-AMAA, (R)-2-Amino-2-(3-hydroxy-5-methyl-4-isoxazolyl) acetic acid (Figure 5.1):

O[CH]([CH]1O)C(O[CH]1COP(OP(OC([CH](O)[CH]2O)O

[CH]([C](O)=O)[CH]2O)(O)=O)(O)=O)N(C=CC3=O)C(N3)=O

- CTfile [146] presents directly a chemical structure by fixed atoms in space and list connections between items. Different from SMILES format, CTfile stores a compound by a list of coordinates of atoms and connection between atoms. The key piece describing a chemical structure is the connection table (Ctab) block indicated all information of the structure. Figure 5.3 shows an example of Mol structure.

5.3 Similarity measures for 2D chemical structures

Methods for measuring the similarity between chemical compounds based on 2D structures have been proposed [147, 122, 123, 148, 149, 150, 125, 151, 152, 153]. The main approaches of the measures are the fingerprint-based comparison and the graph-based comparison. The first approach considers a molecule as a bit-string where each bit shows the presence or absence of either an atom or an important predefined molecular substructure (called the key descriptor or finger) [147]. The similarity between two molecules is then determined by comparing their corresponding bit-strings [122, 123]. In addition, the combination of numerical vector methods and fingerprint-based methods has been used [78, 80, 81, 154]. The main complication of the approach is difficulty in deciding the key descriptors [148, 149].

The graph-based approach measures the similarity between two compounds by comparing their 2D structures. Basically, similarity measures for graphs can be applied to chemical structures. However, similarity measures in use are often estimated by either the maximum common subgraphs (MCS) [150, 125, 151] or the maximum common edge subgraphs (MCES) [152, 153]. Estimating the similarity between two graphs by the number of vertices (or edges) in the MCS (or MCES) shows two problems. Naturally, atoms of a chemical compound are connected directly by a bond or indirectly by a chemical path, in which any two adjacent atoms on the path are directly connected. In other words, the structure graph of a chemical compound is often connected. An interpretation of this nature is that a smaller set of connected atoms with their chemical bonds is usually much more meaningful than a lager set of separated atoms. Thus the size (number of nodes or edges) of MCS or MCES is not a good representative for the similarity between two chemical compounds. In addition, both atoms and chemical bonds play equally importance roles in chemical compounds so to exclude one could not be good solutions.

Some modification of using MCS have been proposed [125, 151]. The main idea is that the same atom species in chemical compounds must be discriminated by different labels because they show different physicochemical properties according to their spatial and chemical circumstances. In fact, 68 atom types were predefined. Then the matched score between two atoms was defined based on the match between two structures related to two atoms.

Since the graph-based similarity measures are designed for generate graphs, there is a gap between the measures and nature sense of chemical compounds. To fill the gap, Le et al. proposed a new view for the similarity between chemical compounds based on their common parts [77]. The idea is to measure the similarity between two

compounds based on the closeness of the common part's structures in comparing with compound structures under conditions that these common parts must be connected and nonoverlapped. These conditions guarantee chemical nature senses of the common parts. In fact, the closeness factor is estimated so that the larger the common parts and the more similar they are to the compound structures, the higher closeness score.

5.4 Experiments with classification

5.4.1 Methodology

To show the merit of similarity measures when applying to classification methods, the nearest neighbor method was chosen as its accuracy strongly relies on similarity measures and presents how good the measures are. The tests were carried out to compare the nonoverlap connected subgraph-based measure and the measures based on the maximum common subgraphs given in Table 4.1. It is easy to see that NN produces the same accuracy results when using any of the measures given in Table 4.1 [79].

Since finding the maximum common subgraph of two graphs is an NP-Hard problem, the approximated algorithm proposed by Kengo et al. in [135] was chosen (See Figure 6.1).

5.4.2 Databases

Eight databases are selected from three sources to guarantee the diversity of data. That helps to avoid bias in selecting databases.

- The first two data sets F188 and F42 are the mutagenicity databases [155, 156]. The databases contain structures information as well as chemical properties of compounds.
- 5 data sets from U.S Environmental Protection Agency [157] including:
  - DBPCAN: Water Disinfection By-Products Database with Carcinogenicity Estimates Carcinogenicity estimates (high, moderate, low concern) by EPA.
  - EPAFHM: EPA Fathead Minnow Aquatic Toxicity Database
  - NCTRER: FDA's National Center for Toxicological Research - Estrogen Receptor Binding Database
  - CPDBRM: Carcinogenic Potency Database Rat and Mouse

Table 5.1: The accuracy of NN with our similarity measure and with Tanimoto coefficient measure

			Accuracy (%)		Running time (s)	
No	Name	Size	Nonover. conn. subgraph-based	Max. Com. Subgraph	Alg. 2	K-opt
1	F42	42	88.10	80.95	108	93
2	F188	188	84.04	77.13	2909	3486
3	CPDBRM	1189	52.73	52.65	1624	56668
4	CPDBHA	79	60.76	60.76	1	83
5	DBPCAN	209	62.20	56.94	3	180
6	EPAFHM	614	57.33	51.63	61	3852
7	NCTRER	230	85.22	70.43	313	4999
8	Ligand	377	65.78	61.42	2914	23295

  - CPDBHA: Carcinogenic Potency Database Hamster

- The last data, Ligand, is obtained from the KEGG database [158].

5.4.3 Results and discussion

Table 5.1 shows the results of NN with the nonoverlap connected subgraph-based measure and with the measures based on the maximum common subgraphs. It can be seen that NN with the nonoverlap connected subgraph-based measure is clearly more accurate than NN with the measures that are based on the maximum common subgraph (i.e., NCTRER: 85.22% in compare with 70.43%) for 5 data sets. For two data sets CPDBRM and CPDBHA, all of these measures produce the same accuracy. The experiment says that the nonoverlap connected subgraph-based measure can boost clearly the accuracy of NN in comparison with similarity measure based on the maximum common subgraph for some real-file data.

The running time for estimating the similarity between compounds of the nonoverlap connected subgraph-based measure is much less than that of K-opt for MCS (Table 5.1). For instance, the nonoverlap connected subgraph-based measure needs 1624 seconds for database CPDBRM meanwhile K-opt does 56668 seconds. That is because the number of connected combinations is much smaller than the number of arbitrary combinations.

5.5 Experiments with clustering

This section presents results when applying a clustering method with the nonoverlap connected subgraph-based measure to more than eleven thousands compounds obtained from the KEGG database [158]. The first experiments are to analyze relations between clusters of compounds of the whole database with other chemical information such as pathways, enzymes, etc. The second experiments are to analyze relations between pathway modules identified by clusters of similar structure compounds and that identified by genomic contexts, namely, operon structures of enzyme genes.

5.5.1 Clustering methods and Database

Clustering methods can be divided into two main approaches: partitioning and hierarchical. Since partitioning methods [34, 35] are not suitable for noncontinuous data, a hierarchical-based clustering method is chosen to cluster compounds. Among hierarchical-based clustering methods, the method with the average complete linkage condition [38] was selected as it can detect variant clusters (See Subsection 3.5.2 for more detail).

The KEGG database contains 11,149 compounds, reactions of compounds, enzymes, pathways, etc. (see [158] for more detail).

5.5.2 Clustering results for the whole database

With the threshold similarity degree of 0.5, 2629 clusters were found and 1261 of them were removed as they contain a single compound.

