On Nonmetric Similarity Search Problems in Complex Domains

TOMÁŠ SKOPAL
Department of Software Engineering, FMP, Charles University in Prague
and
BENJAMIN BUSTOS
Department of Computer Science, University of Chile

The task of similarity search is widely used in various areas of computing, including multimedia databases, data mining, bioinformatics, social networks, etc. In fact, retrieval of semantically unstructured data entities requires a form of aggregated qualification that selects entities relevant to a query. A popular type of such a mechanism is similarity querying. For a long time, the database-oriented applications of similarity search employed the definition of similarity restricted to metric distances. Due to its topological properties, metric similarity can be effectively used to index a database which can be then queried efficiently by so-called metric access methods. However, together with the increasing complexity of data entities across various domains, in recent years there appeared many similarities that were *not* metrics – we call them *nonmetric* similarity functions. In this paper we survey domains employing nonmetric functions for effective similarity search, and methods for efficient nonmetric similarity search. First, we show that the ongoing research in many of these domains requires complex representations of data entities. Simultaneously, such complex representations allow us to model also complex and computationally expensive similarity functions (often represented by various matching algorithms). However, the more complex similarity function one develops, the more likely it will be a nonmetric. Second, we review the state-of-the-art techniques for efficient (fast) nonmetric similarity search, concerning both exact and approximate search. Finally, we discuss some open problems and possible future research trends.

Categories and Subject Descriptors: H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*indexing methods*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*retrieval models*

General Terms: design, algorithms, performance

Additional Key Words and Phrases: similarity retrieval, nonmetric distances, approximate and exact search, similarity measuring

This research has been supported in part by Czech Science Foundation project Nr. 201/09/0683 (first author), and by FONDECYT (Chile) Project 11070037 (second author). Furthermore, we would like to thank Dr. David Hoksza for his valuable comments concerning similarity modeling and search in protein databases. Last but not least, we thank the anonymous reviewers for their helpful comments on the earlier versions of this survey.

Authors' addresses: Tomáš Skopal, Department of Software Engineering, Faculty of Mathematics and Physics, Charles University in Prague, Malostranske nam. 25, 118 00 Prague, Czech Republic, skopal@ksi.mff.cuni.cz, http://siret.ms.mff.cuni.cz/skopal. Benjamin Bustos, Department of Computer Science, University of Chile, Av. Blanco Encalada 2120 3er Piso, Santiago, Chile, bebustos@dcc.uchile.cl.

1. INTRODUCTION

Since the beginning of the new millennium we have experienced a phenomenon of information explosion, where the volume of produced digital data increased exponentially in time. The growth was caused by many factors, like more powerful computing resources, high-speed internet, and diffusion of the information society all over the world. Additionally, an enormous production of data is attributed to the quick dissemination of cheap devices for capturing multimedia data like audio, video, and photography. Nowadays, more than 95% of web space is considered to store multimedia content, and more multimedia data is stored in corporate and scientific databases, personal archives, and digital libraries. In particular, 80 billions of photographs are expected to be taken every year[1], while, at the same time, the complexity of the data grows as well. Such vast amounts of complex data need to be searched both efficiently and effectively. Search engines like `images.google.com` are text-based, reusing the well-established technology of web search. However, the enormous growth of multimedia (and other complex) data and the evolving forms of data deployment show that text-based search lags behind. In particular, the multimedia data (i.e., signals, measurements, etc.) are often delivered as raw data files without any annotation needed for text-based search. Moreover, even an annotated data entity could not be successfully retrieved, because text-based retrieval is inherently imprecise, subjective, and incomplete.

The *content-based retrieval* (CBR) of multimedia data, or other semantically unstructured-type data, is often a more viable approach than the text-based search [Blanken et al. 2007; Deb 2004]. It considers retrieval according to the actual content of complex objects, rather than considering an external description (the annotation). Instead of text-based query, the database is queried by an example object to which the desired database objects should be *similar*, i.e., here the *query-by-example* retrieval scheme is adopted. The concept of pairwise similarity plays the role of a *multi-valued relevance* of every database object to a given query object. For a long time, the similarity functions have been modeled by metric distances (in the mathematical meaning), because the metric postulates allowed researchers to design efficient (fast) access methods for similarity search. However, during the last years the need for less restrictive modeling of similarity has gotten stronger justification because of higher demands on more complex similarity models. Here the restriction to metric case becomes a serious obstacle.

1.1 Paper Scope

The focus of this paper is the usage of nonmetric measures for *efficient and effective* similarity search in a wide variety of research domains. Although the term *nonmetric* simply means that similarity function does not hold some (or all) properties of a metric (see Section 2.3), it may exhibit any other properties. However, in order to reasonably discuss the topic in the limited room of a single survey, we need to

[1]According to Enterprise strategy group, `www.enterprisestrategygroup.com`, 2007

restrict the scope of the paper. In particular, this survey will *not* cover variants of the problem such as:

—*Context-dependent similarity functions.* Similarity functions could be affected by external causes, such as the time of measuring, or similarity learning. For example, in the case of time of measuring, Web search engines can use historical data to define time-dependent similarity measures [Zhao et al. 2006]. Also, the similarity function (as everything in the world) could not be perfect, thus it may be improved (learned or trained) [Mandl 1998]. For example, user feedback may be used to adapt similarity measures for content-based image retrieval [Lee and Lin 2008]. It should be mentioned that the changing semantics of a similarity function (based on context-dependent properties) has been included in the *probabilistic types* of psychological similarity theories [Ashby 1992]. In this survey, we will focus on similarity measures where some of the metric properties may not hold, but they do not depend on the context where they are used.

—*Dynamic similarity functions.* Similarity functions may be tuned with parameters given by the user or the search system. An example of dynamic similarity measures is the multi-metric approach [Bustos and Skopal 2006], where the similarity function is defined as a linear combination of several metrics. The weights for the linear combination must be set at query time either manually by the user or automatically by the search system. In a broader sense, the multi-metric approach is similar to the problem of object specificity, which plays a role in case not all the objects are treated uniformly, e.g., when one compares flowers based on colors, but cars based on shape, or when each object defines its own similarity function [Ciaccia and Patella 2009]. We do not consider this type of similarity functions, as they require a precise definition on how to select the weights/similarity depending on the given object.

In summary, in this paper we consider only similarity functions that are, say, "context-free and static" – a similarity between two objects is constant whatever the context is, i.e., regardless of time, user, query, other objects in database, etc.

1.2 Paper Contribution

In the first part of this paper we gathered a motivating collection of various domains where the nonmetric similarity search is needed. For the first time we show that the need for nonmetric search is not an artificial problem, but there are many disciplines (outside and inside computer science) for which the nonmetric search is crucial. In the second – database-oriented – part of the paper, we survey the state-of-the-art techniques for *efficient* nonmetric similarity search. The paper is organized as follows. Section 2 introduces the concept of similarity measuring, defines the task of similarity search, and discusses some properties of similarity functions. Section 3 presents some examples on general nonmetric functions and overviews a number of domains where the nonmetric similarity search found its assets. Section 4 provides a database-oriented analysis of nonmetric access methods that allow an efficient implementation of nonmetric similarity search. Section 5 concludes the paper, raising some challenges regarding nonmetric similarity search.

Table I. Notation used in this paper

Symbol	Description
$\mathbb{U}$	Universe of valid objects (descriptors)
$\mathbb{S} \subset \mathbb{U}$	Database of objects (descriptors)
$x, y, o_i \in \mathbb{U}$	Objects from $\mathbb{U}$
$q \in \mathbb{U}$	Query object from $\mathbb{U}$
$s : \mathbb{U} \times \mathbb{U} \to \mathbb{R}$	Similarity function between pairs of objects in $\mathbb{U}$
$\delta : \mathbb{U} \times \mathbb{U} \to \mathbb{R}$	Dissimilarity function between pairs of objects in $\mathbb{U}$

2. SIMILARITY MEASURING AND SEARCH

The phenomenon of similarity perceived by human has been studied for centuries, as it implicitly affects everyone's world understanding, the decision making, and any other reasoning in a fundamental way. The realm of similarity modeling were originally psychology and related disciplines. However, due to technological progress, similarity is also studied in various disciplines related to computer science, like computer vision, pattern recognition, data mining, and database systems.

As classified by psychological theories, similarity involves many aspects of the human's perception of the real world, including judged and perceived similarity concepts, deterministic and probabilistic perceptions and decisions, and so on. There is plenty of literature concerning psychological background of similarity [Santini and Jain 1999; Tversky 1977; Ashby and Perrin 1988; Ashby 1992]. In the following, without losing generalization, we narrow our discussion about similarity to the area of complex databases, where the subject of similarity evaluation is multimedia objects (e.g., images) or, more generally, any complex semantically unstructured data (e.g., time series). Table I shows the symbols that we will use through this paper.

2.1 Similarity Spaces

As discussed in Section 1.1, we restrict our attention to similarity as a function that accepts a pair of objects and returns a single real number. Formally, let s be a function assigning to a pair of objects x, y from a universe $\mathbb{U}$ a similarity value from $\mathbb{R}$, defined as

$$s : \mathbb{U} \times \mathbb{U} \mapsto \mathbb{R}.$$

Such a function s is called *pairwise similarity function*. It produces a real number representing a similarity score between two input objects from the universe. The universe $\mathbb{U}$ itself is a set of all models (descriptors) that could be derived from complex objects of a given kind where the objects induce some perceptual stimuli that need to be evaluated as similar or dissimilar. The structure of element $x \in \mathbb{U}$ could be a vector, a sequence, a string, or an (ordered) set consisting of the mentioned structures, recursively. In general, we have no requirements on the internal structure of $\mathbb{U}$, we just assume that a single descriptor in $\mathbb{U}$ corresponds to a single complex object (stimulus).

Instead of a similarity function, there is often required an inverse concept – a *dissimilarity (or distance) function* δ – where a higher dissimilarity score stands for lower similarity score, and vice versa. Hence, a dissimilarity δ equivalent to a similarity s must fulfill $s(x, y) \geq s(x, z) \Leftrightarrow \delta(x, y) \leq \delta(x, z),\ \forall x, y, z \in \mathbb{U}$. The

couple $(\mathbb{U}, s)$ is called a *similarity space* and, analogously, $(\mathbb{U}, \delta)$ is called a *dissimilarity space*. The possibility of choice between similarity and dissimilarity is important especially for practical reasons. There exist many situations where the formula/algorithm defining the function is available just in one of the two forms, while its "manual" transformation to the inverse might lead to serious obstacles (further discussed in Sections 2.4.2 and 2.4.4).

2.2 Similarity Search

At this moment, we have to detail the main subject of the paper – the similarity search. In addition to the definition of the most popular similarity query types, we also discuss the effectiveness and efficiency issues.

2.2.1 *Queries and Retrieval Modalities.* In the following, we consider the query-by-example model for similarity search: Given a database of objects $\mathbb{S} \subset \mathbb{U}$ and a query object $q \in \mathbb{U}$, the search system returns all objects from $\mathbb{S}$ that are similar to q. Let δ be a dissimilarity (distance) function. There are two typical similarity queries defined using δ:

—*Range query.* A range query (q, r), $q \in \mathbb{U}$, $r \in \mathbb{R}^+$, reports all objects in $\mathbb{S}$ that are within a distance r to q, that is, $(q, r) = \{x \in \mathbb{S} \mid \delta(x, q) \leq r\}$. The subspace $\mathbb{V} \subset \mathbb{U}$ defined by q and r (i.e., $\forall v \in \mathbb{V}\ \delta(v, q) \leq r$ and $\forall x \in \mathbb{X} - \mathbb{V}\ \delta(x, q) > r$) is called the *query ball*.

—*k nearest neighbors query (kNN).* It reports the k objects from $\mathbb{S}$ closest to q. That is, it returns the set $\mathbb{C} \subseteq \mathbb{S}$ such that $|\mathbb{C}| = k$ and $\forall x \in \mathbb{C}, y \in \mathbb{S} - \mathbb{C}, \delta(x, q) \leq \delta(y, q)$. The kNN query also defines a query ball (q, r), but the distance r to the k^{th} NN is not known beforehand.

Note that it is possible that many sets of k objects are a valid answer for the kNN search. For example, if there are two or more objects at exactly the same distance from q as the k^{th} NN, any of them can be selected as the k^{th} NN. While this is unusual when using continuous distance functions, it is frequent when using discrete distance functions.

Figure 1a illustrates a range query (q_1, r) and a kNN query $(q_2, k = 3)$ in a 2D vector space using the Euclidean distance. The answers are respectively:
$(q_1, r) = \{o_2, o_6, u_7\}$; $(q_2, k = 3) = \{o_{11}, o_{12}, o_{13}\}$.

Necessary to the query-by-example retrieval is the notion of similarity ranking. Distance δ generates a permutation of the objects $\mathbb{S}$, so-called ranking, where the objects are ordered according to their distances to q. Range queries and kNN queries return the first objects of this ranking. For range queries, the number of retrieved objects is not known a priori (it is a number between 0 and $|\mathbb{S}|$). On the contrary, for kNN queries it is known beforehand.

Another type of similarity query is the incremental search, also known as "give-me-more" query [Hjaltason and Samet 1995]. For this type of query, the search system returns a certain number of relevant objects (the first ones in their corresponding similarity ranking), and the user may request additional relevant objects (like when one performs a query in Google). Additionally, the similarity join [Zezula et al. 2005] between two sets of objects $\mathbb{A}$ and $\mathbb{B}$ is defined as the set of pairs (a, b) ($a \in \mathbb{A}$ and $b \in \mathbb{B}$) such that $\delta(a, b) \leq r$ for some tolerance value $r \geq 0$. Among other

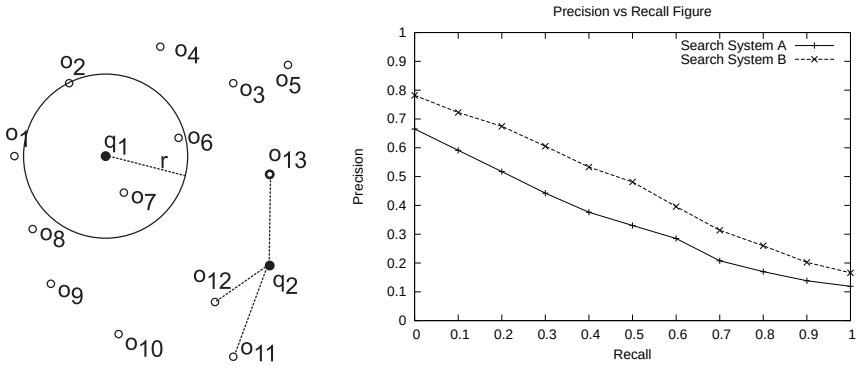

Fig. 1. (a) Range query and kNN query examples (b) Example of precision vs. recall figure

query types, we name (k-)closest pairs [Corral et al. 2000], reverse kNN queries [Tao et al. 2004], and metric skylines [Chen and Lian 2008].

2.2.2 *Effectiveness of Retrieval.* Effectiveness is related to the quality of the answer returned by a query, hence, the effectiveness measures the ability to retrieve relevant objects while at the same time holding back non-relevant ones. Indeed, improving the effectiveness of a search system is at least as important as improving its efficiency. In the task of similarity search, the dominant piece of logic determining the retrieval effectiveness is provided by the underlying (dis)similarity function. An effective distance function should treat two similar objects, according to the human concept of similarity, as two close points in the corresponding space.

There are several effectiveness measures to rate similarity search engines. A popular measure is the precision versus recall figure [Baeza-Yates and Ribeiro-Neto 1999]. Precision is the fraction of the retrieved objects relevant to a given query, and recall is the fraction of the relevant objects retrieved from the set of objects. Normalized precision versus recall figures are based on 11 standard recall levels (0%, 10%, ..., 100%). Figure 1b shows an example of a typical precision vs. recall figure. In the example, the search system B is more effective than the search system A because for each recall level the precision of system B is higher than the precision of system A.

Other widely used effectiveness measure is the R-precision [Baeza-Yates and Ribeiro-Neto 1999], which is defined by the precision when retrieving only the first R objects, with R the number of relevant objects for the issued query. The R-precision gives a single number to rate the performance of a retrieval algorithm. A similar measure is the Bull Eye Percentage, which is defined as the recall after retrieving $2R$ objects.

2.2.3 *Efficiency of Retrieval.* Efficiency is related to the cost of the similarity search (in terms of CPU and I/O time). A naïve method to answer range queries and k nearest neighbors queries is to perform a sequential scan of the database. However, this method may be too slow for real-world applications. Usually, an index structure is used to filter out irrelevant objects during the search, without computing their distances to the query object. In this way, the search system can

avoid the sequential scan. The index is built in a preprocessing step, and the cost of building the index is amortized with the savings obtained on each performed similarity query.