It was induced from clustering results that compounds in the same clusters are strongly alike in structures. Figure 5.4 shows three sample structures of cluster 1: the structures of Cinnamoyl-CoA, Malonyl-CoA Malonyl coenzyme A and Feruloyl-CoA trans-Feruloyl-CoA. In the five largest clusters, the common structure of each (see Figures 5.5, 5.6, 5.7, 5.8, and 5.9) is little different from their original compounds. Also, compounds in the same cluster share common names. This explains why compounds in the same cluster share common properties. As an example, compounds in Cluster 1 have the common name CoA (Coenzyme A). Thus the possess the properties of Coenzyme A such as being required to metabolize fat, carbohydrate and protein and convert them into energy at the cellular level, or being the initiation of the body's energy cycle. This helps to reason why this cluster strongly associates with carbon hydrate, lipid and amino acid metabolism pathways.

Table 5.2: Common formula, names, etc. of the five largest clusters

No.	Size	Com. formula	common name	description of member	KEGG pathways map numbers					
					C	L	AA	BX	second AtR	P&NP
1	188	$C_{22}O_{17}N_6P_3S$	CoA	Coenzyme A	640, 650	62, 71, 120	280	632		
2	115		rna	Ribonucleotid			251, 252, 260, 450		970	
3	98	$C_{19}O$	one	cyclopenta[a] phenanthrene		140, 150				
4	82	$C_9O_{12}P_2$	dp-, ose	pyran, diphosphate, methyl cyclopenta	51, 500, 520, 530				521	522
5	61	C_6O	benz	containing benzene ring			380	362, 632, 623	622	

C: Carbon hydrate Metabolism; L: Lipid Metabolism; AA: Amino Acid Metabolism; BX: Biodegradation of Xenobiotics; Second: Biosynthesis of Secondary Metabolites; AtR: Genetic Information Processing (Translation);P&NP: Biosynthesis of Polyketides and Nonribosomal Peptides

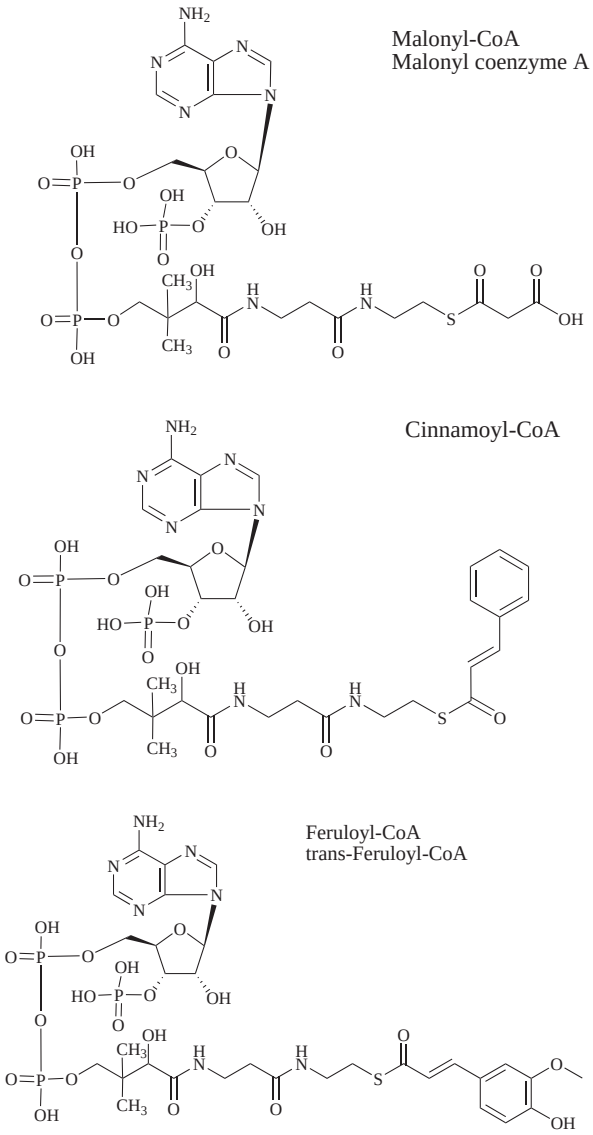


Figure 5.4: Structures of three compounds of cluster 1

With a deeper analysis of the relation between compound clusters and pathways, compound clusters were found to be associated with specific pathways in the KEGG database. For example, Cluster 1 has 28 compounds taking part in *Fatty acid biosynthesis (path 2)*(map00062), Cluster 2 has 40 compounds joining *Aminoacyl-tRNA biosynthesis*(map00970). In addition, it can be seen from Table 5.2 that each cluster associates with certain classes of pathways. For instance, compounds in Cluster 3 strongly associate with *Lipid Metabolism*(map00140, map00150), or compounds in Cluster 2 are assigned mainly to *Amino Acid Metabolism* and *Aminoacyl-tRNA biosynthesis* of genetic information processing.

Moreover, compounds in the same clusters share the same groups of enzymes working on specific radicals in compounds, accordingly catalyzing the reactions they

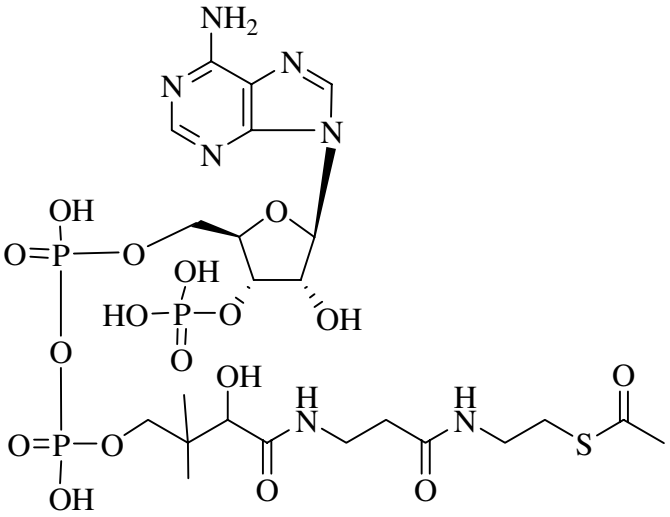


Figure 5.5: The common structure of compounds of Cluster 1: $C_{22}O_{17}N_6P_3S$, CoA

join. For example, compounds in cluster 2 use enzymes of EC 6.1.1 (Ligases Forming Aminoacyl-tRNA and Related Compounds) which mainly catalyze reactions that *rna* compounds take part in. Other introduced groups of enzymes also works on radicals that each cluster's common structure carries (Table 5.3).

In short, compounds in the same clusters not only share common structures and names but also strongly associate with specific pathways, mainly metabolic pathways, and share common groups of enzymes catalyzing their reactions.

5.5.3 Analysis on clustering results of pathway oriented databases

Clustering analysis of the whole database shows a tendency of similar structures to be assigned to specific pathways. Thus, the clustering of compounds along the pathway maps provided by KEGG is an important step to learn more about the metabolic pathways and predict possible operon structures [151].

This part analyzes the result of clustering compounds and the correlation between compound clusters and enzyme clusters within metabolic pathways. Due to space limit, the analysis result on one pathway (pathway map00860) was given as an example. The analysis of other pathways can be downloaded at www.jaist.ac.jp/q̃uang/chemical/PathwayAnalysis/

Clustering of compounds

The result of clustering similar compounds on the pathway maps shows there is a clear tendency of similar structure compounds to take up adjacent positions in reaction steps

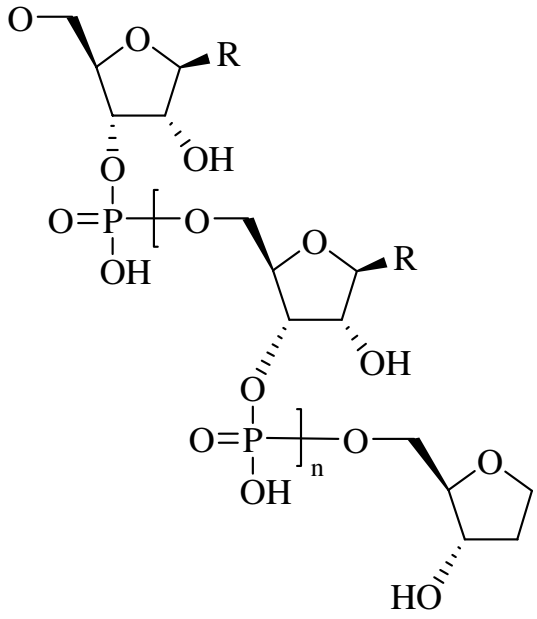


Figure 5.6: The common structure of compounds of Cluster 2: rna

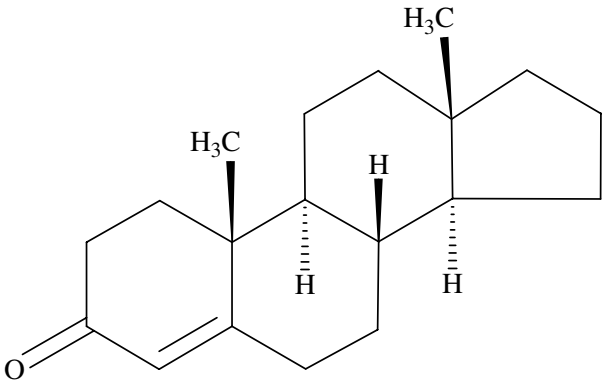


Figure 5.7: The common structure of compounds of Cluster 3: C_{19}, one

of the maps. As a result, the pathway maps are divided into several parts depends on the chemical compounds achieved in each cluster. For example, the clustering compounds on the pathway map00860 (*Porphyrin and chlorophyll metabolism*-Fig. 5.10) identifies 5 noticeable compound clusters as areas enclosed by thinner line, named C1 to C5.

Correlation of enzyme clusters and compound clusters

To find out about the relation between chemical information and genomic information, it is necessary to discover the correlation of compound clusters and enzyme clusters on the metabolic pathways. The enzyme clusters are derived from the ortholog table [159,

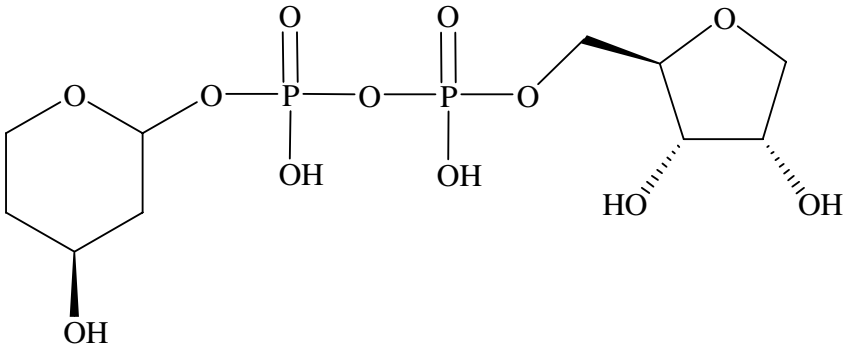


Figure 5.8: The common structure of compounds of Cluster 4: $C_9H_{12}P_2$, dp-, ose

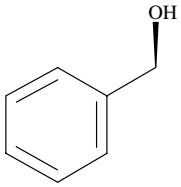


Figure 5.9: The common structure of compounds of Cluster 5: C_6O, Benz

160] which contain the information about orthologous sets of enzyme genes. Analysis of the correlation between compound clusters and enzyme clusters helps to predict possible operon-like structures in selected genomes [151].

The most surprising discovery was achieved when examining the pathway oriented clustering is that clusters of compounds and enzymes often overlap mostly each other on the pathway maps. For instance, in Figure 5.10, the area of C3 overlaps most of that of E1. The intersection of compound clusters and enzyme clusters helps to point out the operon-like structures, e.g. in the intersection of enzyme clusters with C4, the possible operon-like structure(such as in *Pseudomonas aeruginosa*) consists of E2.5.1.17, E6.3.5.10 E6.3.1.10, E2.7.1.156, E2.7.7.62, E2.7.8.26, and another operon-like structure (such as in *Mycobacterium tuberculosis H37Rv*) consisting of E2.1.1.130, E1.14.13.83, E2.1.1.131 (CbiG), E2.1.1.133 is found within C3. Other possible operon-like structures are shown in Table 5.4.

In brief, the clustering of compounds on pathway maps reveals the tendency of similar compounds to take up adjacent steps of reactions on the pathways. Besides, it shows that a set of enzyme genes encoded in an operon often corresponds to a set of enzymes catalyzing successive reaction steps (where compounds in the clusters are nodes) in a specific metabolic pathway. This encourages the new way of discovering

Table 5.3: Compound clusters with their main enzyme requirements in related reactions

Cluster ID	EC number	Fre.	Functions
Cluster 1	EC 1.1	31	Acting on the CH-OH group of donors
	EC 1.2	35	Acting on the aldehyde or oxo group of donors
	EC 1.3	68	Acting on the CH-CH group of donors
	EC 2.3	210	Acyltransferases
Cluster 2	EC 6.1.1	23	Ligases Forming Aminoacyl-tRNA and Related Compounds
Cluster 3	EC 1.1	21	Acting on the CH-OH group of donors
	EC 1.14	18	Acting on paired donors, with incorporation or reduction of molecular oxygen
Cluster 4	EC 1.1	18	Acting on the CH-OH group of donors
	EC 2.4	203	Glycosyltransferases
Cluster 5	EC 1.14	27	Acting on paired donors, with incorporation or reduction of molecular oxygen

knowledge on genome by analyzing structural similarity of chemical compounds.

5.6 Conclusion

In this chapter we presented applications of similarity measures when applying to clustering and classification. Experiments with classification and clustering for real-life databases show the merit of similarity measures. Experiment results show that the nonoverlap connected subgraph-based measure clearly boosts the accuracy of NN in comparing with the similarity measures that are based on the maximum common subgraph. Moreover, clustering for more than eleven thousand compounds in database KEGG/LIGAND discovered (revealed) compound clusters with similar structures that share the same common names, take part in the same pathways with the same requirement of enzymes in reactions. Analysis on clustering results of pathway oriented databases showed that clusters of compounds and clusters of enzymes on the same pathway have a tight relation. This encourages the new way of discovering knowledge on genome by analyzing structural similarity of chemical compounds.