Many of the indices proposed so far ensure that all relevant objects (according to the performed type of similarity query) in the database will be retrieved. This is specially true for metric [Zezula et al. 2005; Böhm et al. 2001] and multidimensional (spatial) indices [Samet 2006; Chávez et al. 2001], which discard only those objects that can be proved to be irrelevant. However, the main bottleneck of the efficiency issue in similarity search is the so-called curse of dimensionality [Chávez et al. 2001], which makes the task of searching some spaces intrinsically difficult, whatever algorithm is used. A recent trend to remove this bottleneck resorts to *approximate search* [Chávez and Navarro 2001; Zezula et al. 2005; Samet 2006], where it has been shown that one can find most of the relevant objects at a fraction of the cost of the *exact search* algorithm. These algorithms are welcome in most applications, because resorting to similarity searching already involves a fuzziness in the retrieval requirements: The process of modeling similarity between objects involves generally some loss of information. Thus, in most cases, finding some similar objects is as good as finding all of them.

2.3 Topological Properties

Although a particular similarity function could exhibit various properties, the topological properties of similarity functions deserve a special attention.

2.3.1 *Metric Postulates.* Among a number of topological properties, the metric postulates (axioms) are widely applied in similarity modeling. The metric postulates are defined as ($\forall x, y, z \in \mathbb{U}$):

$$\begin{array}{rcll} \delta(x,y) & = & 0 & \Leftrightarrow x = y \quad \text{reflexivity} \\ \delta(x,y) & > & 0 & \Leftrightarrow x \neq y \quad \text{non-negativity} \\ \delta(x,y) & = & \delta(y,x) & \text{symmetry} \\ \delta(x,y) + \delta(y,z) & \geq & \delta(x,z) & \text{triangle inequality} \end{array}$$

Reflexivity permits the zero dissimilarity just for identical objects. *Non-negativity* guarantees that every two distinct objects are somehow positively dissimilar. If δ satisfies reflexivity, non-negativity and *symmetry*, it is a *semimetric*. Finally, if a semimetric δ satisfies also the *triangle inequality* it is a *metric* (or metric distance). The triangle inequality is a kind of transitivity property; it says that if x, y and y, z are similar, also x, z are similar. If there is a finite upper bound d^+ such that $\delta : \mathbb{U} \times \mathbb{U} \mapsto [0, d^+]$, then δ is a *bounded metric*. In such case, $\mathcal{M} = (\mathbb{U}, \delta)$ is called a (bounded) *metric space*. To complete the categorization of functions based on combining the metric axioms, we also distinguish *pseudometrics* (not satisfying the reflexivity), and *quasimetrics* (not satisfying symmetry). For a formal definition of the above mentioned properties we refer to Khamsi and Kirk, and Corazza [Khamsi and Kirk 2001; Corazza 1999].

At this moment, we also define the category of *nonmetrics*, which we understand as any dissimilarity functions that do not fulfill one or more of the metric axioms.

2.3.2 *Other Properties.* There have been other topological properties utilized in similarity modeling. Some are more general (like four-point, pentagon, ultrametric,

negative-type inequality) [Marcu 2004; Khamsi and Kirk 2001], some are restricted to vector spaces (like segmental additivity, corner-point inequality) [Jäkela et al. 2008], some are set-theoretic (like matching, monotonicity, independence) [Tversky 1977], etc. Anyways, we consider the metric postulates restrictive enough yet sufficiently general to be shared by a large class of similarity functions. Hence, any other properties of a (dis)similarity function in this paper are discussed just in the context of domain-specific examples.

2.4 Motivation for Nonmetric Similarity

In the following, we discuss the motivation for nonmetric similarity measuring and the drawbacks of the metric case. First of all, we have to emphasize that we observe two database-specific objectives – the *efficiency* of similarity search (the performance issue) and the *effectiveness* of similarity search (the quality of query answers). Historically, the metric similarities did represent a reasonable trade-off concerning the efficiency/effectiveness problem [Zezula et al. 2005]. Metrics allowed some degree of effective similarity modeling, while their postulates could be used to index the database for efficient retrieval. However, as the complexity of retrieval applications has grown in the last years, the demands for nonmetric similarity measuring have become stronger. Hence, the universal effectiveness of metric functions could not be considered anymore as sufficient enough.

2.4.1 *Richness of Similarity Modeling.* The strongest rationale for nonmetric similarities is the increased freedom of similarity modeling – the author of a similarity function (the *domain expert*) is not constrained by metric postulates. Various psychological theories suggest the metric axioms could substantially limit the expressive power of similarity functions [Santini and Jain 1999; Tversky 1977].

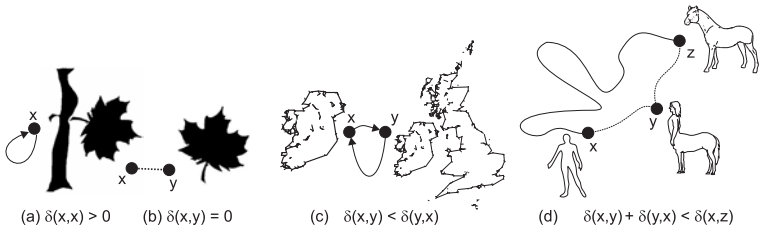

Fig. 2. Objections against metric axioms in similarity measuring: (a) reflexivity (b) non-negativity (c) symmetry (d) triangle inequality

In particular, reflexivity and non-negativity have been refuted by claiming that different objects could be differently self-similar [Krumhansl 1978; Tversky 1977]. For instance, in Figure 2a the image of a leaf on a trunk can be viewed as positively self-dissimilar if we consider a similarity which measures the less similar parts of the objects (here the trunk and the leaf). Or, alternatively, in Figure 2b the leaf-on-trunk and leaf could be treated as identical if we consider the most similar parts of the objects (the leaves). Nevertheless, the reflexivity and non-negativity are the less problematic properties.

Symmetry was questioned by showing that a prototypical object can be less similar to an indistinct one than vice versa [Rosch 1975; Rothkopf 1957]. In Figure 2c, the more prototypical "Great Britain and Ireland" image is more distant to the "Ireland alone" image than vice versa (i.e., a subset is included in its superset but not vice versa).

The triangle inequality is the most attacked property. Some theories point out that similarity has not to be transitive [Ashby and Perrin 1988; Tversky and Gati 1982], as shown by the well-known example: a man is similar to a centaur, the centaur is similar to a horse, but the man is completely dissimilar to the horse (see Figure 2d).

From a more applied point of view, the lack of metric postulates allows us to model similarity functions that exhibit the following desirable properties:

—*Robustness.* A robust function is resistant to outliers (noise or deformed objects), that would otherwise distort the similarity distribution within a given set of objects [Donahue et al. 1996; Howarth and Ruger 2005]. In general, having objects x and y and a robust function δ, then an extreme change in a small part of x's descriptor should not imply an extreme change of $\delta(x, y)$.

—*Locality.* A locally sensitive function is able to ignore some portions of the compared objects. As illustrated in the above example in Figure 2ab, we could model a "smart" similarity function that decides which portions of object x are relevant when evaluating its similarity with object y. This property leads not only to potential violation of non-negativity, but also to the violation of triangle inequality. For example, consider the centaur and the man (see Figure 2d); here we perform such a locally sensitive matching – we compare either the human-like or horse-like parts of the two images. The locality is usually used to privilege similarity before dissimilarity, hence, we rather search for similar parts in two objects than for dissimilar parts [Robinson et al. 2000; Smith and Waterman 1981]. As in real world, highly similar objects are not very common when compared at a global scale. An "augmentation" of similarity by locally sensitive functions provides a way how to distinctly separate similar and dissimilar objects.

2.4.2 *The Comfort of Similarity Modeling.* Another rationale for supporting nonmetric similarities, quite related to the first one, is to provide some comfort to the domain experts. The task of similarity search should serve just as a computer-based tool in various professions often not related to computer science. Hence, the authors of custom similarity functions that are to be employed into the search engines come from domains like computer vision, bioinformatics, medicine, material engineering, meteorology, music, psychology, chemistry, and many others. Naturally, when modeling their similarity functions, the domain experts should not be bothered by some "artificial" constraints laid on their functions, for example, by the metric postulates. As the domain experts often do not need a strong mathematical background, the enforcement of metric postulates to become an obligatory part of their "perfect" similarity functions represents an unpleasant obstacle. This is especially true when the similarity function is given by a complex heuristic algorithm [Wild and Willett 1996], or even as a device digitizing output of some physical phenomena [Tsukada and Watanabe 1995].

It should be noticed that, e.g., turning a semimetric into a metric could be

easily achieved by adding a sufficiently large constant. Or, turning a similarity into dissimilarity could be achieved as $\delta(\cdot,\cdot) = 1/s(\cdot,\cdot)$ or $\delta(\cdot,\cdot) = \text{constant} - s(\cdot,\cdot)$. However, such trivial transformations usually lead to inefficient similarity search, as it will be discussed in Section 4.5.1.

2.4.3 *Design & Implementation Issues.* Besides descriptions of similarity functions in closed forms (by various formulas), one can design a similarity that is described just by an algorithm written in a context-free language – as a black box returning a real-value output from a two-object input [Mandl 1998]. The topological properties (i.e., the metric postulates) of an algorithmically described similarity function are generally undecidable, so one has to treat such a function as a nonmetric.

Moreover, the similarity functions could be embedded in "black-box" hardware devices. In particular, specialized hardware ASIC (Application-specific integrated circuit) co-processors were designed to improve the performance of similarity measuring, following the VLSI (Very-large-scale integration) paradigm. The ASICs offer high performance for specialized tasks, e.g., a particular similarity measuring [Mukherjee 1989]. However, a disadvantage of such specialized devices is their limited usage when the similarity function has to be re-designed. A recent trend in VLSI is the reconfigurable computing, where a general-purpose FPGA (field-programmable gate array) is physically configured to act as a specialized processor (instead of a brand new ASIC design). Unlike CPUs where just the control flow is driven by software, the FPGAs allow to change the "native" data flow throughout the circuit – in simple words, to configure a task-specific hardware design. An unknown FPGA device implementing a similarity function [Freeman 2006; Freeman et al. 2005; Perera and Li 2008] has to be considered also as a nonmetric black box.

2.4.4 *Turning a Similarity into Dissimilarity.* Besides developing "natively" nonmetric dissimilarity functions, one could obtain a nonmetric function easily when transforming a similarity function into dissimilarity. To obtain a dissimilarity, one has to apply some decreasing monotonous function f on the original similarity function. However, application of such a function could lead to violation of reflexivity, non-negativity, and also the triangle inequality. For example, $f(s) = 1/s$ ensures non-negativity (and reflexivity) for positive similarity functions, but it will probably violate the triangle inequality. Even a simple subtraction of the similarity value from a constant number (e.g., from the maximal similarity score) could lead to dissimilarity that violates all the metric postulates except symmetry [Corazza 1999].

3. NONMETRIC SIMILARITY FUNCTIONS AND APPLICATION DOMAINS

In this section, we focus on solutions of domain problems that have employed the nonmetric measuring either for similarity search, or for other retrieval-specific tasks, such as clustering or classification. Note that in this section we are concerned just with identifying similarity-search problems from very different domains, in order to bring a real motivation for nonmetric functions and to justify the efforts spent on implementing nonmetric similarity search techniques.

The identification of a similarity search technique (or a similarity function alone)

in the domain problems is often not obvious (e.g., see protein matching in Section 3.4.2), because of specific terminology, methodologies and conventions within a particular domain. Hence, instead of simple surveying we also reinterpret the techniques in order to clearly separate the concept that represents (implements) a similarity search task. Moreover, we include only approaches employing inherently nonmetric (dis)similarity functions, that is, such functions that could not be trivially transformed into a metric distance suitable for efficient metric similarity search (as it will be discussed in Section 4.5.1).

3.1 General-purpose Nonmetric Similarities

In the following, we list a dozen of general nonmetric similarity functions used in various domains. Some domain-specific similarity functions will be additionally presented within the discussion on domain applications from Section 3.2 on.

3.1.1 *Fractional L_p distances.* A widely used family of distances for computing similarity in vector spaces (i.e., $\mathbb{U} = \mathbb{R}^d$ for some dimensionality d) is the *Minkowski distance* (L_p). Given two vectors $x, y \in \mathbb{R}$, the L_p distance is defined as

$$L_p(x, y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p}$$

For parameter $p \geq 1$, it is well known that the L_p distance holds the properties of a metric. Some examples of Minkowski metrics are the *Manhattan distance* $L_1(x, y) = \sum_{i=1}^{d} |x_i - y_i|$; the *Euclidean distance* $L_2(x, y) = \sqrt{\sum_{i=1}^{d} |x_i - y_i|^2}$; and the *Maximum distance* $L_\infty(x, y) = \max_{i=1}^{d} |x_i - y_i|$.

If $p \in [0, 1]$, such L_p distance function is known as *fractional distance* [Aggarwal et al. 2001]. The triangle inequality does not hold in fractional distances, thus it is only a semimetric. In Figure 3 see the ball shapes for various L_p distances (the ball border shows all the points at a fixed L_p distance from q).

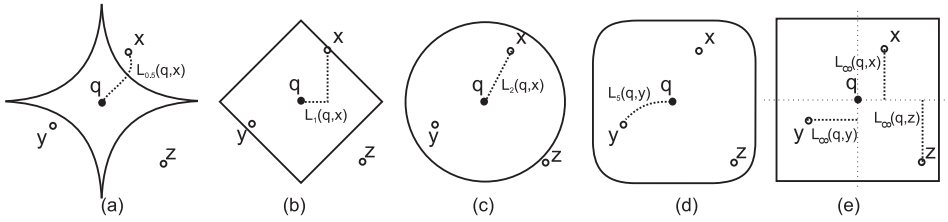

Fig. 3. Ball regions of L_p distances: (a) $L_{0.5}$ (b) L_1 (c) L_2 (d) L_5 (e) L_∞

3.1.2 *Dynamic Partial Function (DPF).* The Dynamic Partial Function (DPF) [Goh et al. 2002] is also related to the Minkowski distance. In DPF, only a few of the coordinate values are used to compute the distance between two objects. Let $c_i = |x_i - y_i|$, where x_i and y_i corresponds to the i-th coordinate of vectors x and y, respectively. Let Δ_m be the set of the m smallest values of $\{c_1, \ldots, c_d\}$ ($m \leq d$). The DPF is defined as

$$\delta_{DPF}(x,y) = \left(\sum_{c_i \in \Delta_m} |x_i - y_i|^p \right)^{1/p}, \; p \geq 1$$

Note that for different pairs of objects (x, y) it may be possible that different coordinates are the ones with minimum difference (i.e., the ones that belong to Δ_m), thus DPF does not satisfy the triangle inequality.

3.1.3 *Cosine measure & distance.* The cosine measure [Baeza-Yates and Ribeiro-Neto 1999] measures the cosine of the angle between two vectors. It is a suitable similarity function in situations where the magnitude of the vectors is not important – we are only concerned with the direction of the vectors. The cosine measure is defined as

$$s_{\cos}(x,y) = \frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum_{i=1}^{d} {x_i}^2 \cdot \sum_{i=1}^{d} {y_i}^2}}$$

Since $s_{\cos}(x, y)$ is a similarity function, by defining $\delta_{\cos}(x, y) = 1 - s_{\cos}(x, y)$ we get an equivalent semimetric distance, the *cosine distance*. When applying *arccos* on $s_{\cos}$ we get the *angle distance* $\delta_{\text{angle}}(x, y) = arccos(s_{\cos}(x, y))$, which is a metric distance.

3.1.4 *Kullback-Leibler divergence (KLD).* The Kullback-Leibler divergence (KLD) [Rubner et al. 2001] is used as a dissimilarity function between histograms based on the information theory. It is defined as

$$\delta_{KLD}(x,y) = \sum_{i=1}^{d} x_i \cdot \log\left(\frac{x_i}{y_i}\right)$$

The KL-divergence, according to Rubner et al., "measures how inefficient on average it would be to code one histogram using the other as the true distribution for coding" [Rubner et al. 2001]. Note that this distance function does not satisfy symmetry, nor non-negativity, nor the triangle inequality, and $\lim_{y_i \to 0^+} \delta_{KLD}(x, y) = \infty$ for any bin i of the histogram.

3.1.5 *Jeffrey-Divergence (JD).* The Jeffrey-divergence (JD) [Rubner et al. 2001] is also motivated by the information theory, and is defined as

$$\delta_{JD}(x,y) = \sum_{i=1}^{d} x_i \cdot \log\left(\frac{x_i}{\frac{x_i+y_i}{2}}\right) + y_i \cdot \log\left(\frac{y_i}{\frac{x_i+y_i}{2}}\right)$$

JD satisfies symmetry, but it does not satisfy the triangle inequality.