Table 5.4: Possible operon-like structure from KEGG Pathway map00860

Cluster area	Possible operon
Cluster 1	E4.1.1.37, E1.3.3.3
Cluster 2	E6.6.1.1, E2.1.1.11
Cluster 2	E1.3.1.33
Cluster 3	E2.1.1.130, E1.14.13.83, E2.1.1.131 (CbiG), E2.1.1.133, E2.1.1.152, E1.3.1.54, E2.1.1.132 (CbiD), E5.4.1.2, E6.3.5.9, E6.3.1.-, E6.6.1.2
Cluster 3	E1.3.1.-, E4.99.1.-
Cluster 4	E2.5.1.17, E6.3.5.10, E6.3.1.10, E2.7.1.156, E2.7.7.62, E2.7.8.26
Cluster 5	E3.1.3.73, E2.4.2.21

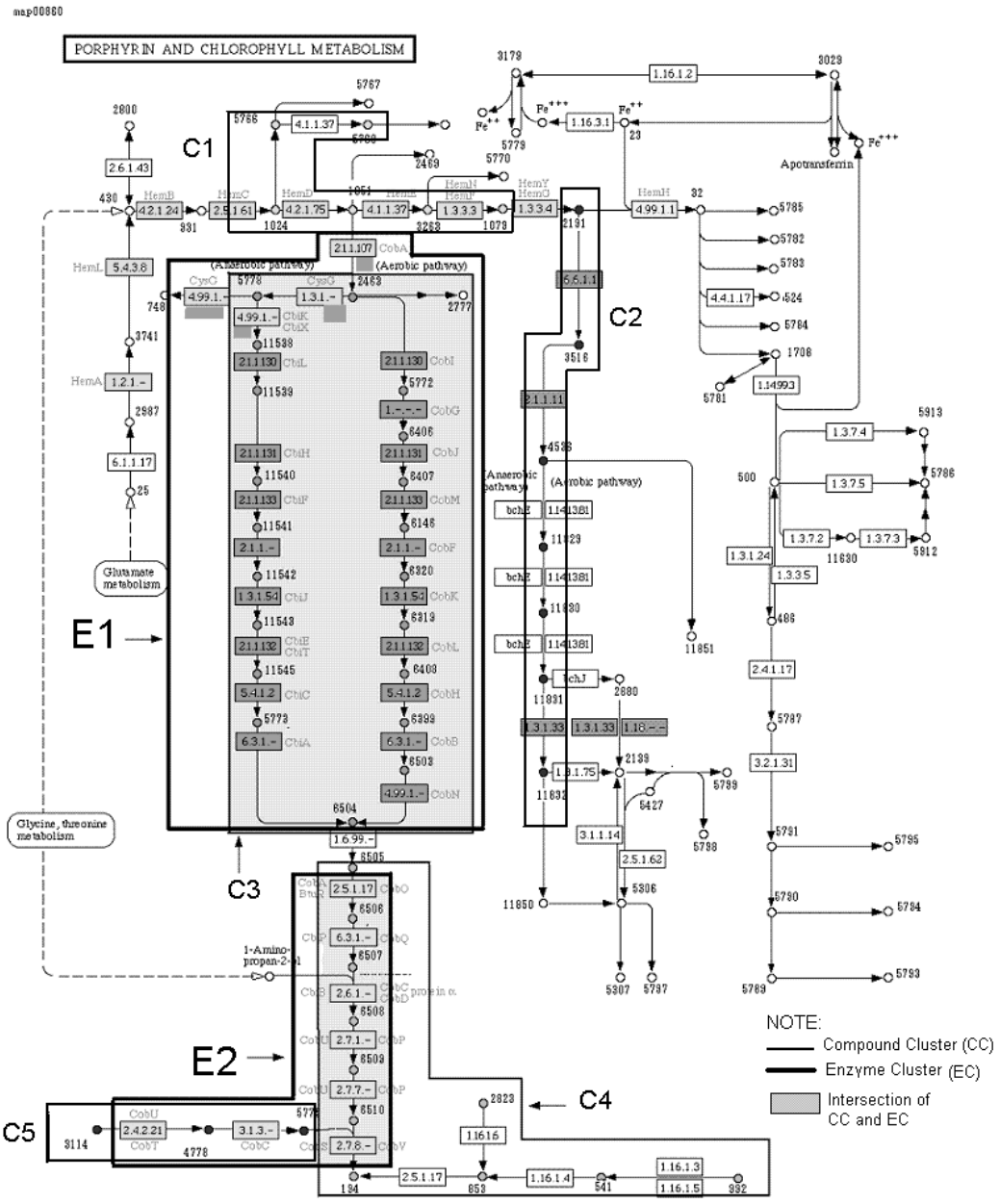


Figure 5.10: Example of compound/enzyme clusters in pathway oriented

Chapter 6

Conclusion

As an aid to the reader, this final chapter restates the research problem and reviews the major methods used in this study. The major part of this chapter summarizes the results and discusses their implications.

6.1 Summary and review

6.1.1 Problems

The importance of similarity in our daily life is often underestimated, but it is clearly pointed out in cognitive sciences, including psychological and philosophical aspects. Not only that the main inspiration for similarity in computer science is the research done in psychology, but there are also parallels of the way information has to be processed based on similarity by computers and humans. In computer science, distance function design remains at the core of many important data mining applications. Most applications such as clustering, classification and nearest neighbor searching use distance functions as a key subroutine in their implementation. Clearly, the quality of the resulting distance function significantly affects the success of the corresponding application in finding results.

Even thought Similarity measures/distances for standard data have been studied for longtime, the similarity measure problem is still opened for categorical data, heterogenous data and graph data. Due to special properties and characteristics of complex data such as poor structures, heterogeneity or complex structure, similarity measures for these data have special requirements that lead to difficulty in estimating similarity of objects of these data. In this dissertation, we reported and introduced similarity measures for these data.

6.1.2 Summary

In chapter 2, we summarized employed similarity measures for binary vectors for categorical data. Properties and characteristics were investigated to see their advantages and disadvantages. A new similarity measure that based on relations between attributes was introduced. Experiments with the nearest neighbor classification that show the merit of the similarity measures were described.

In chapter 3, we reported similarity measure for heterogenous data including Gowda and Diday methods and Minkowski generated metrics, and introduced a framework, ordered probability-based similarity measure, that is based on probability distributions and order relations. Applications with clustering applying to real-life datasets that showed the merits of the ordered probability-based similarity measure were presented.

In chapter 4, we showed similarity measures for graph data including the ϕ distance, the Measure of Papadopoulos and Manolopoulos, similarity based on the maximal common subgraph, and the edit distance. A new similarity measures, the nonoverlap connected subgraph-based measure, that based on nodes, edges, and connectivity of subgraphs were introduced.

In chapter 5, we reported experiments when applying similarity measures of graph data to 2D chemical structures. Comparisons of the results when applying similarity measures based on the maximal common subgraph and that when applying the nonoverlap connected subgraph-based measure with the nearest neighbor classification were presented. Then, experiments with clustering for the Ligand database were carried out.

6.1.3 Reviews

Categorical data

Employed similarity measures for binary vectors are fast to computer, simple and comprehensive for humans. However, the simple similarity between two values, 0 or 1 when depending on the fact if they are are identical or not, makes these similarity measures become poor in variance. The association-based measure that estimates similarity between values based on similarity between their associated values makes similarity between two values more various. Experiments when applying this measure and similarity measures for binary vectors to the nearest neighbor classification show that this measure has some advantages in boosting accuracy. However, this measure cannot be applied to databases whose attributes are absolutely independent.

Heterogenous data

The similarity measures of Gowda and Diday, and that of the framework Minkowski metric use the same strategy that estimate similarity between values of different data types based on common factors/aspects. Thus, they face with the problems of determining factors/aspects that are suitable for all data types. Ordered probability-based similarity measure avoids the problems by estimating similarity between values based on order relations and probability distributions. However, this measure is faced with time consumption problems as the complex of estimating similarity between values is $O(N^2)$ where N denotes is a database size.