3.1.6 χ^2 *distance.* The χ^2-statistic distance [Rubner et al. 2001] measures if two empirical distributions were produced from the same underlying true distribution. Let $m(i) = \frac{x_i+y_i}{2}$ be the mean value of x_i and y_i. The χ^2 distance is defined as

$$\delta_{\chi^2}(x,y) = \sum_{i=1}^{d} \frac{x_i - m(i)}{m(i)}$$

This distance is not symmetric, it is not non-negative, and it does not satisfy the triangle inequality.

3.1.7 *Dynamic Time Warping Distance (DTW).* The Dynamic Time Warping [Berndt and Clifford 1994] is a dissimilarity function for comparing time series. Suppose that $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ are two time series, and let M be an $n \times m$ matrix with $M[i,j] = (x_i - y_j)^2$. A warping path $W = \{w_1, \ldots, w_t\}$, $\max\{m,n\} \leq t \leq m+n-1$, is a set of cells from M that are "contiguous", that is, (a) $w_1 = M[1,1]$, $w_t = M[m,n]$ (boundary condition), (b) if $w_k = M[a,b]$ and $w_{k-1} = M[a',b']$, then $a - a' \leq 1 \wedge b - b' \leq 1$ (continuity), and (c) if $w_k = M[a,b]$ and $w_{k-1} = M[a',b']$, then $a - a' \geq 0 \wedge b - b' \geq 0$ (monotonicity) [Keogh and Ratanamahatana 2005]. The DTW computes the warping path that minimizes the dissimilarity between the time series, thus

$$\delta_{DTW}(x,y) = \min_W \left\{ \sqrt{\sum_{k=1}^{t} w_k} \right\}$$

The DTW violates the triangle inequality.

3.1.8 *Longest Common Subsequence (LCS).* Formally [Cormen et al. 2001], a sequence x is a subsequence of a sequence y if there is a strictly increasing sequence of indices such that there is a match between symbols in x and y (the symbols in y must not be necessarily adjacent). Given two sequences x and y, a third subsequence z is a common subsequence of x and y if it is a subsequence of both x and y. The longest common subsequence (LCS) is the maximum length common subsequence of x and y. LCS is a similarity function s_{LCS} (with $s_{LCS}(x,y) = 0$ if x and y do not have a common subsequence). Even if modified to dissimilarity (e.g., $\delta_{LCS}(x,y) = s_{max} - s_{LCS}(x,y)$, with s_{max} the maximum possible value returned by s_{LCS}), it still does not satisfy triangle inequality.

3.1.9 *Earth Mover's Distance (EMD).* The Earth Mover's Distance (EMD) [Rubner et al. 1998] measures the least amount of work required to transform one distribution of values (a feature vector) into another one. To compute EMD, one needs to solve an instance of the transportation problem. Let c_{ij} be the cost of transforming one unit from x_i to one unit from y_j (we assume that the vectors are non-negative and that the weight of both vectors is normalized). The EMD computes the flows f_{ij} such that the transform cost is minimum subject to some constraints:

$$\begin{array}{rcl} \delta_{EMD}(x,y) & = & \min \left\{ \sum_{i=1}^{d} \sum_{j=1}^{d} c_{ij} f_{ij} \right\} \\ \text{subject to} & & \\ f_{ij} & \geq & 0 \\ \sum_{i=1}^{d} f_{ij} & = & y_j \;\; \forall j = 1, \ldots, d \\ \sum_{j=1}^{d} f_{ij} & = & x_i \;\; \forall i = 1, \ldots, d \end{array}$$

The EMD provides us with a tool to match not only corresponding dimensions, but any pair of dimensions. Depending on the cost values c_{ij} (also referred as "ground distance"), the EMD may be a metric (e.g., it satisfies the triangle inequality if $c_{ik} \leq c_{ij} + c_{ik}$ $\forall i,j,k$). But, if the ground distance is nonmetric, then

the EMD is also a nonmetric (only symmetry is guaranteed).

3.1.10 *Hausdorff Distance Variants.* The Hausdorff distance (HD) [Huttenlocher et al. 1993] is a metric distance for sets of objects. Given sets $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ and a function $h(A,B) = \max_{a \in A} \min_{b \in B} \delta(a,b)$ (the directed Hausdorff distance), with $\delta(a,b)$ an underlying distance between objects, the Hausdorff distance is defined as

$$\delta_{HD}(A,B) = \max\{h(A,B), h(B,A)\}$$

Some variations of the Hausdorff distance have been used for image retrieval. For example, the partial Hausdorff distance (PHD) [Huttenlocher et al. 1993] only considers subsets from A and B to compute the similarity between the sets. The PHD is defined as

$$\delta_{PHD}(A,B) = \max\{h_L(A,B), h_K(B,A)\},$$

where $1 \leq L \leq n$ and $1 \leq K \leq m$ are parameters that limits the computation of $h(B,A)$ ($h(A,B)$) to the K^{th} (L^{th}) ranked object (according to distance δ) in B (A). $h_L(A,B)$ and $h_K(B,A)$ are known as partial distance functions. The PHD violates the triangle inequality, thus it is nonmetric.

Another variant is called the modified Hausdorff distance (MHD) [Dubuisson and Jain 1994], which considers the average of $h_1(A,B)$ distances. It is defined as

$$\delta_{MHD}(A,B) = \max\{h_{MHD}(A,B), h_{MHD}(B,A)\},$$

with $h_{MHD}(A,B) = \frac{1}{|A|} \sum_{a \in A} \min_{b \in B} \delta(a,b)$.

The described nonmetric variants are only two examples of the many variants of Hausdorff-like distances, while most of them are nonmetric (either reflexivity or triangle inequality is violated). See their categorization in Dubuisson and Jain [Dubuisson and Jain 1994].

3.1.11 *Normalized Edit Distance (NED).* The edit distance (or Levenshtein distance) measures the minimum number of edit operations (insertions, deletions, and substitutions) needed to transform one string into another one [Levenshtein 1966]. Each edit operation can be weighted by a non-negative real value γ ($\gamma(a \to b)$ for substitutions, $\gamma(a \to \lambda)$ for deletions, and $\gamma(\lambda \to b)$ for insertions). An editing path P [Marzal and Vidal 1993] between strings x and y is a sequence of ordered pairs (i_k, j_k) ($0 \leq k \leq m$, with $m(P)$ the number of associated edit operations) such that

$$\begin{gathered} 0 \leq i_k \leq |x|,\ 0 \leq j_k \leq |y|;\ (i_0, j_0) = (0,0);\ (i_m, j_m) = (|x|, |y|), \\ 0 \leq i_k - i_{k-1} \leq 1;\ 0 \leq j_k - j_{k-1} \leq 1,\ \forall k \leq 1, \\ i_k - i_{k-1} + j_k + j_{k-1} \geq 1 \end{gathered}$$

The weights can be associated to the corresponding editing path,

$$W(P) = \sum_{k=1}^{m} \gamma(x_{i_{k-1}} + 1 \ldots i_k \to y_{j_{k-1}} + 1 \ldots j_k),$$

and it follows that $\delta_{edit}(x,y) = \min\{W(P)\}$.

The normalized edit distance (NED) [Marzal and Vidal 1993] between two strings x and y takes into account the length of the editing path. Let $\hat{W}(P) = W(P)/m(P)$. The normalized edit distance is defined as

$$\delta_{NED}(x, y) = \min\{\hat{W}(P)\}$$

It has been shown that the NED does not satisfy the triangle inequality [Marzal and Vidal 1993].

3.1.12 *Sequence Alignment Distance (SAD).* The SAD could be understood as a nonmetric generalization of the classic edit distance [Levenshtein 1966]. In sequence matching problems, one wants to compute the best alignment between a sequence $y = y_1 y_2 \dots y_d$ and a set of sequences $\mathbb{S}$. The best match corresponds to the object $x \in \mathbb{S}$ that minimize a certain distance function δ, which is defined using the notion of sequence alignment. The distance δ is defined such that y matches the sequence that needs the minimum number of edit operations (match, replace, delete and insert) to be converted into x [Parker et al. 2007]. Let $c(a, b)$ be the cost of matching character a with b, $c(a, -)$ the cost of inserting character a, and $c(-, b)$ the cost of deleting character b. The alignment cost for sequences x and y (starting from position i and position j respectively) is defined as

$$\delta_{SAD}(x, y, i, j) = \min \begin{cases} c(x_i, y_j) + \delta_{SAD}(x, y, i+1, j+1) \\ c(-, y_j) + \delta_{SAD}(x, y, i, j+1) \\ c(x_i, -) + \delta_{SAD}(x, y, i+1, j) \end{cases}$$

Depending on how the cost function c is defined, SAD may violate some or all metric axioms. Among the nonmetric ones, the Smith–Waterman (SW) algorithm can be used to efficiently find the best alignment between sequences [Smith and Waterman 1981]. Unlike other sequence alignment algorithms, the SW algorithm is a local-alignment function (i.e., allowing to ignore the non-matching parts of the sequences), while it furthermore supports scoring matrices and specific weights for inserting or enlarging gaps.

3.1.13 *Combined Functions.* Besides modeling the similarity functions by formulas/algorithms associated with the internal form of descriptors, the similarity functions could originate also as combinations of some underlying particular functions. The combinations include summing of particular functions, their multiplication, picking a median similarity, etc. Here we have to emphasize that even if the underlying functions are metrics, their combination could lead to nonmetric. For example, summing two metrics always leads to a metric, however, their multiplication or picking a median similarity results in a nonmetric distance. Naturally, if the underlying similarities are nonmetric, their combinations are nonmetrics as well.

Figure 4 shows the region balls for various combined similarity functions (one metric and three nonmetrics violating the triangle inequality). The $L_2 + \delta_{\text{angle}}$ is a summation of Euclidean distance and the angle distance (being thus a metric), the $L_2 + \delta_{\cos}$ is similar, however using the cosine distance instead of angle distance, being thus nonmetric. The $L_{0.3}$·QFD is a multiplication of nonmetric fractional L_p distance and the metric quadratic-form distance (resulting to nonmetric), while the 2-med(QFD, δ_{angle}, wL_2) distance picks the second smallest distance among the quadratic-form distance, the angle distance, and the weighted Euclidean distance (leading to nonmetric).

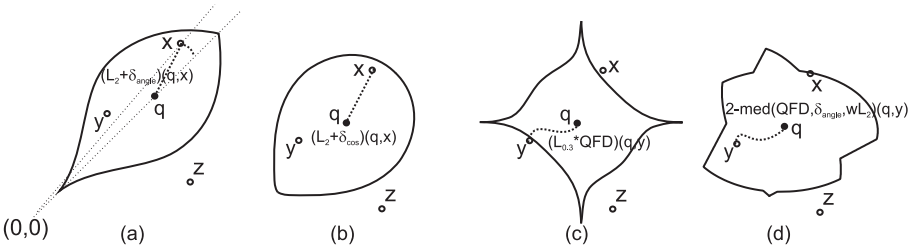

Fig. 4. Ball regions of combined distances: (a) $L_2 + \delta_{\text{angle}}$ (b) $L_2 + \delta_{\cos}$ (c) $L_{0.3}$·QFD (d) 2-med(QFD, δ_{angle}, wL_2)

In the particular case of vector spaces, as depicted in Figure 4, it might appear that balls belonging to metrics should be of convex shape, while balls belonging to nonmetrics should be nonconvex. However, this is not generally true as can be seen in Figure 4a (nonconvex ball for a metric) and Figure 4b (convex ball for a nonmetric). Another example could be the squared L_2 distance, which is a nonmetric but its ball shape is the same as for L_2 (Figure 3b). In summary, the notion of shape convexity/concavity is meaningful only in Euclidean vector spaces, so the shape of a particular non-Euclidean region says nothing about triangle inequality of the distance used.

3.1.14 *Computational Complexity of Nonmetric Functions.* The time complexity of algorithms implementing the similarity functions can vary from linear to exponential (regarding the size of the objects x and y to be compared). Fractional distances are computed in $O(d)$ time, and the DPF can be computed in $O(d + kd)$. KLD, JD, and χ^2 distances have $O(d)$ time complexity. DTW, LCS, and SAD are $O(nm)$ when computed by dynamic programming. In practice, the EMD is usually implemented by a linear programming method (e.g., simplex). The simplex algorithm has exponential worst-case complexity, but linear programming problems can be solved in (weakly) polynomial time. In particular, if both feature vectors have the same number of dimensions, the EMD can be computed in $O(n^3 \log n)$ time [Rubner and Tomasi 2001]. The PHD can be computed in $O(nm) \cdot O(d)$, where $O(d)$ is the time complexity of the partial distance function. The NED can be computed in $O(mn^2)$ time and $O(n^2)$ space, where $|x| = n$, $|y| = m$, and assuming $m \geq n$.

3.1.15 *Summary.* In Table II we give an overview of various properties concerning the general-purpose nonmetric functions mentioned in this section. In particular, a function could be similarity or dissimilarity (as defined in Section 2.1), it satisfies some of the metric postulates, it is either global or local (as discussed in Section 2.4.1). The time complexity ranges from cheap (i.e., $O(n)$) to expensive (e.g., $O(n^2)$, $O(2^n)$). Furthermore, we give a real domain example and also a reference to the section describing details of each function.

3.2 Nonmetric Similarity Functions in Multimedia Databases

The area of multimedia databases was one of the first suitable environments for similarity search. Because multimedia documents capture fragments of the nature

Table II. Properties of some general-purpose nonmetric functions (D = dissimilarity, S = similarity, R = reflexivity, N = non-negativity, S = symmetry, G = global, L = local)

	sim/dissim	metric postulates satisfied	global/local	time complexity	domain example	definition in section	domain usage in section
Frac. L_p	D	RNS	G	O(n)	image retrieval	3.1.1	3.2.1
DPF	D	RNS	L	O(n)	image retrieval	3.1.2	3.2.1
cosine sim.	S	S	G	O(n)	text retrieval	3.1.3	3.2.5
KLD	D	R	G	O(n)	audio retrieval	3.1.4	3.2.3
JD	D	RNS	G	O(n)	image retrieval	3.1.5	3.2.1
χ^2	D	R	G	O(n)	image retrieval	3.1.6	3.2.1
EMD	D	at least S	G	$O(n^3 \log n)$ or $O(2^n)$	detection of phishing web pages	3.1.9	3.2.5
DTW	D	RNS	G	$O(n^2)$	time series	3.1.7	3.3.2
NED	D	RNS	G	$O(n^2)$	hand-written digit recognition	3.1.11	3.5.1
LCS	S	NS	L	$O(n^2)$	music retrieval	3.1.8	3.2.4
SAD	D	some	G/L	$O(n^2)$	music retrieval	3.1.12	3.2.4
Hausdorff	D	at least NS	G/L	$\geq O(n^2)$	talker identification	3.1.10	3.5.2
combinations	D/S	some	G/L	some	chemical retrieval	3.1.13	3.4.3

that can be seen and heard, measuring similarity of multimedia documents provides a very natural tool for modeling human cognition of audio-visual stimuli.

3.2.1 *General Images & Video.* Content-based image similarity search has been a very active research area in the last decade. A recent survey on image retrieval cites almost 300 articles, and reports that publications in this area continue to grow at a fast rate [Datta et al. 2008]. Indeed, it could be argued that image retrieval is one of the most important research domains of similarity search.

The main problem in content-based image retrieval (CBIR) is to design algorithms to find similar images, based primarily on the content (color values of the pixels) of the image. Similarity is an inherently subjective concept, but in the case of image similarity this "subjectiveness" is more pronounced. There are many different ways to define when two images should be considered similar, for example, when they have similar color histograms, or when the textures of the objects depicted in the image are similar, or when the distribution of the objects in the image is similar, and so on. For this reason, the flexibility that nonmetric similarity functions give us is an important characteristic to be considered when implementing CBIR systems.

Nonmetric functions for image retrieval have been used for a long time. For example, Rubner et al. [Rubner et al. 2001] made an empirical comparison of several dissimilarity functions for color and texture attributes in images. Among the dissimilarity functions, they tested the χ^2 distance, Kullblack-Leibler divergence, Jeffrey-divergence, and the Earth Mover's distance (although the last one in its metric form). Their experimental evaluation showed that for classification and image retrieval tasks with large sample sizes the nonmetric functions performed best. Other nonmetric functions, like fuzzy feature contrast [Santini and Jain 1999], have

also been used for texture retrieval with illumination invariant features [Vacha and Haindl 2008].

The fractional L_p distances have been suggested for robust image matching [Donahue et al. 1996] and retrieval [Howarth and Ruger 2005]. Unlike the classic L_p metrics, the fractional L_p variants allow us to inhibit extreme differences in coordinate values (e.g., in a few bins of color histogram). This can be viewed as a robust behavior to outliers compared with traditional distance functions like the Euclidean distance. In another approach, the dynamic partial function (DPF) distance [Goh et al. 2002] was used for image classification, where 144-dimensional vectors of images were established (the features included color histograms, color means, color variances, color spreadness, color-blob, elongation, and texture features in three orientations). In an empirical study [Goh et al. 2002] a more effective classification was achieved using the DPF when compared with the metric Minkowski distances. Yet another nonmetric distance that has been used for image retrieval includes a nonlinear model based on a Gaussian function [Cha 2006]. Additionally, image classification with nonmetric distances (Partial Hausdorff Distance and fractional L_p distances) has also been studied [Jacobs et al. 2000].

Content-based video retrieval is another domain that is recently attracting much attention from the research community. Since 2003, the TREC Video Retrieval Evaluation [Smeaton et al. 2006] has been a common venue for researchers in this area. There have been a few works where nonmetric functions have been proposed for video retrieval. For example, Zhou et al. [Zhou et al. 2007] proposed to map a video segment into a set of symbols (representing a cluster that contains similar frames), and then to use the Probability-based Edit Distance for comparing symbol sequences. This similarity function takes into account both the temporal ordering in the video segment and the similarity between symbols.

3.2.2 *Geometries & Shapes.* The problem of searching similar shapes in 2D and 3D arises in a number of fields. Example domains include Computer Aided Design/Computer Aided Manufacturing (CAD/CAM), virtual reality (VR), medicine, molecular biology, and entertainment. The improvement in 3D scanner technology and the availability of 3D models widely distributed over the Internet are rapidly contributing to the emergence of large databases of multimedia data. Additionally, the rapid advances in graphics hardware are making the fast processing of these complex data possible and available to a wide range of potential users at a relatively low cost.

As 3D models are used in diverse application domains, different forms for object representation, manipulation, and presentation have been developed. In the CAD domain, objects are often built by merging patches of parameterized surfaces. Also, constructive solid geometry techniques are often employed, where complex objects are modeled by composing primitives. 3D acquisition devices usually produce voxelized object approximations (e.g., computer tomography scanners) or clouds of 3D points (e.g., in the sensing phase of structured light scanners). Probably the most widely used representation to approximate a 3D object is by a mesh of polygons (usually triangles). For 3D retrieval, basically all of these formats may serve as input to a similarity query.

There are several examples of nonmetric functions used for shape retrieval. For

example, the Partial Hausdorff Distance (PHD) has been shown to be useful for shape-based image retrieval [Huttenlocher et al. 1993]. The problem is to match a 2D segment into an image, taking into account translations of the segment (but no rotations). The PHD distance allows the system to effectively measure the degree of similarity between the 2D segment superimposed to the image. In particular, the PHD was shown to work well on images with some level of noise.

Additionally, a Fourier-based approach for shape similarity search called WARP [Bartolini et al. 2005] uses the dynamic time warping distance (DTW) to compare different shape descriptors. WARP computes features from 2D shapes by selecting some of the coefficients obtained after applying the Discrete Fourier Transform to the shape. Then, it uses the DTW for comparing the feature vectors. Its advantage over the more usually used Euclidean distance is that DTW considers elastic shiftings, thus it can match similar signals even if they are not completely aligned.

Nonmetric functions have also been used for 3D model retrieval. For example, Pu et al. [Pu et al. 2007] proposed to compute features from 2D projections of the 3D models. The feature vectors are obtained using the spherical harmonics transform applied to the 2D projections. Then, a similarity function based on the Euclidean distance is defined between the computed feature vectors. However, the match is performed between projections with minimum Euclidean distance, which means that for different pairs of objects different projections may be compared. Thus, this similarity function violates the triangle inequality.

3.2.3 *Audio – Speech & Music.* A technique for classification and search of audio records using nonmetric similarities was introduced by Foote [Foote 1997]. The original audio waveform is converted (by use of sliding window) into Mel Frequency Cepstral Coefficients (MFCCs), which are quantized and a histogram is produced. A histogram could be produced not only for a single audio record, but also for multiple records (e.g., a class of records). The histograms can be then compared by various dissimilarity functions. Besides the Euclidean distance, there were also nonmetric distances used, like the symmetric form of Kullback-Leibler divergence and the Correlation distance (which is similar to the cosine distance). The approach provides a general similarity search in databases of audio and music records, but allows us also to classify music genres (by comparing histogram of a query record with a histograms of classes associated with particular genres), and also distinguish music and speech from non-vocal sounds.

Another approach to nonmetric music retrieval was proposed by Pampalk [Pampalk 2006]. The descriptor of music record was divided into four subdescriptors – the average spectral shape (using MFCCs), fluctuation pattern (modulation of the loudness amplitudes), the gravity of the fluctuation pattern, and the bass of the fluctuation pattern. The nonmetric dissimilarity between two records was then evaluated as a weighted linear combination of four particular distances – 70% of the Kullback-Leibler divergence on the average spectral shape and three times 10% of the Euclidean distance on the rest of the descriptors.

Yet another approach [Logan and Salomon 2001] proposed a descriptor of music record that consisted of an ordered set of clusters, where the clusters were determined from MFCCs of the frames in the record. Each cluster in the set was

represented by just three values, the mean, covariance and weight of the cluster. The similarity of songs was then measured by the Earth mover's distance (EMD) on the descriptors. Because the Kullback-Leibler divergence was used as the ground distance on particular clusters, the resulting dissimilarity became nonmetric (for the definition of EMD see Section 3.1.9).

A different approach was presented for speech audio retrieval [Ratanamahatana and Tohlong 2006]. Instead of MFCCs, the descriptor of a speech record was formed by time series consisting of frequency distribution obtained by Fast Fourier Transformation of the waveform (additionally smoothed and normalized). Then, DTW was used as a dissimilarity function of two speeches, exhibiting a significant improvement in classification precision and speed with respect to MFCCs.

3.2.4 *Musical Scores.* Besides the raw recordings, a piece of music can be represented in a symbolic notation that does not depend on a real musical performance. The symbolic notation could be either in form suitable for reading by human (the musical score), or in a derived form suitable for computerized processing (e.g., MIDI files or processed audio signal).

In Typke et al. [Typke et al. 2003] the monophonic score (i.e., just the melody) was transformed into a set of weighted 2D points, where each note (musical symbol, respectively) in the score was modeled by a point. The position of the point in the space was determined by the timing of the note and its pitch, while the weight of a point was determined by a musical property of the note. In addition to the metric version of Earth Mover's Distance (EMD), the pseudometric modification of EMD – called the Proportional Transportation Distance [Giannopoulos and Veltkamp 2002] – was used for measuring similarity of two scores.

A method of similarity search in MIDI databases by use of audio-recording queries (or vice versa) was introduced in [Hu et al. 2003]. A MIDI file or audio recording is processed into so-called chromagram – a time series where each element of the series consists of 12-dimensional chroma vector. The similarity of chromagrams was then computed by DTW, where the Euclidean distance was used as ground distance on the 12-dimensional vectors. When compared with MFCCs and pitch histograms, the experiments showed that chromagrams with DTW performed the best.

Guo and Siegelmann proposed the time-warped longest common subsequence similarity (TWLCS) for monophonic symbolic music matching [Guo and Siegelmann 2004]. The TWLCS is an alignment technique inspired by the suitable properties of DTW and the longest common subsequence (LCS) nonmetrics. Like DTW, the TWLCS is robust with respect to variations in speed and rhythm. Like LCS, the TWLCS does not necessarily penalize missing, redundant, or noisy notes.

In another approach, the classic LCS is used for retrieval of polyphonic music given in symbolic representation (here MIDI files) [Suyoto et al. 2007]. However, only the query is expected to be given in natively symbolic representation, while the symbolic music scores in the database are produced by an automatic processing of the raw audio waveform. The automatic transformation process produces unnecessarily many notes – up to three times as many notes as actually present. Nevertheless, the LCS is capable to ignore the extra notes by an alignment of the (not noisy) query with the longest best-matching subsequence with the database score, thus making the retrieval more robust.

A study of music retrieval in folk songs was proposed [Parker et al. 2007], where a song/query was represented as a sequences of notes, which themselves were represented as tuples ⟨pitch, duration⟩. Moreover, instead of absolute values, the sequences contained "relative notes", i.e., changes in pitch and duration with respect to the previous note. Futhermore, the note tuples were binned to give a finite alphabet, while the sequence alignment distance (SAD) was used as a dissimilarity function on the resulting strings. The experiments have shown that such a model performs an effective music retrieval.

3.2.5 *Digital libraries & Web pages.* One of the first domains employing the nonmetric similarity search was the vector model of text retrieval [Baeza-Yates and Ribeiro-Neto 1999]. The text documents (e.g., plain texts, or web pages) are modeled as vectors in high-dimensional space, where each particular dimension in a vector belongs to a term (word) from the dictionary of terms appearing within the entire collection of texts. The value of a coordinate on i-th position in the vector then represents the contribution (weight) of i-th term in the dictionary to the document. There was a number of techniques proposed for creating the weights, the *tf-idf* scheme is probably the most popular.

The similarity of text documents is then evaluated by the cosine measure applied on the vectors. The reason for favoring cosine measure over the Euclidean distance is two-fold. First, the cosine measure ignores the quantity of text, that is, if we compare a document A with document B, or document A with B concatenated with its copy, we get the same similarity score. In other words, the angle between the vectors is important, not the actual Euclidean distance. Second, because the document vectors are very sparse (just tens out of hundreds of thousands dimensions are non-null), text collections under the cosine measure are efficiently indexable by the inverted index (for details, see Section 4.7.2). Note that in text retrieval domain (or web search) we rather talk about ranking on the documents, instead of retrieving a query result.

The latent semantic indexing (LSI) [Berry and Browne 1999; Berry et al. 1995] is an algebraic extension of the classic vector model. Its benefits rely on discovering so-called latent semantics hidden in the text collection. Informally said, LSI discovers significant groups of terms in the collection (called concepts) and represents the documents as linear combinations of the concepts. Moreover, the concepts are ordered according to their significance in the collection, which allows us to consider only the first k concepts as important (the remaining ones are interpreted as "noise" and discarded). As for the classic vector model, also LSI uses the cosine measure to rank the documents. To name the advantages, LSI helps to solve problems with synonymy and homonymy. Furthermore, LSI is often referred to be more successful in recall when compared to classic vector model [Berry et al. 1995], which was observed for pure (only one topic per document) and style-free collections [Papadimitriou et al. 1998].

A similarity-based approach for detecting phishing web pages was proposed by Fu et al. [Fu et al. 2006]. The method uses low-resolution images made of visual layouts of web pages, while a nonmetric version of the Earth Mover's Distance on the images is used for similarity search (phishing web page detection).

3.2.6 *XML databases.* XML documents can be used to represent a large variety of semi-structured data. Thus, huge XML collections have recently arisen (e.g., the DBLP Computer Science Bibliography, which to date lists more than 1.2 million publications). We expect that this trend will continue, as XML is a widely used approach for exchanging documents in the Web.

An important problem in large XML collections is to quickly find XML documents that have a similar structure or that have a subtree similar to a given pattern [Sanz et al. 2008]. Also, there have been attempts to define semantic similarity of XML attributes to combine it with the structural similarity [Tekli et al. 2007]. Buttler [Buttler 2004] presents a survey of similarity functions for structured documents. For example, the Tree Edit Distance compute the minimum number of edit operations to transform one tree into another. This distance is nonmetric, although it is not difficult to convert it into a metric [Buttler 2004]. Additionally, the Weighted Tag Similarity (WTS) computes how similar the tag sets of two documents are, weighted by the tag frequency.

Many of these similarity functions for comparing XML documents are transformed into distances that do not necessarily hold the metric axioms. For example, the Level-based similarity [Sanz et al. 2008] is forced to be positive by assigning the value 0 if it results to be negative. Thus, it does not hold reflexivity.

3.3 Nonmetric Similarity Functions in Scientific and Medical databases

Scientific and medical databases usually consist of various types of signals (1D, 2D, or even 3D) produced by measuring some physical phenomena. Because of the inherent noise in the signal, the employed similarity functions are often designed as robust to noise, and/or as locally sensitive, becoming thus nonmetric.

3.3.1 *Imaging data.* Nonmetric functions have been successfully applied in image databases for medical applications. For example, Antani et al. [Antani et al. 2004] proposes a complex nonmetric similarity function (based on the minimum integral that measures the similarity between two shapes) for comparing spine X-ray images. This approach obtained a better performance approach than the L_1 distance between Fourier descriptors. Also, several similarity functions in a content-based image retrieval system for tomography databases were tested in Tsang et al. [Tsang et al. 2005]. Their main result is that the Jeffrey-divergence was the most accurate similarity function compared with metrics like L_1 and L_2. Saha and Bandyopadhyay [Saha and Bandyopadhyay 2007] proposed a clustering technique based on nonmetric functions, as part of a fully automatic segmentation of magnetic resonance images (MRI) of the brain. Schaefer et al. [Schaefer et al. 2005] propose to use nonmetric multidimensional scaling to implement a visualization tool for medical images.

3.3.2 *Time series.* The similarity search in times series has a huge number of applications, including financial analysis, medical informatics, material engineering, sensor networks, and many others. Among the nonmetric similarity functions for time series, the dynamic time warping distance (DTW, see Section 3.1.7) and the Longest Common Subsequence similarity (LCS, see Section 3.1.8) are the most popular ones.

A newly proposed similarity for time series is the *General Hierarchical Model* [Zuo

and Jin 2007] (GHM), which computes correspondences between points belonging to the same hierarchy or role. It uses a ground distance D (e.g., DTW) to measure the distance between points in the same hierarchy. Thus, if D is nonmetric, GHM will be also a nonmetric distance. Additionally, the *Edit Distance on Real sequence* (EDR) [Chen et al. 2005] is a similarity function based on the Edit Distance that is robust to noise, local time shifting, length of trajectories, and outliers. The EDR violates the triangle inequality [Chen et al. 2005], thus it is nonmetric. Finally, the *Fast Time Series Evaluation* (FTSE) [Morse and Patel 2007], while not a similarity score for time series per se, is an algorithm that can speed up the exact computation of similarity functions that rely on dynamic programming for their computation (like the LCS and the EDR). Some application domains for time series are:

—*ECG.* The DTW proved its effectiveness in matching electrocardiograms (ECG) [Tuzcu and Nas 2005]. Unlike wavelet analysis and Euclidean distance, DTW is able to differentiate ventricular tachycardia from supraventricular tachycardia in ECG signals, while differentiation of these two rhythm types has significant clinical implications. The authors note that DTW can potentially be used for automatic pattern recognition of ECG changes typical for various rhythm disturbances. A DTW-based approach to ECG classification was also proposed by Huang and Kinsner [Huang and Kinsner 2002].

—*Seismological signals.* Another playground for DTW dissimilarity offers the area of seismologic signal search and classification. Angeles-Yreta et al. propose a combination of Piecewise Aggregate Approximation (PAA), as a dimension reduction technique, together with DTW for matching seismologic signals [Angeles-Yreta et al. 2004].

—*Multivariate time series.* Multivariate (multidimensional) time series (MTS) are a common form of data representation in multimedia, medical, and financial applications. A nonmetric similarity function (called Eros) for MTS was proposed by Yang and Shahabi [Yang and Shahabi 2004], showing its superiority to Euclidean distance, DTW, and others. Since MTS could be understood as a matrix, there can be an eigenvector matrix obtained using principal component analysis for each MTS. The Eros similarity (being an extension of Frobenius norm) then uses the eigenvector matrices to compute weighted sum of cosines of angles between corresponding eigenvectors of the two compared MTSs.

—*Trajectories of moving objects.* The trajectory of a moving object could be regarded as a multidimensional time series, where each element of the series describes a position of the object in space and time. We already mentioned the EDR [Chen et al. 2005] as a nonmetric function for trajectories of objects. Another method for robust matching of trajectories employing an extended version of LCS was proposed in [Vlachos et al. 2005]. Because LCS in its basic form is suitable only for symbolic time series (i.e., allowing just a match or mismatch of two symbols), the authors have generalized LCS to work also with trajectories. In particular, points on trajectories match if they are sufficiently close (achieved by a threshold parameter ϵ). The authors have also proposed a more advanced version of LCS, the so-called LCS sigmoidal similarity, for which the parameter ϵ has not to be specified. The results have shown that LCS-based similarities outperformed Euclidean distance (being sensitive to noisy points in the trajectory)

but also DTW (which provides a global alignment, so it cannot effectively match fragments of the trajectories).

3.3.3 *Graph databases.* A related problem to XML similarity retrieval is searching and mining in graph databases, which has also become a relevant problem with applications, for example, in bioinformatics, RDF databases and social networks. Here, the main task is to search in complex structured data [Yan et al. 2005] or to compute correlations between graphs [Ke et al. 2008]. Several similarity functions can be defined for these kind of data. For example, the graph distance [He and Singh 2006] is defined as the minimum edit distance (cost of transforming one graph into another) under all possible graph mappings. This distance relies on vertex and edge distances, thus if they are nonmetric the graph distance is also nonmetric.