Graph data

Each of the similarity measure presented in Chapter 4 estimates the similarity between these two graphs based on particular aspect of two graphs. The aspects are often the common parts or related to common parts of two graphs. The proposed similarity measure, Nonoverlap connected subgraph-based measure, is somehow suitable for areas when nodes, edges, and connectivity are all-important, i.e. chemical areas. Experiments with the nearest neighbor classification show that this measure has advantage in boosting accuracy in comparing with similarity measures based on the maximal common subgraph. Besides, experiments when applying this measure to clustering also discovered surprised relations between structures and enzymes.

6.2 Further study

Many problems have been revealed after this dissertation.

6.2.1 Categorical data

Three questions of similarity measures for categorical data are still opened:

- Since the association-based measure is limited to databases whose attribute are dependent on each others, there is a need of expanding or adapting this measure to databases whose attributes are absolutely independent.
- There should be evaluations when applying the similarity measures for different problems such as clustering and searching.
- The idea of using association between attributes to estimate the similarity can be also applied to mixed or heterogenous data as it can overcome the difference

between data types. However, how to apply this measure to different data types is still an opening question.

6.2.2 Heterogenous data

The main problem of algebra-based approaches is to determining factors/aspects that are suitable for all data types. Some views of these factors and aspects have been introduced but there are still limitations when applying to real-life databases. Thus,

- There should be evaluations of these views when using with clustering, classification, searching, etc. to see their advantages and disadvantages.
- There should be more investigations to discover more suitable factors/aspects.

For the ordered probability-based similarity measure, the problem lies in time consumption to estimate the similarity between values. Algorithms for effectively estimating similarity between values of different data types are essentially required. Besides, there should be investigations to see the affects of order relations in this measure.

6.2.3 Graph data

Complexity problem

Complexity is the main problem for most similarity measures of graph data. Most of the similarity measure meets an NP-Hard problem when estimating the similarity score between graphs. Hence, to apply the similarity measures to large databases, it requires approximated algorithms for the NP-Hard problems. Investigations and studies of the approximated algorithms are therefore necessary. Surveys for existing algorithms with their advantages and disadvantages are essentially important to users.

Applications to other fields

Experiments show the advantage of the nonoverlap connected subgraph-based measure in classification and clustering for chemical structure data. However, there is a need of applying it to other areas such as image processing and protein structures to see how useful this measure is in the real-life. More over, it would be useful if there is a survey that reports advantages and disadvantages of similarity measures when applying to different fields. This survey will help users save their time to choose right measures for their particular purposes.

Bibliography

[1] Woodford K. and Jackson J., editors. *Cambridge Advanced Learner's Dictionary*. Cambridge University Press, 2003.

[2] D. Thompson, editor. *The Concise Oxford Dictionary of Current English*. Oxford University Press, ninth edition, 1995.

[3] M. William, editor. *The American Heritage Dictionary of the English Language*. Houghton Mifflin, Boston, third edition, 1996.

[4] E.W. Weisstein. *The CRC Concise Encyclopedia of Mathematics*. CRC Press, 2000 N.W.Corporate Blvd., Boca Raton, FL 33431-9868, 1999.

[5] W. Rudin. *Principles of Mathematical Analysis*. New York: McGraw-Hill, 1976.

[6] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceeding of the ACM SIGMOD Intl.Conf.on Management of data*, pages 47–57, Boston, MA,, 1984.

[7] T.K. Sellis, N. Roussopoulos, and C. Faloutsos. The r+-tree: A dynamic index for multi-dimensional objects. In Stocker P.M. and Kent W., editors, *Proc.of the 13th Int.Conf.on Very Large Data Bases, VLDB87*, pages 507–518, Los Altos, CA, 1987. Morgan Kaufmann Publishers.

[8] J. Nievergelt, H. Hinterberger, and K.C. Sevcik. The grid file: An adaptable, symmetric multikey file structure. *ACM Transactions on Database Systems*, 9(1):38–71, 1984.

[9] J. Orenstein. A comparison of spatial query processing techniques for native and parameter spaces. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 19(2):343–352, June 1990.

[10] J. Kittler. Feature selection and extraction. In Young T.Y. and Fu K.S., editors, *Handbook of Pattern Recognition and Image Processing*, pages 59–83, Orlando, FL, 1986. Academic Press.

[11] H.V. Jagadish. A retrieval technique for similar shapes. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 20(2):208–217, June 1991.

[12] C. Faloutsos and K.-I. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia data-mining and visualization of traditional and multimedia. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 163–174, May 1995.

[13] L. Jin, C. Li, and S. Mehrotra. Efficient record linkage in large data sets. In *Eighth International Conference on Database Systems for Advanced Applications (DASFAA 03)*. IEEE Computer Society, March 2003.

[14] W. James. *The Principles of Psychology*. Holt, New York, 1890.

[15] R.L. Goldstone. *The MIT Encyclopedia of the Cognitive Sciences*. MIT, 1999 Press.

[16] W.R. Hamilton. *The Mathematical Papers of Sir William Rowan Hamilton*, volume 4. University Press, Cambridge, 2002.

[17] M. Ester, . Kriegel, H.P, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Evangelos Simoudis, Jiawei Han, and Usama Fayyad, editors, *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, Oregon, 1996. AAAI Press.

[18] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *Proceedings 1994 ACM SIGMOD Conference, Mineapolis, MN*, pages 419–429, 1994.

[19] B.K. Yi, H. V. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *ICDE*, pages 201–208, 1998.

[20] E.J. Keogh. Fast similarity search in the presence of longitudinal scaling in time series databases. In *ICTAI*, pages 578–584, 1997.

[21] S.W. Kim, S. Park, and W.W. Chu. An index-based approach for similarity search supporting time warping in large sequence databases. In *ICDE*, pages 607–614, 2001.

[22] R. Agrawal, C. Faloutsos, and A.N. Swami. Efficient similarity search in sequence databases. In D. Lomet, editor, *Proceedings of the 4th International*

Conference of Foundations of Data Organization and Algorithms (FODO), pages 69–84, Chicago, Illinois, 1993. Springer Verlag.

[23] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3/4):231–262, 1994.

[24] E. G. M. Petrakis and C. Faloutsos. Similarity searching in large image databases. Technical Report CS-TR-3388, University of Maryland, 1994.

[25] Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*, 1993.

[26] S. Prabhakar, D. Agrawal, and A.E. Abbadi. Efficient disk allocation for fast similarity searching. In *ACM Symposium on Parallel Algorithms and Architectures*, pages 78–87, 1998.

[27] H. Ferhatosmanoglu and A.E. Agrawal, D.and Abbadi. Optimal partitioning for efficient I/O in spatial databases. *Lecture Notes in Computer Science*, 2150:889–??, 2001.

[28] D. Shasha, J.T.L. Wang, and R. Giugno. Algorithmics and applications of tree and graph searching. In *Symposium on Principles of Database Systems*, pages 39–52, 2002.

[29] K. Bryson, M. Luck, M. Joy, and D. T. Jones. Applying agents to bioinformatics in geneweaver. In *Cooperative Information Agents*, pages 60–71, 2000.

[30] Gionis A., Indyk P., and Motwani R. Similarity search in high dimensions via hashing. In *The VLDB Journal*, pages 518–529, 1999.

[31] E. Bertino, A.A. Saad, and M.A. Ismail. Clustering techniques in object bases: A survey. *Data Knowl. Eng.*, 12(3):255–275, 1994.

[32] Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[33] J. Han and M. Kamber. *Data mining: concepts and techniques*. Data Management Systems. Morgan Kaufmann, 2000.

[34] J. MacQueen. Some methods for classification and analysis of multivariate observation. In *Proceedings 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[35] L. Kaufmann and P.J. Rousseeuw. Clustering by means of medoids. *Statistical Data Analysis based on the L1 Norm*, pages 405–416, 1987.