3.4 Nonmetric Similarity Functions in Biological and Chemical databases

The similarity search in databases of (bio)chemical molecules/compounds has an unforeseeable number of applications, including chemical activity prediction, biological function assessment, rapid drug trials, and many others. Simultaneously, the complexity of chemical descriptors opens space for various nonmetric approaches to similarity measuring. In the following, we point to several examples from the large number of approaches to (bio)chemical similarity.

3.4.1 *Proteins – primary structure.* In the 1990's there appeared tools for automated protein sequencing, so that the field of bioinformatic research got an opportunity to expand substantially. As any protein is a chain of amino acid bases (there are 20 types of amino acids), its sequential representation is straightforward – each symbol in the sequence stands for an amino acid within the protein chain. The sequential representation of a protein is called the *primary structure* of a protein. The databases of protein sequences serve for a variety of bioinformatic tasks (e.g., a classification of unknown biological functions), while the similarity between proteins plays a key role in most of the tasks.

Apart from general-purpose similarity functions for strings (e.g., edit distance), there appeared specific similarity functions (algorithms for sequence alignment, respectively) that reflect not the pure sequential similarity, but should rather score the biological similarity (i.e., similar biological functions). The biology-specific extensions include various scoring matrices (e.g., PAM or BLOSUM [Dayhoff et al. 1978; Henikoff and Henikoff 1992]) which allow one to determine the similarity of a particular pair of symbols (a probability of mutation of one amino acid into another). Another extension is a specific treatment of gaps in the alignment, while the penalty (cost) for opening a new gap is greater than the penalty for a continuation of the gap. A particular effect of such extensions is a violation of the triangle inequality, so the protein alignment similarities are mostly nonmetric.

In particular, the *Needleman-Wunch* algorithm [Needleman and Wunsch 1970] provides a global alignment of two proteins, hence, it is useful for measuring the similarity of entire proteins. The *Smith-Waterman algorithm* [Smith and Waterman 1981] provides a local alignment, hence, it is useful for matching only the most similar parts of two proteins. A very popular algorithm for local alignment is also the *BLAST* (Basic local alignment search tool) [Altschul et al. 1990], which represents not only a heuristic local-alignment similarity function, but it is designed

as a standalone search technique (i.e., the semantics of the function is adjusted to the search algorithm). The *FASTA* algorithm [Lipman and Pearson 1985] has a similar purpose as BLAST, though it was developed earlier and is less used than BLAST.

The local alignment techniques for protein matching are more popular than the global ones, because the biological functions are encoded as local fragments within the protein chain.

3.4.2 *Proteins – tertiary structure.* Due to the advances in methods suitable for analyzing proteins (e.g., X-ray crystallography, NMR spectroscopy), in the last two decades there appeared more sophisticated three-dimensional representations of proteins, the so-called *tertiary structures* of proteins[2]. Although there is no single widely accepted type of tertiary structure, the basic tertiary structure of a protein could be viewed as an open 3D polygon, where each vertex is labeled by a symbol for an amino acid. That is, a tertiary structure could be used to derive the primary structure but not vice versa, because the spatial information (especially the angles between edges) plays an important role.

Since the tertiary structure provides a more precise representation of proteins, it could be also used for more precise functional similarity measuring and search. The scheme for evaluating a similarity between two proteins given by their tertiary representations consists of two steps:

—*Alignment.* The vertices of two 3D polygons have to be aligned in a similar way as the sequences (primary structures) are being aligned. However, the alignment for tertiary structures is not based just on the symbols (sequence of amino acids), but there come also spatial factors into play, like how deeply the atoms are buried, hydrophobicity of submolecules, etc.

—*Optimal rotation & similarity evaluation.* The aligned polygons (their subparts, respectively, because the alignment could also be local) are rotated in order to find their optimal spatial superposition. For this reason the Kabsch algorithm is used [Kabsch 1976], which tries to find a rotation that minimizes the root mean square distance (RMSD) between the two polygons. Since RMSD is based on Euclidean distance (being sensitive to outliers), there were also alternative classifiers of the rotation proposed, like the elastic similarity score [Holm and Sander 1993]. The resulting spatial matching score of the two polygons determines the similarity of the two proteins.

The two-step implementation of similarity on tertiary structures, as discussed above, appeared in a number of methods, like *SAP* [Taylor and Orengo 1989], *ProSup* [Lackner et al. 2000], *STRUCTAL* [Gerstein and Levitt 1998], and *MAMMOTH* [Ortiz et al. 2002]. Because of the two steps and because of many particular stages of the measuring, the similarities are nonmetric.

3.4.3 *General Molecules & Compounds.* The similarity search in databases of general chemical compounds and molecules represents a huge research subarea of

[2]For completeness, the *secondary structures* of proteins provide just an intermediate step towards the tertiary structure – they describe particular fragments of the tertiary structure, rather than an entire protein.

chemoinformatics [Nikolova and Jaworska 2003; Willett et al. 1998]. The similarity-related problems are essential to tasks like compound property prediction, virtual screening, diversity analysis, pharmacophore searching, ligand docking, etc. Many approaches for feature extraction from molecules and compounds have been proposed together with many (dis)similarity functions complying with these feature descriptors.

The most widely used descriptors for chemicals are strings of binary features (fingerprints) and various real-valued vectors (e.g., CATS descriptors). Within the huge area of chemical similarity search we can find many nonmetric dissimilarity functions violating either triangle inequality or symmetry. For example, the cosine distance and dice distance are simple nonmetric distances for measuring global similarity of chemical descriptors. We could also find combined nonmetric distances, such as the multiplication of Soergel metric and the squared Euclidean distance [Dixon and Koehler 1999], which is a nonmetric (not only due to the squared Euclidean, but also because of the multiplication, see Section 3.1.13). Although the Soergel metric alone showed a bias towards molecule size and the squared Euclidean distance alone placed greater emphasis on structural diversity of large compounds, their multiplication showed to provide a reasonable trade-off.

Other types of descriptors include 3D models of molecules, where the local (nonmetric) alignment of molecules is essential for ligand docking [Robinson et al. 2000]. In addition to the violated triangle inequality (which is a typical property of partial/local matching), the local alignment could be also asymmetric [Willett 1998; Willett et al. 1998]. This allows one to model directional compound similarity where a (smaller) molecule is present as a fragment in another molecule, but not vice versa.

A technique for matching two molecules represented by 3D structures (their molecular electrostatic potentials, respectively) was introduced in [Wild and Willett 1996]. The quality of a match was measured by the Carbo similarity (a variant of cosine measure), while the best match of the molecules was searched by use of a genetic algorithm (GA). The GA seeks to identify a combination of translations and rotations that will align one molecule with the other, where the Carbo similarity plays the role of a fitness function (i.e., the algorithm tries to maximize the similarity). The resulting match of molecules then provides their final similarity score – the Carbo index on (sub)optimally aligned molecules. As GAs could lead to unpredictable suboptimal solutions (like any other soft-computing technique), we have to generally assume the GA-based molecular similarity as a nonmetric one.

3.5 Nonmetric Similarity Functions in Biometric databases

The similarity search in biometric databases usually serves to identification or authentication of a person, by means of anatomic (or physiologic) biometric features (like human fingerprint, voice, iris, face, signature, etc.). Although the Hamming or Euclidean distance represent a sufficient solution for identification by fingerprints or irises, the signature, voice or face identification requires more complex (nonmetric) similarity functions.

3.5.1 *Handwritten recognition.* The recognition of handwritten signatures or handwritten text has many practical applications, such as the scanning of heritage

documentation, the implementation of authentication systems, and the automatic processing of forms. In this case, the recognition problem is not as easy as with printed documents, where optical character recognition (OCR) techniques have been well studied and have a high rate of effectiveness. Usually, the recognition is performed at the character or word level by comparing the actual text with samples. The main tasks in handwritten recognition are identification (i.e., to recognize what was written) and verification (i.e., to validate that what was written corresponds to an a priori known text).

In [Marzal and Vidal 1993] the authors use normalized edit distance (NED) for classification of hand-written digits, where each image of digit is transformed into a string. The alphabet here represents different discrete contours in the digit. The experimental results show that, given the string representation of digits, the NED classifies better than not normalized or post-normalized classic edit distances. In another work, Zhang and Srihari [Zhang and Srihari 2002] propose a nonmetric function for binary feature vectors and a k-NN algorithm for handwritten digit recognition. The similarity function is based on L_1 but it violates reflexivity in general. In another related work, Wirotius et al. [Wirotius et al. 2004] present an approach based on DTW for online handwritten signature verification. Also, Cha et al. [Cha et al. 2005] compare different similarity functions for handwritten character recognition, and propose the so-called azzo similarity, which is based on the inner product of binary feature vectors.

3.5.2 *Talker identification.* An approach similar to the music and speech similarity retrieval (see Section 3.2.3) has been applied to talker identification [Foote and Silverman 1994]. A talker's utterance waveform is converted into Mel Frequency Cepstral Coefficients (MFCCs), which are quantized and a histogram is produced. Then, given a database of talkers' utterances, for a query utterance the nearest-neighbor utterance from the database is retrieved by use of applying the symmetric form of Kullback-Leibler divergence (applied on the query histograms and each of the database objects). A talker is positively identified if the similarity threshold to its nearest neighbor is high enough.

Another approach to talker identification makes use of the so-called nearest neighbor distance (NND) [Higgins et al. 1993]. A speech descriptor is represented by a set of feature vectors, determined from the MFCCs of the speech waveform (as discussed in Section 3.2.3). Actually, NND is similar to the modified Hausdorff distance (see Section 3.1.10), where the squared Euclidean distance is used as the ground distance on feature vectors.

3.5.3 *2D face identification.* A method for identification of a particular face in a raster image using the modified Hausdorff distance (as defined in Section 3.1.10) was proposed in [Jesorsky et al. 2001]. Having a database of face models (where each model is represented by a set of 2D points describing the face), one would like to determine whether a face in the query image matches a model face in the database. The basic idea is to transform the raster image into a set of points and compare sets of model faces in the database with the set obtained from the query image. In the simplest form, a pair query image/database model is compared by use of the modified Hausdorff distance. However, since the query image is generally

not normalized, the modified Hausdorff distance is enhanced by an additional transformation step, defined as $min_{p \in \mathcal{P}}$ mHD$(A, T_p(B))$, where p is a parameter into a transformation T_p, mHD is the modified Hausdorff distance, A is the database face and B is the query face. Such a transformation T_p is used, which minimizes the modified Hausdorff distance between A and B. In other words, we obtain a transformation-invariant matching of faces by use of modified Hausdorff distance, where the transformational extension makes the already nonmetric distance even more nonmetric.

3.5.4 *3D face identification.* A modern approach to human identification by faces is the matching of 3D models (surfaces) which are obtained either natively (by use of 3D scanner) or as a 3D reconstruction from raster images (multiple pictures of a person are taken from different angles). In particular, various deformable models were proposed, where the 3D templates were matched (and deformed) to fit each other as good as possible. For instance, in [Lu and Jain 2008] the authors propose a matching model as an optimization problem consisting of four steps. First, a coarse alignment of the two 3D models is performed by use of 3 anchor points. Second, an iterative closest point algorithm is applied to obtain the optimal rotation and translation of the models. Third, the BFGS quasi-Newton method is used to compute deformation parameters while minimizing the overall cost function (regarded as the dissimilarity of the models). Fourth, steps 2-4 are repeated until a convergence is reached, that is, a minimal value of the cost function is determined. Since the matching algorithm performs highly nonlinear transformations, the resulting dissimilarity function is nonmetric (the triangle inequality and symmetry are violated).

4. EFFICIENT SEARCH IN NONMETRIC SPACES

The *efficient* (fast) similarity search is crucial for large-scale and/or query-intensive applications. Even an extremely *effective* retrieval system (exhibiting high precision/recall values) is useless if it is not efficient, that is, requiring full sequential scan of the database. Hence, the problems of efficiency and effectiveness must be necessarily solved together. In the previous section, we have enumerated a number of approaches to effective domain-specific nonmetric similarity search applications. On the other hand, in this section we focus on principles and access methods that offer efficient implementation of various nonmetric searches. While trivial sequential processing of a single similarity query requires $n = |\mathbb{S}|$ (dis)similarity computations (so-called computation costs), an efficient search should reduce the computation costs to be sublinear on the database size.

4.1 The Framework for Efficient Nonmetric Search

Before we start a discussion on efficient similarity search in more detail, in this section we propose a framework (or a "guide map") that could help to orient the reader in the problem. However, we emphasize we are not going to propose a strongly formalized framework because of the vague nature of the problem. Since the term "nonmetric" just says something is *not* metric, we have completely no property common to all similarity functions. Hence, there is no fixed property that could be used as a basis for a mathematical framework in a similar way the metric

postulates are employed by a framework related to metric similarity search.

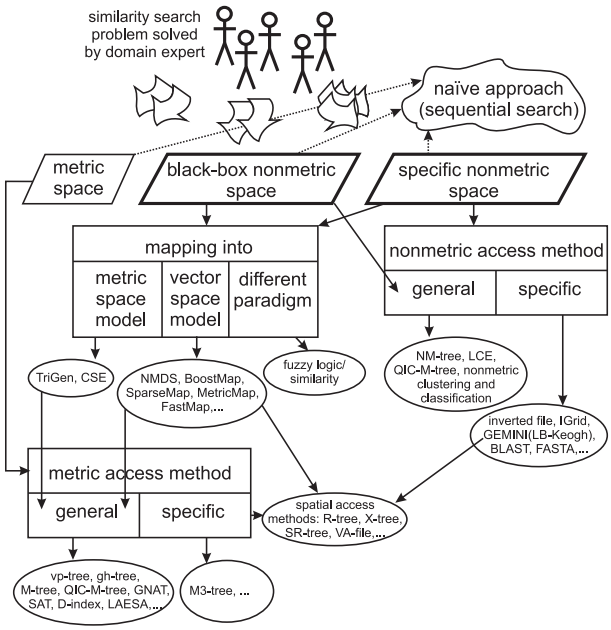

Fig. 5. The framework of nonmetric similarity search

Nevertheless, we propose a rather practical schema (see Figure 5) of the state-of-the-art techniques, that can be combined into processes, that provide more or less efficient nonmetric similarity search. At the top of the figure we assume a domain expert with a clearly defined similarity search problem. The expert knows the model, i.e., the techniques that transform some complex raw data into the descriptors of the universe $\mathbb{U}$ and the domain-specific similarity function on $\mathbb{U}$. We also assume the domain expert requires efficient solution of similarity search for her/his application, otherwise the naïve solution (the sequential search) is sufficient. In other words, the database is large enough and/or the similarity function is computationally expensive enough to refuse the naïve solution.

The proposed framework/map shows that a domain expert has several options depending on certain characteristic of the particular problem:

—If the problem can be modeled in a metric space, there is plenty of efficient solutions. The best one for a specific application will depend on the exact data type, data distribution, intrinsic dimensionality, etc.

—If the problem uses a specific nonmetric distance function, the expert may have luck and there is an efficient specific index available (e.g., inverted file, FASTA, BLAST, see Section 4.7). A specific index is usually more efficient than a general one, so this is the best case for the overall performance (i.e., best efficiency and effectiveness).

—If the problem is modeled as black-box similarity or there is no specific index for the used nonmetric distance then:
  —One can use mapping of the problem into another space/paradigm, but this may imply loosing discriminative power or effectiveness of the (dis)similarity function. Moreover, a mapping usually requires some preprocessing of the database (slowing down the indexing process), while the database is often required as static and known beforehand (preventing from searching in dynamic databases). An advantage is usually a faster query processing in the simpler target space (e.g., metric space).
  —One can use some of the few general nonmetric index structures/algorithms. The advantage is an all-in-one solution providing no or little parameterization – the method alone analyzes the (sample of) database and indexes it automatically. The drawback is usually slower query processing and also only approximate search (thus decreasing the retrieval effectiveness).

As we motivated in the previous section, there are many application domains where complex data types and complex (dis)similarity functions are used to compare objects. However, as will be discussed further in this section, only a few distance-specific and general nonmetric index structures were proposed so far. We believe this is a huge motivation to continue developing algorithms and data structures for performing efficient similarity search in complex domains, as applications from very different domains would benefit from advances in this area.

4.2 The Metric Case

We start the discussion with an introduction to efficient metric similarity search. Although the metric similarity search is not in the scope of this paper, we could reuse the metric case when turning a nonmetric search problem into a metric one (as discussed in Section 4.5).

The *metric access methods* (MAMs) [Chávez et al. 2001; Zezula et al. 2005; Samet 2006] provide data structures and algorithms by use of which the objects relevant to a similarity query can be retrieved efficiently (i.e., quickly). MAMs build a persistent auxiliary data structure, called *metric index*, so we also talk about metric indexing. The main principle behind all MAMs is the utilization of the triangle inequality property (satisfied by every metric), due to which MAMs can organize/index the objects of $\mathbb{S}$ within distinct classes. In fact, all MAMs employ the triangle inequality to cheaply construct lower and/or upper bounds of a distance $\delta(q,x)$ by use of an object p (called pivot), where the distances $\delta(p,x), \delta(p,q)$ are known but $\delta(q,x)$ is not (see Figure 6a). The construction of lower bound and upper bound distance by use of pivot p is easily computed as

$$|\delta(q,p) - \delta(p,x)| \leq \delta(q,x) \leq \delta(q,p) + \delta(p,x)$$

As the non-hierarchical MAMs use directly these bounds for search (e.g., AESA, LAESA), hierarchical MAMs, which is the vast majority (e.g., M-tree, GNAT, SAT, vp-tree, D-index, etc.), use the bounds to form a hierarchy of search regions that cover the underlying data. When a query is processed, many non-relevant database objects or entire search regions are filtered out just by use of the lower/upper bounds, so the searching becomes more efficient.

In the rest of this subsection we survey several most representative MAMs based on different techniques – the direct usage of lower/upper bounds (pivot tables), hierarchy of ball regions (M-tree), hierarchy of hyperplane partitioned regions (GNAT), and metric hashing (D-index).

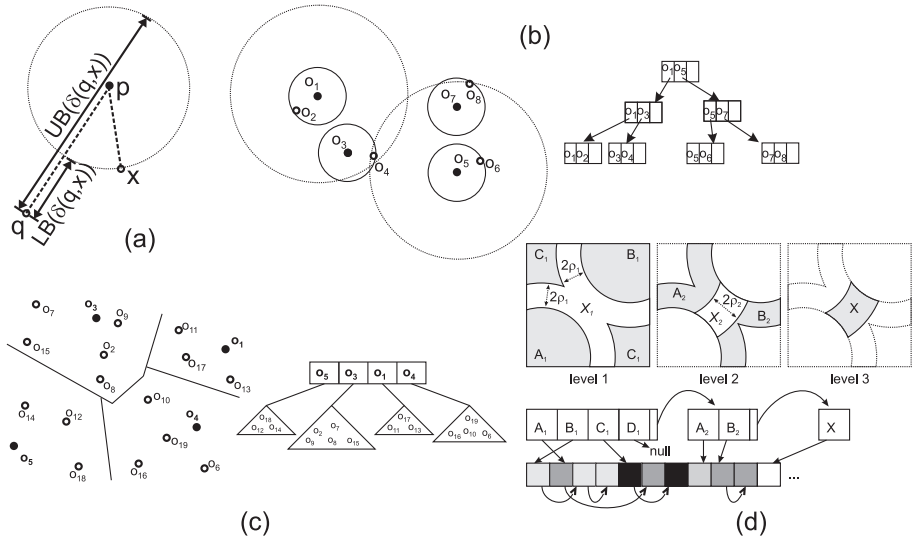

Fig. 6. (a) Creating lower- and upper-bound distances using triangle inequality (b) M-tree (c) GNAT (d) D-index

4.2.1 *Pivot Tables.* The most straightforward utilization of the lower-bound distances are various methods called *pivot tables* or *distance matrix methods*. In principle, there are some pivots selected [Bustos et al. 2003] from the database and for each object x in the database a vector consisting of all the distances from x to the pivots are computed. The distance vectors of all the database objects then form a distance matrix (or pivot table).

When querying by a range query[3] (q, r), a distance vector for the query object q is computed the same way as for a database object. Then, a database object x is cheaply discarded from the search if its lower-bound distance to q is greater than r. The lower bound distance is computed as the maximum value lower bound (given by the formula above) over all the pivots p_i we use. Conversely, if the minimal upper bound distance is lower than r, the object x is cheaply confirmed as a part of the result. If the object is neither filtered nor confirmed it must be checked by (expensively) computing the actual distance $\delta(q, x)$. Of course, the more pivots we use, the more tight the lower/upper bounds are and thus the more efficient the query processing is. On the other hand, if we use too many pivots, the computation of the distance vector for the query object q will become expensive, so the overall performance will deteriorate. There have been proposed many MAMs based on

[3]For kNN query the procedure is similar, but some heuristics for determining the dynamic query radius r must be additionally taken into account.

pivot tables, namely, the AESA [Vidal 1986], LAESA [Micó et al. 1994], PM-tree (based on M-tree and pivot tables) [Skopal 2004], etc.

4.2.2 *M-tree.* The *M-tree* [Ciaccia et al. 1997] is a dynamic metric access method that provides good performance in database environments. The M-tree index is a hierarchical structure, where some of the data objects are selected as local pivots of ball-shaped regions, and the remaining objects are partitioned among the regions in order to build up a balanced and compact hierarchy (see Figure 6b). Each region (subtree) is indexed recursively in a B-tree-like (bottom-up) way of construction. The queries are implemented by traversing the tree, starting from the root. Those nodes are accessed, the parent regions of which are overlapped by the query region, e.g., by a range query ball (q, r). Moreover, each node contains the distances from the pivots/objects stored in the node to the pivot of its parent region. Hence, some of the M-tree branches can be filtered without the need of a distance computation, thus avoiding the "more expensive" direct overlap check.

4.2.3 *GNAT.* The Geometric Near-Neighbor Access Tree (GNAT) [Brin 1995] is a metric access method that extends the Generalized-Hyperplane Tree [Uhlmann 1991]. The main idea behind GNAT is to partition the space into zones that contain close objects. The root node of the tree contains m objects selected from the database, the so-called *split-points*. The rest of the objects is assigned to their closest split-point. The construction is based on a greedy algorithm that selects the split-points, such that they are far away from each other. Each zone defined by the selected split-points is partitioned recursively in the same way (possibly using a different value for m), thus forming a search hierarchy (see Figure 6c). At each node of the tree, a $O(m^2)$ table stores the range (minimum and maximum distance) from each split-point to each zone defined by the other split-points.

4.2.4 *D-index.* An approach based on metric hashing was proposed by Dohnal et al. [Dohnal et al. 2003], called the *D-index*. The database is partitioned by *ball partitioning ρ-split functions* $bps^{1,\rho,j}$, defined as:

$$bps^{1,\rho,j}(o_i) = \begin{cases} 0 \text{ if } \delta(o_i, p_j) \leq d_m - \rho \\ 1 \text{ if } \delta(o_i, p_j) > d_m + \rho \\ 2 \text{ otherwise} \end{cases}$$

where p_j is a fixed pivot object assigned to the function $bps^{1,\rho,j}$, ρ is a splitting parameter, and d_m is a median distance. When combined k such functions (and k pivots), we obtain a complex hashing function $bps^{k,\rho}$ partitioning the database among 2^k partitions and one exclusion set. For each indexed object a hash key of its target partition is computed as a combination of binary values $0, 1$ returned by particular $bps^{1,\rho,j}$ functions. In case that (at least) one 2 is returned by a $bps^{1,\rho,j}$ function, the object is assigned to the exclusion set. In simple words, the exclusion set stands for a "border territory" separating the partitions, so that objects in different partitions can be easily distinguished during search.

The D-index itself then consists of an h-level hashing table, such that 2^{k_i} buckets are maintained at the i-th level (k_i is the number of pivots used at i-th level), where every bucket corresponds to one partition, and is accessible by the i-th-level hashing function $bps_i^{k_i,\rho}$. For the objects hashed into the i-th-level exclusion set, the i+1-th

level of the table is created and the remaining objects are repartitioned by function $bps_{i+1}^{k_{i+1},\rho}$. The last level consists of a single bucket belonging to the exclusion set of the entire D-index (see Figure 6d). For different D-index levels, the hashing functions $bps_i^{k_i,\rho}$ can vary in the number of ρ-split functions and, consequently, in the number of pivots.

4.3 Measuring Indexability

The implicit topological properties of a dissimilarity function itself (e.g., the metric postulates) is just a partial information needed for designing an efficient access method. The dissimilarity function δ together with the descriptor universe $\mathbb{U}$ provide us with some topological characteristics of the entire "full" space. However, according to some other properties, the particular databases embedded into the universe space could vary substantially. In particular, the volume of every database is by far smaller with respect to the universe volume, and also the data distribution in the universe could significantly vary from database to database.

In general, the distribution-specific properties of a database together with the topological properties of the dissimilarity function could be used to establish various concepts of data "separability" or "indexability". Such a quantitative characteristic should say to what extent could a database be partitioned in a way suitable for efficient search, that is, for search in as few candidate partitions as possible. In the following, we discuss two approaches to indexability measures, the intrinsic dimensionality and the ball-overlap factor, even though their application is fully relevant just for the metric case. Both of the indexability measures use some statistics obtained from pairwise distances between database objects.

Since we are not aware of a general nonmetric indexability concept (which would have to require some other constraint, anyways), an application of the metric-case indexability concepts can still be useful. First, a nonmetric search problem could be mapped to metric search problem (as discussed in Section 4.5), so here the metric indexability concepts are fully correct. Second, even when using directly a nonmetric distance, the metric postulates could be violated only slightly, so that even in this case the metric indexability concepts might provide some information.

4.3.1 *Intrinsic Dimensionality.* The distance distribution can reveal a structure inside the database, that is, whether there are clusters of objects and how tight they might be. Given a database $\mathbb{S}$ and a metric distance δ, the efficiency limits of any metric access method are indicated by the *intrinsic dimensionality*[4], defined as

$$\rho(\mathbb{S}, \delta) = \frac{\mu^2}{2\sigma^2}$$

where μ and σ^2 are the mean and the variance of the distance distribution in $\mathbb{S}$ (proposed by Chávez et al. [Chávez et al. 2001]).

The intrinsic dimensionality is low if there are tight clusters of objects – some objects are close to each other and far from the other ones. If all the indexed objects are almost equally distant, then the intrinsic dimensionality is high, which

[4]Actually, there exist other definitions of intrinsic dimensionality, e.g., the fractal dimensionality [Faloutsos and Kamel 1994] or mapping dimensionality [Kao et al. 1997]. However, we consider the one presented in this paper as the most appropriate to similarity search purposes.

means the database is poorly intrinsically structured. A high ρ value says that many (even all) of partitions created on $\mathbb{S}$ are likely to be overlapped by every possible query, so the query processing deteriorates to a sequential search in all the partitions. The problem of high intrinsic dimensionality is, in fact, a generalization of the well-known *curse of dimensionality* [Weber et al. 1998; Chávez et al. 2001] into metric spaces.

Figure 7a shows an example of a distance distribution histogram of a database $\mathbb{S}$ under a metric δ. Since the mean of the distribution is quite high and the variance is low, the resulting intrinsic dimensionality is quite high as well. The negative impact of high intrinsic dimensionality on metric access methods can be recognized easily. Remember that the triangle inequality $a + b \geq c$ (which holds when a, b, and c are distances between objects) is the only filtering tool of any MAM. To be effective in filtering, by using distance estimators the $a + b$ component of the inequality must be lower than c (e.g., the sum of ball radii must be lower than the distance between the balls' centers). However, from the histogram we can observe that almost all distances sampled on the database objects are greater than the half of the maximum distance d^+, thus the sum of any two nonzero distances is likely to be greater than any other distance. Hence, the filtering by any MAM must fail on such a database.

A "mechanical" application of the intrinsic dimensionality for general nonmetric distances is possible, however, its value becomes questionable. Since the general nonmetrics do not satisfy the triangle inequality, we cannot evaluate a particular intrinsic dimensionality (the histogram, respectively) as too high for indexing, except for some exotic cases (e.g., a single value in the distribution).

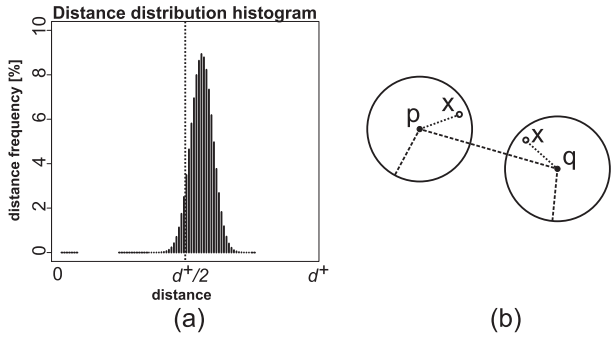

Fig. 7. (a) Distance distribution histogram exhibiting high intrinsic dimensionality (b) Incorrect filtering by nonmetric dissimilarity – a single object located in multiple "nonoverlapping" regions

4.3.2 *Ball-Overlap Factor.* Given a database and a dissimilarity function, the intrinsic dimensionality gives us a spaceless and indirect prediction (or cost model) about the indexing efficiency. However, instead of simple statistical properties like mean and variance of distances, we would like to capture rather an information about real relationships between data clusters described by some regions in the distance space. The regions should be of a shape/form used by metric access methods.

The *ball-overlap factor* (BOF) [Skopal 2007] was proposed as a simple region-based indicator, more suitable for predicting the efficiency of ball-based metric access methods, defined as

$$\mathrm{BOF}_k(\mathbb{S},\delta) = \frac{2}{|\mathbb{S}^*| * (|\mathbb{S}^*|-1)} \sum_{\forall o_i,o_j \in \mathbb{S}^*, i>j} sgn(|(o_i,\delta(o_i,kNN(o_i)))\bar{\cap} \bar{\cap}(o_j,\delta(o_j,kNN(o_j)))|)$$

where $\delta(o_i, kNN(o_i))$ is the distance to o_i's k-th nearest neighbor in a sample of the database $\mathbb{S}^*$ and $(o_i, \delta(o_i, kNN(o_i)))$ is thus the ball in metric space centered in o_i of radius $\delta(o_i, kNN(o_i))$. The statement $sgn((\cdot,\cdot)\bar{\cap}(\cdot,\cdot))$ returns 1 if the two balls overlap (in geometric-based, not data-based, meaning) and 0 if they do not. The ball overlap condition is defined as $\delta(o_i, kNN(o_i)) + \delta(o_j, kNN(o_j)) \leq \delta(o_i, o_j))$.

In simple words, the BOF_k calculates the ratio of overlaps between ball regions, where each region is made of an object (from the database sample) and of such a covering radius which guarantees k data objects are located inside the ball. In such a way the balls can be regarded as indexing regions. The overlap ratio then predicts the likelihood that two arbitrary ball-shaped regions will overlap or not. The BOF factor can thus serve as a more appropriate MAM-independent efficiency indicator for metric access methods based on ball partitioning, e.g., the M-tree.

Similarly as the intrinsic dimensionality, the ball-overlap factor cannot be quite correctly used for nonmetric distances. In Figure 7b see an illustration of two "nonoverlapping" balls in a nonmetric space (i.e., the sum of their radii is lower than the distance between their centers). Although in the metric case the balls cannot share any common object (so they really do not overlap), this is not true for the nonmetric case. Although all objects of one region have their distance to the region's center smaller than the radius, the absence of triangle inequality allows the objects to appear also in the second region, so the concept of region overlaps is more or less incorrect in nonmetric spaces. In consequence, when used with nonmetric distances the ball-overlap factor would produce a skewed information about the database indexability.

4.4 Strategies for Efficient Nonmetric Indexing

Unlike metric search, an efficient nonmetric indexing and search cannot rely on the metric postulates, however, the reflexivity, non-negativity and symmetry could be easily added to any nonmetric, as discussed in the next subsection. On the other hand, missing triangle inequality represents an absence of a powerful indexing tool. Based on this fact, we can choose from two strategies that substitute or adopt the full metric case:

4.4.1 *Statistical (data-driven) indexing.* The first strategy to nonmetric search is an endeavor to analyze the only available implicit description of the similarity function – the (portion of) distance matrix built on the database objects (possibly including also query objects). Based on the distance distribution statistics, the original database could be either transformed into another space (metric, vector, or other space), or suitably clustered. In the former case, the transformed database can be indexed by metric, spatial, or other access methods. The latter case (clus-

tering) could be used as a standalone access method. The drawbacks of statistical indexing is its restriction to more or less approximate search. Moreover, an expensive preprocessing is often needed prior to the indexing itself. The data-driven methods are overviewed in Sections 4.5 and 4.6.

4.4.2 *Indexing based on alternative analytic properties.* Instead of the metric postulates, a nonmetric function could be explicitly equipped by some other analytical properties, e.g., various relaxed forms of triangle inequality, transitivity rules, etc. The alternative properties could be used to develop specific indexing schemes or methods that are not dependent on the database statistics. In the extreme case, the analytic properties could be represented by the definition of a particular similarity function itself. The advantage of distance-specific access methods is their higher search efficiency, however, traded for application to only a limited class of similarity functions. Another advantage is an ability of both exact and approximate search. The distance-specific methods are overviewed in Section 4.7.