[36] M. Jambu. *Classification automatique pour l'analyse des données*. Dunod Paris, 1978.

[37] P.H.A. Sneath. The application of computers to taxonomy. *Journal of general microbiology*, 17:201–226, 1957.

[38] R.R. Sokal and C.D. Michener. Statistical method for evaluating systematic relationships. *University of Kansas science bulletin*, 38:1409–1438, 1958.

[39] L.L McQuitty. Expansion of similarity analysis by reciprocal pairs for discrete and continuous data. *Education and Psychological measurements*, 27:253–255, 1967.

[40] L.L McQuitty. Hierarchical linkage analysis for the isolation of types. *Education and Psychological measurements*, 20:55–67, 1960.

[41] L. Fu, G. H.L. Dion, and S. S.B. Foo. The effect of similarity measures on the quality of query clusters. *Journal of Information Science*, 30(5):396–407, 2005.

[42] Y.C. Martin, J.L. Kofron, and L.M. Traphagen. Do structurally similar molecules have similar biological activity ? *Journal of Medicinal Chemistry*, 45(19):4350–4358, 2002.

[43] A. Schuffenhauer, P. Floersheim, P. Acklin, and E. Jacoby. Similarity metrics for ligands reflecting the similarity of the target proteins. *Journal of Chemical Information and Computer Sciences*, 43:391–405, 2003.

[44] T.M. Cover and P.E Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, 1967.

[45] B.V. Dasarathy. *Data mining tasks and methods: Classification: nearest-neighbor approaches*. Oxford University Press, Inc., New York, NY, USA, 2002.

[46] H. Alt. The nearest neighbor. In *Computational Discrete Mathematics*, pages 13–24, 2001.

[47] Klaus Hinrichs, Jürg Nievergelt, and Peter Schorn. A sweep algorithm and its implementation: The all-nearest-neighbors problem revisited. In *WG*, pages 442–457, 1988.

[48] L. Lee. Distributional similarity models: Clustering vs. nearest neighbors. In *ACL*, 1999.

[49] T. Roos. k-nearest-neighbor voronoi diagrams for sets of convex polygons, line segments and points. In *WG*, pages 330–340, 1989.

[50] S. Cost, S.and Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10(1):57–78, 1993.

[51] R. Alonso, J.A. Bloom, H. Li, and C. Basu. An adaptive nearest neighbor search for a parts acquisition eportal. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 693–698, New York, NY, USA, 2003. ACM Press.

[52] K.M. Ting. Discretisation in lazy learning algorithms. *Artif. Intell. Rev.*, 11(1-5):157–174, 1997.

[53] E. Kushilevitz, R Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 614–623, New York, NY, USA, 1998. ACM Press.

[54] C.X. Ling and H. Wang. Computing optimal attribute weight settings for nearest neighbor algorithms. *Artificial Intelligence Review*, 11(1-5):255–272, 1997.

[55] C.G. Atkeson, A.W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1-5):11–73, 1997.

[56] R. N. Shepard. Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space. *Psychometrika*, 22(4):325–345, 1957.

[57] R.D. Luce. *Detection and recognition. In Handbook of mathematical psychology.* New York: Wiley., luce, r.d. and bush, r.r. and galanter, e. edition, 1963.

[58] J.E.K. Smith. *Models of identification. In Attention and performance*, volume VIII. Hillsdale, NJ: Erlbaum, nickerson, r. edition, 1980.

[59] J.E.K. Smith. Recognition models evaluated: A commentary on keren and baggen. *Percept. Psychophys.*, 31:183–89, 1982.

[60] J.T. Townsend. Theoretical analysis of an alphabetic confusion matrix. *Percept. Psychophys.*, 9:40–50, 1971.

[61] J.T. Townsend and D.E. Landon. An experimental and theoretical investigation of the constant-ratio rule and other models of visual letter confusion. *J. Math. Psychol.*, 25:119–162, 1982.

[62] R.M. Nosofsky. Overall similarity and the identification of separable-dimension stimuli:a choice model analysis. *Percept. Psychophys*, 38:415–432, 1985.

[63] R.M. Nosofsky. Relations between exemplar-similarity and likelihood models of classification. *J. Math. Psychol.*, 34:393–418, 1990.

[64] W.K. Estes. Array models for category learning. *Cognitive Psychology*, 18:500–549, 1986.

[65] D.L. Hintzman. Schema abstraction in a multiple-trace memory model. *Psychol. Rev.*, 93:411–428, 1986.

[66] D. L. Medin and M. M. Schaffer. Context theory of classification learning. *Psychological Review*, 85:207–238, 1978.

[67] R. M. Nosofsky. Attention, similarity, and the identification-categorizat relationship. *Journal of Experimental Psychology: General*, 115(1):39–57, 1986.

[68] R.M. Nosofsky. Exemplar-based accounts of relations between classification, recognition, and typicality. *J. Exp. Psychol.: Learn. Mem. Cognit.,*, 14:700–708, 1988.

[69] R.M. Nosofsky. Tests of an exemplar model for relating perceptual classification and recognition memory. *J. Exp. Psychol.: Hum. Percept. Perform.*, 17:3–27, 1991.

[70] R. M. Nosofsky, S. E. Clark, and H. J. Shin. Rules and exemplars in categorization, identification and recognition. *Journal of Experimental Psychology:Learning, Memory, and Cognition*, 15:282–304, 1989.

[71] G. Gillund and R.M. Shiffrin. A retrieval model for both recognition and recall. *Psychol. Rev.,*, 91:1–67, 1984.

[72] P. Jaccard. The distribution of the flora of the alpine zone. *New phytologiest*, 11:37–50, 1912.

[73] L. R. Dice. Measures of the amount of ecological association between species. *Ecology*, 26:297–302, 1945.

[74] S.Q. Le and T.B Ho. Conditional probability distribution-based dissimilarity measure for categorical data. In *8th Pacific-Asia Conference on Knowledge Discovery and Data Mining PAKDD*, pages 580–589. Lecture Notes in Artificial Intelligence, LNAI 3056, Springer, 2004.

[75] S.Q. Le and T.B Ho. An association-based dissimilarity measure for categorical data. *Pattern Recognition Letters*, 2005.

[76] S.Q. Le and T.B Ho. Measuring the similarity for heterogenous data: An ordered probability-based approach. In *Discovery Science DS'04*, pages 129–141. Springer LNAI 3245, 2004.

[77] S.Q. Le, T.B. Ho, and T.T.H. Phan. A novel graph-based similarity measure for 2d chemical structure. In *Genome Informatics*, volume 14, pages 82–91, 2004.

[78] A.M Liebetrau. *Measures of association*. Newbury Park, CA: Sage, 1983.

[79] F.B Baulieu. Classification of presence/absence based dissimilarity coefficients. *Journal of Classification*, 6(6):233–246, 1989.

[80] J.C Gower. Coeffecients of association and similarity, based on binary (presence-absence) data: an evaluation. *Biometrics*, 27:857–871, 1971.

[81] J.C Gower and P. Legendre. Metic and euclidean properties of dissimilarity coefficients. *Journal of classification*, 3:5–48, 1986.

[82] D.H. Krantz, R.D. Luce, P. Suppes, and A. Tversky. *Foundations of Measurement*, volume I. New York: Academic Press, 1971.

[83] M. Albert. *Measures of Association*, volume 32 of *Quantitative Applications in the Social Sciences*. Newbury Park, CA: Sage, 1983.