In the following sections, we survey existing approaches to nonmetric similarity search, including the statistical indexing, as well as indexing based on alternative properties.

4.5 Mapping

In this section, we discuss transformational approaches to nonmetric search, where the original nonmetric space is mapped into a target metric, vector or even Euclidean space. An advantage of mapping methods is the ability of subsequent indexing by some of the previously mentioned metric access methods (in case of target metric space), or by a *spatial access method* [Samet 2006; Böhm et al. 2001] (e.g., R-tree, X-tree, in case of target Euclidean or L_p space).

4.5.1 *Trivial Transformation of a Nonmetric into Metric.* Given a nonmetric similarity s, there are various trivial ways how to "automatically" turn s into a metric δ. The motivation for such a transformation might be a belief that database under the transformed metric space will be efficiently searchable by a metric access method. Unfortunately, any trivial (automatic) transformation usually leads to a failure. In particular, three of the four metric axioms are usually not a big problem (nonetheless, they may be in a particular case!), however, the triangle inequality mostly becomes the crucial obstacle. Let us analyze a transformation of general similarity into a metric.

Optionally, we need to turn similarity s into dissimilarity δ, which could be easily done by applying a monotonously decreasing function h on s (already discussed in Section 2.4.4), i.e., $\delta(\cdot,\cdot) = h(s(\cdot,\cdot))$. The non-negativity could be satisfied by shifting δ by the minimum distance d^- (i.e., $\delta(\cdot,\cdot) = h(s(\cdot,\cdot)) - d^-$)[5]. Concern-

[5]The minimum distance d^- and maximum distance d^+ could be provided by the dissimilarity function in case the structure of input universe $\mathbb{U}$ is known. Or, if it is not, we can sample a number of distances $h(s(x,y)), x, y \in \mathbb{S}$ and determine the approximate minimum/maximum distances. The outlier distances (exceeding the maximum distance or falling below the minimum distance) can be represented directly by d^- or d^+, while two objects falling into the "d^--bucket" are regarded as *at most* d^--distant. Similarly, two objects falling into the "d^+-bucket" are regarded as *at least* d^+-distant. When searching, the possibly relevant objects involved in outlier distances $\delta(q,x)$ (where q is a query object) are filtered sequentially in the original space $(\mathbb{S}, s)$.

ing the reflexivity property, we just declare that every two identical objects are zero-distant, while two non-identical are positively distant. That is, we shift the distance of two non-identical objects by a sufficiently small constant $\epsilon > 0$. Furthermore, searching by an asymmetric function δ_{asym} could be partially provided by a symmetric function δ_{sym}, e.g., $\delta_{sym}(x, y) = max\{\delta_{asym}(x, y), \delta_{asym}(y, x)\}$. Using the symmetric function, some non-relevant objects can be filtered out, while the original asymmetric function δ_{asym} is then used to rank the remaining non-filtered objects.

Finally, the triangle inequality can be enforced by an application of function

$$g(x) = \begin{cases} 0 & (\text{for } x = 0) \\ \frac{x+d^+}{2} & (\text{otherwise}) \end{cases}$$

which turns every d^+-bounded semimetric into a metric by shifting the distance into the upper half of the distance domain. Unfortunately, such a metric is useless for searching, since all classes of objects maintained by a MAM are overlapped by every query, so the retrieval always deteriorates to sequential search. This behavior is reflected by maximal ball-overlap factor, i.e., $\mathrm{BOF}_k(\mathbb{S}, \delta^g) = 1$. The intrinsic dimensionality $\rho(\mathbb{S}, \delta^g)$ is also high, however, it is not appropriate here since it does not recognize whether δ^g is useless for indexing or the database is just bad-structured. On the other hand, the distance distribution histogram used for determining the intrinsic dimensionality provides a better information – it contains only distances greater than $d^+/2$.

As the trivial transformations of semimetric spaces into metric ones are not suitable for efficient similarity search, in the following we overview some more useful transformational approaches.

4.5.2 *Constant Shifting Embedding.* A bit more advanced approach to the previously mentioned trivial distance shifting is the *constant shifting embedding* (CSE) [Roth et al. 2002]. Instead of just scaling the distances into the upper half of the distance domain, the CSE turns a semimetric into metric by adding a suitable constant c (i.e., $\delta'(\cdot, \cdot) = \delta(\cdot, \cdot) + c$). Obviously, an application of δ' instead of δ leads to fulfillment of the triangle inequality for a sufficiently large c.

In Roth et al. [Roth et al. 2002], the c value is set to the minimum eigenvalue of distance matrix that consists of pairwise distances among all the database objects. However, an analysis of the c-shifted distance matrix shows that this minimum eigenvalue is quite large, thus making it meaningless to prune by the triangle inequality. When employing just query objects from the database, CSE with the minimum-eigenvalue c provides an exact similarity search. However, the usability of CSE for efficient search is questionable, because the entire matrix must be computed, hence, the exact queries imply sequential search (a row for every possible query object is already computed in the distance matrix).

4.5.3 *The TriGen algorithm.* The recently introduced *TriGen algorithm* [Skopal 2007; 2006] can non-trivially put more or less of the triangle inequality into any semimetric δ, while keeping the database indexable, i.e., $\mathrm{BOF}_k(\mathbb{S}, \delta) < 1$. Thus, any semimetric distance can be turned into an equivalent (nearly) metric, allowing (almost) exact search by MAMs, or into a semimetric which satisfies the triangle inequality to some user-defined extent, allowing approximate search by MAMs.

For its functionality, the TriGen needs a (small) sample of the database objects. In fact, the TriGen algorithm generalizes the idea of constant shifting (see the previous section) by "functional" shifting. The original semimetric distance is modified by a function, making the resulting shifting value-sensitive.

The principle behind TriGen is a usage of triangle triplets and T-bases. A triplet of numbers (a, b, c) is *triangle triplet* if $a + b \geq c, b + c \geq a, a + c \geq b$. The triangle triplets can be viewed as witnesses of triangle inequality of a distance δ – if all triplets $(\delta(x, y), \delta(y, z), \delta(x, z))$ on all possible objects x, y, z are triangle triplets, then δ satisfies the triangle inequality. Using triangle triplets we measure the *T-error* – a degree of triangle inequality violation, computed as the proportion of non-triangle triplets in all examined distance triplets.

A *T-base* $f(v, w)$ is an increasing function (where $f(0, w) = 0$) which turns a value $v \geq 0$ of an input (semi)metric δ into a value of a target (semi) metric δ^f, i.e., $\delta^f(\cdot, \cdot) = f(\delta(\cdot, \cdot), w)$. Besides the input distance value v, the T-base is parameterized also by a fixed weight $w \in (-\infty, \infty)$ which determines how concave or convex f should be. The higher $w > 0$, the more concave f, which means also the lower T-error of any δ^f. Conversely, the lower $w < 0$, the more convex f and the higher T-error of any δ^f.

4.5.4 *Embeddings into Vector Spaces.* A bunch of methods was proposed for mapping a database modeled in metric or Euclidean space into Euclidean space (or generally into L_p spaces). In the context of similarity search, the benefits of mappings are two-fold. First, when mapping a metric space into L_p space, we aim to preserve the distance distribution as precise as possible, while reducing the complexity of the original metric distance to be linear with the dimensionality. Second, when mapping a high-dimensional L_p space into lower-dimensional L_p space, the mappings serve as a dimensionality reduction tool where the reduced dimensionality turns into smaller database and cheaper evaluation of L_p distance. A query-specific requirement on such mappings is an ability to quickly map an unknown query object into the target space, otherwise they could not be effectively employed in most similarity search tasks. From the dozens of mapping methods (designed to map from metric or vector spaces) we name FastMap [Faloutsos and Lin 1995], MetricMap [Wang et al. 2000], SparseMap [Hristescu and Farach-Colton 1999], and BoostMap [Athitsos et al. 2004]. Regardless of a particular technique, the mapping methods exhibit several major drawbacks. The mapping is often computationally expensive (not scalable with database size), approximate (often without any error bounds), and inherently static (i.e., newly added/mapped objects increase the approximation error). To the best of our knowledge we are not aware of a particular application of the mentioned metric mapping methods for the nonmetric case. However, we believe they might be used also for mapping nonmetric data, since all of the mentioned methods use a criterion on preserving the distances in the target space, regardless of Euclidean, general metric, or general nonmetric source space. Moreover, the following "explicitly nonmetric" approaches are based on more or less similar principles as the metric methods (a kind of distance-preserving criterion). Anyways, such a hypothesis needs to be verified in the future.

Among the explicitly nonmetric approaches, the *nonmetric multidimensional scaling* (NMDS) represents the classic approach to embedding nonmetric spaces.

In particular, the Shepard-Kruskal scaling (SKS) algorithm [Shepard 1962; Kruskal 1964] creates an embedding of an arbitrary symmetric zero-diagonal dissimilarity matrix (representing semimetric pairwise distances between objects from $\mathbb{S}$) into Euclidean vector space. The SKS embedding was posed as an optimization problem, where the stress-1 functional (measuring the aggregated error between the original and mapped L_2 distances) is being minimized. The optimization procedure involves two parameters – the actual mapping X of $\mathbb{S}$ into multidimensional Euclidean space, and the actual monotonic function θ (applied on the original dissimilarities in the stress function). The algorithm iterates two alternating steps. First, having θ fixed, the mapping X is fit to a particular transformation of the dissimilarities using gradient descent. Second, having X fixed, θ is aligned with the actual mapping X with isotonic regression. The classic Shepard-Kruskal algorithm is extensively used in data mining (i.e., classification and clustering rather than similarity search).

Another nonmetric approach to mappings are the *query-sensitive embeddings* [Athitsos et al. 2005] (a nonmetric extension of the BoostMap algorithm), where the description of target distance changes depending on the query object employed. The target distance is similar to weighted L_1 distance, but it is not a metric because the weights change for each query object (for the object x in $\delta(x, y)$, respectively). Using the AdaBoost algorithm the method trains a number of classifiers, which are used to map the source objects into the target vectors (each classifier maps an object into one dimension of its target vector). Experiments have shown that query-sensitive embeddings outperform the other mapping methods in both the embedding precision (effectiveness) and the search efficiency, when searching the embedded database using a kind of filter-and-refine strategy.

4.5.5 *Different Paradigms.* As yet, we have considered mapping of nonmetric space into a metric or even L_p space. Although there exist many mappings of this kind, they all share the metric paradigm – the transformed similarity function and/or data are "metrized" in order to be indexable by use of the usual metric postulates.

However, a mapping approach could be based on completely different paradigm, replacing the metric postulates by alternative properties. A recent example could be transformation of the nonmetric search problem into the realm of fuzzy logic [Vojtáš and Eckhardt 2009; Eckhardt et al. 2009]. Instead of modifying the similarity function and/or data, the fuzzy approach is based on adjustment of the fuzzy logic formulas that are used for effective filtering of database objects. In particular, instead of triangle inequality, the fuzzy approach provides a kind of transitive inequality $s(x, z) \geq T(s(x, y), s(y, z))$, where T is a fuzzy conjunction. Furthermore, given a residuation $T(x, y) \leq z \rightarrow I_T(x, z) \geq y$ (where I_T is a fuzzy implication residual to T), there can be derived similarity estimation between a query q and a database object x (by use of a pivot p) by the following formulas:

$$T(s(q, p), s(p, x)) \leq s(q, x)$$

$$s(q, x) \leq min\{I_T(s(q, p), s(p, x)), I_T(s(p, x), s(q, p))\}$$

The above formulas are analogous to the lower bound and upper bound inequal-

ities used for filtering in the metric case model (see Section 4.2), however, here the similarity function s (being similarity, not dissimilarity) is not restricted by metric postulates. On the other hand, to index a database under an arbitrary similarity s, there must be a conjunction T found so that the transitivity property holds. As a conjunction we could choose the well-known parametric family of Frank t-norms

$$T_\lambda^F(v,w) = log_\lambda\left(1 + \frac{(\lambda^v - 1)(\lambda^w - 1)}{\lambda - 1}\right)$$

where the λ parameter could be tuned as necessary. Here we can observe some duality or analogy to the "metrization" of semimetric space (provided by the TriGen algorithm, see Section 4.5.3). Similarly to T-bases in TriGen, also here we solve a problem with finding the right parameter λ that guarantees the transitivity of similarity scores among the objects in database.

4.6 General Nonmetric Access Methods

In the following we discuss three general methods for nonmetric indexing, which we could pronounce as *nonmetric access methods* (NAMs). Although all of them reuse mapping approaches already mentioned in Section 4.5, they consider a more complex scenario, including algorithms for indexing and querying.

4.6.1 *NM-tree.* The recently introduced *NM-tree* [Skopal and Lokoč 2008] applies the indexing power of M-tree into the realm of nonmetric spaces by use of the TriGen algorithm (described in Section 4.5.3). Using the TriGen algorithm, an input semimetric is turned into a (nearly) full metric, i.e., preserving zero T-error. The metric is subsequently used as the indexing metric in M-tree, hence, the database could be efficiently queried by (almost) exact nonmetric queries. Moreover, the NM-tree allows faster approximate nonmetric search, where the desired retrieval error is specified by the user at query time. Actually, when the T-modifier leading to exact metric is determined by the TriGen algorithm (prior to the indexing), there are also other modifiers determined for various levels of nonzero T-error tolerance. At query time the user can specify a threshold T-error level (the associated T-modifier, respectively), while the metric distances stored in the NM-tree are re-interpreted to more or less nonmetric ones, with respect to the T-modifier used. Because of its lower intrinsic dimensionality, searching using the modified metric leads to more efficient (but approximate) query processing.

4.6.2 *QIC-M-tree.* The *QIC-M-tree* [Ciaccia and Patella 2002] has been proposed as an extension of the M-tree (the key idea is applicable to other MAMs), allowing (not only) nonmetric similarity search. The M-tree index is built by use of an index distance δ_I, which is a metric lower-bounding the query distance δ_q (up to a scaling constant $S_{I\to q}$), i.e. $\delta_I(x,y) \leq S_{I\to q}\delta_q(x,y), \forall x,y \in \mathbb{U}$. As δ_I lower-bounds δ_q, a query can be partially processed by δ_I (which, moreover, could be computationally much cheaper than δ_q), such that many non-relevant classes of objects (subtrees in M-tree) are filtered out. All objects in the non-filtered classes are compared against q using δ_q. Actually, this approach is similar to the usage of contractive mapping methods, but here the objects generally need not to be mapped into a vector space. However, this approach has two major limitations. First, for

a given nonmetric distance δ_q there is no general way how to find the metric δ_I. Although δ_I could be found "manually" for a particular δ_q (as in [Bartolini et al. 2005]), this is not possible for δ_q given as a black box. Second, the lower-bounding metric should be as tight approximation of δ_q as possible, because this "tightness" heavily affects the intrinsic dimensionality, the number of MAMs' filtered classes, and so the retrieval efficiency. Although both QIC-M-tree and NM-tree (Section 4.6.1) originate from M-tree, they are applicable under different conditions (NM-tree is more general but could be less efficient, and vice versa).

4.6.3 *Local Constant Embedding.* Another recently introduced technique – the *Local Constant Embedding* (LCE) [Chen and Lian 2008] – was inspired by the Constant Shifting Embedding (see Section 4.5.2). However, instead of a single global constant, in LCE the database is partitioned into multiple groups, where each group G_i owns a local constant c_i. The motivation for LCE is an expectation that a suitably partitioned database would lead to lower constants c_i, which, in turn, would reflect in more effective pruning of particular database groups when searching. However, the construction of groups G_i and their constants c_i is not that simple because a so-called grouping property must be satisfied. For every triplet of objects $x, y \in G_i$ and $z \in G_j$, it must hold $max(M) - (min(M) + min(M - \{min(M)\})) \leq c_i$, where $M = \{\delta(x,y), \delta(x,z), \delta(y,z)\}$. The authors of LCE have proposed a heuristic algorithm of cubic complexity that produces groups satisfying the grouping property, while keeping the local constants as low as possible. When querying by database objects, the grouping property guarantees no false dismissals when pruning the groups, i.e., an exact search. In addition to the main idea, the authors have proposed an indexing schema based on mapping the objects using pivots (each group has its own pivots) and subsequent indexing by iDistance [Jagadish et al. 2005] (i.e., mapping + B^+-tree based indexing).

Similarly as CSE, the drawback of LCE is the need to compute the entire distance matrix on the database, which makes the exact search using LCE (involving just query objects from the database) meaningless. Nevertheless, for query objects outside the database the LCE can be used for approximate similarity search.

4.6.4 *Clustering & Classification.* Cluster analysis is an essential task in many application domains. It allows one to find natural clusters and describe their properties (*data understanding*), find useful and suitable groupings (*data class identification*), find representatives for homogeneous groups (*data reduction*), find unusual objects (*outliers detection*), find random perturbations of the data (*noise detection*), and so on. A clustering algorithm identifies a set of categories, classes, or groups (called *clusters*) in the database, such that objects within the same cluster shall be as similar as possible, and objects from different clusters shall be as dissimilar as possible.