[84] V. Batagelj and M. Bren. Comparing resemblance measures. *Journal of Classification*, 12(1), 1995.

[85] Z. Hubálek. Coefficients of association and similarity, based on binary (present-absence) data: an evaluation. *Biological Review*, 57(57):669–689, 1982.

[86] K.C. Gowda and E. Diday. Symbolic clustering using a new dissimilarity measure. *In Pattern Recognition*, 24(6):567–578, 1991.

[87] K.C. Gowda and E. Diday. Unsuppervised learning throught symbolic clustering. *In Pattern Recognition lett.*, 12:259–264, 1991.

[88] K.C. Gowda and E. Diday. Symbolic clustering using a new similarity measure. *IEEE Trans. Syst. Man Cybernet*, 22(2):368–378, 1992.

[89] F.A.T. de Carvalho. Proximity coefficients between boolean symbolic objects. In Diday E.et.al., editor, *New Approaches in Classification and Data Analysis*, volume 5 of *Studies in Classification, Data Analysis, and Knowledge Organisation*, pages 387–394. Springer-Verlag, Berlin, 1994.

[90] F.A.T. de Carvalho. Extension based proximity coefficients between constrained boolean symbolic objects. In Hayashi C.et al., editor, *IFCS96*, pages 370–378. Springer, Berlin, 1998.

[91] D.W. Goodall. A new similarity index based on probability. *Biometrics*, 22:882–907, 1966.

[92] M. Ichino and H. Yaguchi. Generalized minkowski metrics for mixed feature-type data analysis. *IEEE Transactions on Systems Man, and Cybernetics*, 24(4), 1994.

[93] H.O. Lancaster. The combining of probabilities arising from data in discrete distributions. *Biometrika*, 36:370–382, 1949.

[94] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[95] Z. Rached, F. Alajaji, and L.L. Campbell. Rényis divergence and entropy rates for finite alphabet markov sources. *IEEE Transactions on Information theory*, 47(4):1553–1561, 2001.

[96] S. Kullback. *Information theory and statistics*. John Wiley and Sons, New York, 1959.

[97] S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[98] C.L. Blake and C.J. Merz. (uci) repository of machine learning databases, 1998.

[99] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Knowledge Discovery and Data Mining*, pages 80–86, 1998.

[100] T.C. Fogarty. First nearest neighbor classification on frey and slates letter recognition problem. *Machine Learning*, 9(4):387–388, 1992.

[101] Anderberg M.R. *Clustering analysis for applications*. NewYork: Academic, 1973.

[102] S. Geist, K. Lengnink, and R. Wille. An order-theoretic foundation for similarity measures. In Diday E. and Lechevallier Y., editors, *Ordinal and symbolic data analysis, studies in classification, data analysis, and knowledge organization*, volume 8, pages 225–237, Berlin, Heidelberg, 1996. Springer.

[103] R.A Fisher. *Statistical methods for research workers*. Oliver and Boyd, 11th edition, 1950.

[104] S.A Stouffer, E.A Suchman, L.C Devinney, and R.M Williams. Adjustment during army life. *The American Solder*, 1, 1949.

[105] G.s Mudholkar and E.O George. The logit method for combining probabilities. In J.Rustagi, editor, *Symposium on Optimizing methods in statistics*, pages 345–366. Academic press, NewYork, 1979.

[106] G.N. Lance and W.T. Williams. A generalised sorting strategy for computer classifications. *Nature*, pages 212–128, 1966.

[107] G.N. Lance and W.T. Williams. A general theory of classificatory sorting stragies i hierarchical systems. *Computer journal*, 9:373–380, 1967.

[108] L.L McQuitty. Similarity analysis by reciprocal pairs for discrete and continuous data. *Education and Psychological measurements*, 26:825–831, 1966.

[109] J. Podani. New combinatorial clustering methods. *Vegetatio*, 81:61–77, 1989.

[110] J.D.J. Ward. Hierarchical grouping to optimize an objective function. *Journal of the american statistical association*, 58:236–244, 1963.

[111] D. Wishard. An algorithm for hierarchical classifications. *Biometrics*, 25:165–170, 1969.

[112] J.C. Gower. A comparison of some methods of cluster analysis. *Biometrics*, 23:623–638, 1967.

[113] B. Berendt, A. Hotho, and G. Stumme. Towards semantic web mining. In *International Semantic Web Conference*, pages 264–278, 2002.

[114] Tom Blank. Behavioral modeling for system design (panel). In *DAC*, page 196, 1988.

[115] W. Eberle, G. Vandersteen, S. Wambacq, P.and Donnay, G.G. E. Gielen, and Hugo De Man. Behavioral modeling and simulation of a mixed analog/digital automatic gain control loop in a 5 ghz wlan receiver. In *DATE*, pages 10642–10649, 2003.

[116] N.G. Bourbakis, A. Mogzadeh, S. J. Mertoguno, and C. Koutsougeras. A knowledge-based expert system for automatic visual vlsi reverse-engineering: Vlsi layout version. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 32(3):428–436, 2002.

[117] J.M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The web as a graph: Measurements, models, and methods. In *COCOON*, pages 1–17, 1999.

[118] A.O. Mendelzon. Review - authoritative sources in a hyperlinked environment. *ACM SIGMOD Digital Review*, 1, 2000.

[119] C.R. Palmer, P.B. Gibbons, and C. Faloutsos. Anf: a fast and scalable tool for data mining in massive graphs. In *KDD*, pages 81–90, 2002.

[120] S. Kramer, Luc De Raedt, and Helma C. Molecular feature mining in hiv data. In *KDD*, pages 136–143, 2001.

[121] M. Deshpande, M. Kuramochi, and G. Karypis. Frequent sub-structure-based approaches for classifying chemical compounds. In *ICDM*, pages 35–42, 2003.

[122] R.D. Brown and Y.C. Martin. Use of structure - activity data to compare structure-based clustering methods and descriptors for use in compound selection. *Journal of the American Chemical Society*, 36:572–584, 1996.

[123] R.D. Brown and Y.C. Martin. The information content of 2d and 3d structural descriptors relevant to ligand-receptor binding. *Journal of the American Chemical Society*, 37:1–9, 1997.

[124] G.M. Downs and P. Willett. Similarity searching in databases of chemical structures. *Reviews in Computational Chemistry*, 7:1–66, 1995.

[125] M. Hattori, Y. Okuno, S. Goto, and M. Kanehisa. heuristics for chemical compound matching. In G. Michael, K. Minoru, M. Satoru, and T. Toshihisa, editors, *Genome Informatics*, pages 144–153, 2003.

[126] G. Chartrand, G. Kubicki, and M. Schultz. Graph similarity and distance in graphs. *Aequationes Mathematicae*, 55(1-2):129–145, 1998.

[127] W.D. Wallis, P. Shoubridge, M. Kraetz, and D. Ray. Graph distances using graph union. *Pattern Recognition Letters*, 22(6/7):701–704, 2001.

[128] D. Ellis, J. Furner-Hines, and P. Willett. On the measurement of inter-linker consistency and retrieval effectiveness in hypertext databases. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 51–60. Springer-Verlag New York, Inc., 1994.

[129] Y. Takahashi, S. Maeda, and S. Sasaki. Automated recognition of common geometrical patterns among a variety of three-dimensional molecular structures. *Analytica Chimica Acta*, 200:363–377, 1987.

[130] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3):255–259, 1998.

[131] G. Levi. A note on the derivation of maximal common subgraphs of two directed or undirected graphs. *Calcolo*, 9:341–354, 1972.

[132] J.M. James. Backtrack search algorithms and the maximal common subgraph problem. *Software-Practice and Experience*, 12(1):23–34, 1982.

[133] R.U. Julian. An algorithm for subgraph isomorphism. *J.ACM*, 23(1):31–42, 1976.

[134] E. Marchiori. Genetic, iterated and multistart local search for the maximum clique problem. In *Applications of Evolutionary Computing*, volume LNCS 2279, pages 112–121. Springer, 2002.

[135] K. Kengo, H. Akihiro, and N. Hiroyuki. Solving the maximum clique problem by k-opt local search. In *The 5th Metaheuristics International Conference (MIC-2003)*, Kyoto, Japan, 2003.

[136] R. Battiti and M. Protasi. Reactive local search for the maximum clique problem. *Algorithica*, 29(4):610–637, 2001.

[137] V. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics-Doklady*, 10:707–710, 1966.

[138] R.A. Wagner and J.M Fisher. The string to string correction problem. *Journal of the ACM*, 21(1):168–173, 1974.

[139] A. Sanfeliu and K.S. Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man and Cybernetics,*, 13(3):353–362, 1983.

[140] L. Wiskott, J.M Fellous, N. Kruger, and Malsburg C. von der. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–559, 1997.

[141] E. Kubicka, G. Kubicki, and I. Vakalis. Using graph distance in object recognition. In *Proc. ACM Computer Science Conference*, pages 43–38, 1990.

[142] A. Papadopoulos and Y. Papadopoulos. Structurebased similarity search with graph histograms. In *Proc. DEXA/IWOSS Int. Workshop on Similarity Search*, pages 174–178. IEEE Computer Society Press, 1999.

[143] K. Zhang and T. Jiang. Some max snp-hard results concerning unordered labeled trees. *Information Processing Letters*, 49:249–254, 1994.

[144] P. Willet. Chemical similarity searching. *Journal of Chemical Information Computer Science*, 38:983–996, 1998.

[145] D. Weininger. Smiles, a chemical language and information system 1. introduction and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28:31–36, 1988.

[146] A. Dalby, J.G. Nourse, W.D. Hounshell, A.K. Gushurst, D.L. Grier, Leland B.A., and Laufer J. Description of several chemical structure file formats used by computer programs developed at molecular design limited. *Journal of Chemical Information and Modelling*, 32:244–255, 1992.

[147] D. Weiniger. Introduct of encoding rules. *Journal of the American Chemical Society*, 28:31–36, 1988.

[148] J.K. Wegner, H. Fröhlich, and A. Zell. Feature selection for descriptor based classification models: Part i theory and ga-sec algorithm. *Journal of the American Chemical Society*, 44:921–930, 2004.

[149] J.K. Wegner, H. Fröhlich, and A. Zell. Feature selection for descriptor based classification models: Part ii human intestinal absorption (hia). *Journal of the American Chemical Society*, 44:921–930, 2004.

[150] J.W Raymond and P. Willet. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *J.Comput.-Aided Mol.Des.*, 16:521–533, 2002.

[151] M. Hattori, Y. Okuno, S. Goto, and M. Kanehisa. Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. *Journal of the American Chemical Society*, 125(39):11853–65, 2003.

[152] J.W Raymond, E.J Gardiner, and P. Willet. Rascal: Calculation of graph similarity using maximum common edge subgraphs. *Comput.J.*, 45:631–644, 2002.

[153] J.W Raymond, E.J Gardiner, and P. Willet. Heuristics for similarity searching of chemical graphs using a maximum common edge subgraph algorithms. *Journal of the American Chemical Society*, 42:305–316, 2002.

[154] D.R Flower. On the properties of bit-string measures of chemical similarity. *Journal of the American Chemical Society*, 38:379–386, 1998.

[155] R.D. King, S.H. Muggleton, A. Srinivasan, and M.Ster. Structure-activity relationships derived by machine learning: the use of atoms and their bond connectives to predict mutagenicity by inductive logic programming. In *Proceedings of the National Academy of Sciences*, volume 93, pages 438–442, 1996.

[156] A. Srinivasan, S.H. Muggleton, M.J.E. Sternberg, and R.D. King. Theories for mutagenicity: a study in first-order and feature-based induction. *Artificial Intelligence*, 84:277–299, 1996.

[157] A.M. Richard and C.R. Williams. *QSARs of Mutagens and Carcinogens*, chapter 5, pages 151–179. CRC Press, 2003.

[158] M. Kanehisa, S. Goto, S. Kawashima, and A. Nakaya. The kegg databases at genomenet. *Nucleic Acids Res.*, 30:42–46, 2002.

[159] H. Ogata, W. Fujibuchi, S. Goto, and M. Kanehisa. A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. *Nucleic Acids Res*, 28:4021–8, 2000.

[160] W. Fujibuchi, H. Ogata, H. Matsuda, and M. Kanehisa. Automatic detection of conserved gene clusters in multiple genomes by graph comparison and p-quasi grouping. *Nucleic Acids Research*, 28:4029–36, 2000.

Appendix

K-Opt Algorithm

Procedure K-Opt

In: Subgraph $sG =< sV >$

Out: Subgraph building from sG

Begin

$AddNodes = \{1..n\} \setminus sV$, $RemoveNode = sV$

$Next = Connect(sV, AddNodes)$

$Out = sV$

while ($RemoveNode \neq \emptyset$ or $Next \neq \emptyset$) **do**

if ($Next \neq \emptyset$) **then**

$v^* = \arg\max_{v \in Next}\{degree(v, Next)\}$

$sV = sV + \{v^*\}$

$AddNode = AddNode \setminus \{v^*\}$

$Next = Connect(sV, AddNodes)$

if ($|Out| < |sV|$) **then**

$Out = sV$

end if

else

$v^* = \arg\max_{v \in RemoveNode}\{|Connect(sV \setminus \{v\}, AddNodes|)\}$

$sV = sV \setminus \{v^*\}$

$Next = Connect(sV, AddNodes)$

$RemoveNode = RemoveNode \setminus \{v^*\}$

end if

end while

return Out

End

Figure 6.1: K-Opt

Procedure MaximumCompleteSubgraph

In: $G =< V, E >$

Out: Max Complete subgraph

Begin

$CurrentSubgraph = \emptyset$

$stop$ = false;

while (!stop) **do**

$NewSubGraph = k - opt(CurrentSubgraph)$

if $(|NewSubGraph| > |CurrentSubGraph|)$ **then**

$CurrentSubgraph = NewSubgraph$;

else

$stop = true$

end if

end while

return $CurrentSubgraph$

End

Figure 6.2: Algorithm for determining the maximum complete subgraph of graph G (K-Opt)

Procedure Connect

In: sub set sV of V

Out: Set of nodes in *Addnodes*, connected to all nodes in sV.

Begin

$Out = \emptyset$

for $v \in AddNodes$ **do**

if $(e(v, v') : \forall v' \in sV)$ **then**

$Out = Out \cup \{v\}$

end if

end for

End

Figure 6.3: Connect procedure (K-Opt)

Procedure Try

In: sV

Begin

for $(v, v') : v \in G$ and $v' \in G'$ **do**

if $(v, v'$ is isomorphic with respect to $sV)$ **then**

$sV = sV + \{(v, v')\}$

if $(|sV| > |max_s V|)$ **then**

$max_s V = sV$;

end if

Try(sV)

$sV = sV \setminus \{(v, v')\}$

end if

end for

End

Figure 6.4: Try procedure (K-Opt)