Standard clustering algorithms, like k-means, assume that the objects are points in a vector space and that the distance used is a metric (e.g., Euclidean distance). For nonmetric spaces, there are a few algorithms that have been proposed in the literature. For example, Su and Chou [Su and Chou 2001] proposed a k-means algorithm with a nonmetric distance, so-called *point symmetry distance*. Also, Becker and Potts [Becker and Potts 2007] presented a nonmetric clustering method

based on distances to so-called *fiduciary templates* (some selected random objects from the set). The distances to these fiduciary templates form a vector, which is used to decide to which cluster a new object belongs. More recently, Ackermann et al. [Ackermann et al. 2008] proposed a k-median clustering algorithm for nonmetric functions (specifically, the Kullback-Leibler divergence), that computes a $(1+\varepsilon)$-approximation of the k-median problem.

Quite many attempts to nonmetric nearest neighbor (NN) search have been tried out in the classification area. Let us recall the basic three steps of classification. First, the database is organized in classes of similar objects (by user annotation or clustering). Then, for each class a description consisting of the most representative object(s) is created; this is achieved by *condensing* [Hart 1968] or *editing* [Wilson 1972] algorithms. Third, the NN search is accomplished as a classification of the query object. Such a class is searched, to which the query object is "nearest" – there is an assumption the nearest neighbor is located in the "nearest class". For nonmetric classification there have been proposed methods enhancing the description of classes (step 2). In particular, condensing algorithms producing *atypical points* [Goh et al. 2002] or *correlated points* [Jacobs et al. 2000] have been successfully applied. The drawbacks of classification-based methods reside in static indexing and limited scalability, while the querying is restricted just to approximate (k-)NN.

4.6.5 *Combinatorial approach.* A recent framework for similarity search based on a combinatorial approach has been proposed by Lifshits [Lifshits 2009]. In this framework, the only available information is the one provided by a *comparison oracle*, which given three objects x, y, z answers whether $s(x,y) > s(x,z)$ or vice versa. Additionally, the rank $rank_x(y)$ of y with respect to x is defined as the position of y in the ranking of elements in $\mathbb{S}$ ordered by similarity from x in decreasing order. Then, the *disorder inequality* is defined as

$$rank_y(x) \leq D \cdot (rank_z(x) + rank_z(y)),$$

where D is the *disorder constant*. This property is then used for performing efficient similarity search. Combinatorial algorithms, that is, algorithms that only rely on the comparison oracle to perform the search, for the nearest neighbor search have been proposed [Goyal et al. 2008]. Note that this approach works with similarity functions (thus, triangle inequality does not hold), and it also does not assume that s holds symmetry. Thus, the combinatorial approach may be a useful tool to deal with complex data representations and similarity functions. However, finding an optimal disorder constant is not easy.

4.7 Distance-specific Nonmetric Access Methods

Because of the absence of triangle inequality, the nonmetric methods need to compensate this deficiency by some other tools. In the previous sections, we have discussed statistical approaches, that is, an "aid" obtained using statistical processing of the indexed database. In this section, we overview several methods which follow the opposite strategy – instead of preprocessing data, the dissimilarity function is supposed to be equipped by an alternative topological property useful for indexing. Such a property does not depend on the database distribution, hence, any data preprocessing is not necessary. In its extreme form, a nonmetric method

could be designed for a single particular dissimilarity function, where formula of the similarity function itself plays the role of the requested topological property.

4.7.1 *General Alternative Properties.* A method for exact nearest neighbor search in constrained nonmetric spaces was proposed by Farago et al. [Farago et al. 1993]. Instead of triangle inequality, the nonmetric distance is expected to be equipped by suitable pivots $p_1, p_2 \ldots, p_k \in \mathbb{U}$ and suitable numbers α, β, such that for any $x, y \in \mathbb{S}$ the following relaxed bounding property holds:

$$\alpha\delta(x, y) \geq |\delta(x, p_i) - \delta(y, p_i)|, i = 1, \ldots, k$$

and

$$max_{1 \leq j \leq k} |\delta(x, p_j) - \delta(y, p_j)| \geq \beta\delta(x, y)$$

Obviously, the triangle inequality is a special case of the above property (then, e.g., $\alpha = 1, \beta = \frac{1}{2}$ + any set of pivots). On the other hand, for any $\alpha \to \infty$ and $\beta \to 0$ the property holds for any nonmetric. However, an efficient similarity search using the relaxed bounding property could be only useful when the α is as low as possible and β is as high as possible (while still guaranteeing the exact search). The authors proposed a nearest neighbor algorithm which makes use of the relaxed bounding property. The complexity of nearest neighbor search is $O(n)$ in the worst case, while the authors provide a probabilistic analysis showing that in average case the nearest neighbor could be accomplished in O(1) time[6]. The optimal choice of the external parameters (the pivot set and α, β) is left to the provider of particular nonmetric distance (the domain expert).

4.7.2 *Inverted file.* The classic inverted file, used for implementation of vector model in text retrieval under cosine measure [Berry and Browne 1999], represents a domain-specific exact nonmetric access method. In vector model, a text document is represented by vector of term weights, where each weight stands for significance of a given term in the document. The query is represented the same (as a vector of weights of terms in the query). For each term, the inverted file maintains a list of documents (their identifiers, resp.) which have nonzero weight for that term.

The implementation of a vector query using inverted file is simple. Because the cosine measure (or dot product) is used as the similarity function, it is sufficient to process only the lists of those terms which contribute to the query. Because a typical query consists of only a few terms, there must be only a few lists processed, so the query processing by inverted file is extremely efficient. However, note that the advantage of query processing by inverted file is only possible due to the usage of cosine measure. The lists belonging to non-query terms can be skipped because their weights in the query are zero, while the cosine measure applies multiplication of weights (which would lead to zero for any weight in the list).

The inverted file is an excellent example that nonmetric access method could be more efficient than a metric access method, even for the same case. If we replace

[6]However, the nearest neighbor search exhibits an exponential dependence on dimensionality of some embedding of the database. This fact makes the O(1) result quite questionable, because the dimensionality of the embedding depends on the database size.

the cosine measure by Euclidean distance and normalize the document vectors to become unitary, we get the same similarity search model (providing the same similarity ranking as the cosine measure does). However, a usage of inverted file under Euclidean distance is inefficient, because the query weights are being summed with the document weights, hence, we cannot skip the lists belonging to non-query terms.

4.7.3 *IGrid.* Inspired by the inverted file, the *IGrid* (inverted grid) was proposed for high-dimensional vector indexing under a nonmetric modification of L_p distances [Aggarwal and Yu 2000]. Unlike classic L_p distances, the proposed L_p-inspired distance was designed to "flatten" the distance distribution in the database, regardless of the dimensionality of the data space. The flatness of distance distribution was achieved by ignoring those dimensions of two data vectors, which values were "too distant". In particular, let us assume that each dimension of the data space is divided into k_d ranges. The ranges $[n_i, m_i]$ have to be equi-depth, that is, each range is matched by a fraction $1/k_d$ of the total number of vectors in the database. When measuring the distance of two vectors x, y, we ignore those dimensions where the values of the two vectors do not share a single range. In other words, we pick only those dimensions from the data space, where the vectors x, y are sufficiently close – such a subset of close dimensions is called the proximity set, denoted as $\mathcal{S}[x, y, k_d]$. The similarity function is then defined as

$$PI_{dist}(x, y, k_d) = \left[\sum_{i \in \mathcal{S}[x,y,k_d]} \left(1 - \frac{|x_i - y_i|}{m_i - n_i}\right)^p \right]^{\frac{1}{p}}$$

To efficiently search according to the proposed similarity function, the IGrid index maintains a 2D grid of size $d \times k_d$ (i.e., the dimensionality times the number of ranges). Each cell of the grid references a list of relevant vectors ids, that is, those vectors whose values fall into the given range at a given dimension.

The IGrid approach killed even four birds with one stone. First, it established a robust nonmetric similarity function that suppressed the extreme differences in coordinate values (even more effectively than the fractional L_p distances did, see Section 3.1.1). Second, the uniformity of distance distribution means low intrinsic dimensionality (see Section 4.3.1) which allows to efficiently index a database. Third, the distribution uniformity is achieved regardless of the data space dimensionality, thus, reversing the curse of dimensionality to some extent. Fourth, even though the similarity function is nonmetric, the IGrid provides exact similarity search. On the other hand, the IGrid similarity function is context-dependent (and thus not stable), as it depends on the data distribution.

4.7.4 *Indexing DTW and LCS.* Although DTW is a nonmetric distance (see Section 3.1.7), it is possible to build an exact index for DTW. The idea is to compute a lower bounding function of DTW. This can be used, for example, to implement search algorithms based on filtering and refinement [Faloutsos 1996], or to employ the QIC-M-tree (see Section 4.6.2). Several lower bounds for the DTW have been proposed so far [Yi et al. 1998; Kim et al. 2001], but the lower bound proposed by Keogh and Ratanamahatana [Keogh and Ratanamahatana 2005] (the

so-called LB-Keogh) has been shown to be in practice very tight.

LB-Keogh makes some assumptions on how the DTW is computed. First, it assumes that both sequences have the same length. Second, it assumes that the warping path is constrained, that is, there is a limit on how far from the diagonal the path can be. Keogh and Ratanamahatana discuss in their paper that, in practice, both assumptions do not really add restrictions to the DTW [Keogh and Ratanamahatana 2005]. The authors adopted the GEMINI framework [Faloutsos et al. 1994], that, given a lower-bounding function to a query distance function, provides fast and exact similarity search in time series databases by use of a spatial access method (e.g., the R*-tree [Beckmann et al. 1990]). The actual index (let us call it GEMINI(LB-Keogh) because authors did not introduce a specific name) maps first the time series to low-dimensional vectors by piecewise aggregate approximation [Keogh et al. 2000]; then the vectors are packed into minimum bounding rectangles (MBR). The defined distance between vectors is proved to be a lower bound of the LB-Keogh (and thus, by transitivity, a lower bound of the DTW). Then, another lower bound distance is defined between a vector and an MBR. Finally, a kNN algorithm, very similar to the incremental search algorithm proposed by Hjaltason and Samet [Hjaltason and Samet 1995], is presented. This search algorithm ensures that the exact answer will be always retrieved. Similarly, an upper bound value has been defined for the longest common subsequence (LCS) [Vlachos et al. 2003], which may be used also for indexing purposes.

4.7.5 *Similarity Search in Protein Databases.* In the area of bioinformatics, the Smith-Waterman algorithm for local sequence alignment (SW, see Section 3.4.1) is the most popular similarity function for protein sequences. Nowadays, the search using SW is implemented in *ScanPS* [Barton 2002] (original SW algorithm enhanced by so-called iterative profile search), *SSEARCH* [Green 1993] (SW with SWAT optimizations), or *MPsrch* (parallelized version of the original SW algorithm)[7].

The SW similarity gives the optimal solution to local alignment but it is computationally expensive. Hence, there have been developed cheaper heuristics which approximate the SW alignment. The first wide-spread method of this type was the *FASTA* [Lipman and Pearson 1985], while nowadays the *BLAST* (Basic Local Alignment Tool) [Altschul et al. 1990] algorithm is widely used. In most cases, BLAST is faster than FASTA and gives better results (i.e., it suffers from less false dismissals with respect to sequential search using SW alignment). Both FASTA and BLAST search for seeds – contiguous fragments that are present in both of the compared sequences. The seeds are then extended to larger portions to achieve a final alignment. In particular, FASTA searches for totally aligned subsequences, where each of them is valuated by use of a PAM scoring matrix. Then, top k of the identical alignments is embraced by a narrow belt (in the alignment matrix), a Needleman-Wunch alignment (NW, see Section 3.4.1) is computed on the reduced matrix (the belt) which noticeably decreases the number of necessary computations. Only sequences with sufficiently low NW score are passed to the final step where full SW alignment is performed.

On the other hand, BLAST divides a query sequence into n-grams (for proteins

[7]Online at `http://www.ebi.ac.uk/Tools/MPsrch/index.html` (12/2009).

$n = 3$, typically), where for each n-gram N the set of all such n-grams (from the n-gram universe) is determined and stored into an n-gram index, that are more similar to N than a given threshold. Then, those database sequences are identified that match an n-gram in the n-gram index. A particular match represents a rough local alignment of the query sequence and a database sequence, while the alignment is extended until the alignment score is higher than a given threshold. Finally, the scores of the aligned database sequences are computed while those not having sufficiently good alignment are dismissed. The remaining candidates are refined by the classic SW algorithm.

4.8 Summary

Table III shows a summary and a comparison of the techniques presented in this section, including the sequential scan as a naïve method for searching in nonmetric spaces. The main characteristics of techniques detailed in the table are: index designed for general-purpose nonmetric or distance-specific index; index performs exact or approximate search; index is static or allows insertions/deletions; index is build in main memory or it can be natively maintained in secondary memory; other characteristics or comments on the technique.

We observe that the available exact techniques are either slow in the general case (e.g., sequential scan), require large amounts of space (e.g., CSE), require a deeper knowledge of the nonmetric function (e.g., QIC-M-tree), or are indeed efficient but distance-specific (e.g., Inverted file). NAMs that implement approximate similarity search may cope better with general nonmetric functions (e.g., NM-tree). Most of the index structures are static, except (mainly) those based on the M-tree.

5. CONCLUSIONS

In this paper, we have surveyed the current situation concerning the employment of nonmetric similarity functions for effective and efficient similarity search in complex domains. One of the main results of the paper is a surprising revelation that nonmetric similarity measuring is widely used in isolated domains, spanning over many areas of interdisciplinary research. This includes multimedia databases, time series, medical, scientific, chemical and bioinformatic tasks, among others.

The assembled mosaic shows us that the need for an employment of nonmetric similarity functions gets the stronger the more increasing complexity of the underlying data space model is. Since simple descriptors (like low-dimensional vectors) become insufficient and more complex descriptors replace them often (like high-dimensional vectors, time series, 3D meshes, graphs, etc.), the classic metric paradigm becomes quickly less effective.

Another contribution of this paper is a summary of the existing database indexing techniques suitable for efficient nonmetric search. Although there "hesitantly" appear some pioneer approaches, the vast majority of similarity indexing techniques still relies on the metric space restriction.

So far, the domain communities use their nonmetric techniques (mostly) without any database indexing support, i.e., they use a simple sequential search over a set of descriptors. However, as the volumes of all kinds of complex data expand tremendously, the lack of efficient data processing would slow in the near future any further progress in solving the domain tasks. Hence, it is our belief that the

Table III. Overview of Nonmetric Access Methods

	Method	specialized/ general	approximate/ exact search	static/dynamic database	main-memory/ persistent	other characteristics	details in section
	Sequential scan	Gen.	Exact	Dynamic	Both	Requires no index	n/a
mapping	CSE	Gen.	Exact	Static	Main-mem.	Requires $O(n^2)$ space	4.5.2
mapping	TriGen	Gen.	Approx.	Static	Main-mem.	Simplifies the problem to metric case	4.5.3
mapping	Embeddings into vector spaces	Gen.	Approx.	Static	Main-mem.	Simplifies the problem to L_p space	4.5.4
mapping	Fuzzy logic	Gen.	Approx.	Static	Main-mem.	Provides transitive inequality, not implemented yet	4.5.5
general NAMs	NM-tree	Gen.	Approx.	Dynamic	Persistent	Based on M-tree, uses TriGen	4.6.1
general NAMs	QIC-M-Tree	Gen.	Exact	Dynamic	Persistent	Based on M-tree, requires user-defined metric lower bound distance	4.6.2
general NAMs	LCE	Gen.	Approx.	Static	Main-mem.	Exact only for database objects	4.6.3
general NAMs	Classification	Gen.	Approx.	Static	Main-mem.	Requires cluster analysis, limited scalability	4.6.4
general NAMs	Combinatorial approach	Gen.	Approx.	Static	Main-mem.	No implementation yet, only for NN search. Exact for large enough D.	4.6.5
specific NAMs	Inverted file	Spec.	Exact	Dynamic	Persistent	Cosine measure	4.7.2
specific NAMs	IGrid	Spec.	Exact	Static	Main-mem.	Specific L_p-like distance	4.7.3
specific NAMs	GEMINI(LB-Keogh)	Spec.	Exact	Both	Main-mem.	Uses lower bound distances	4.7.4
specific NAMs	FASTA/BLAST	Spec.	Approx.	Dynamic	Main-mem.	Approximate alignment	4.7.5

database research community should pay an increased attention to the nonmetric similarity search techniques.

5.1 Challenges for Database Research

The increasing efforts spent on employing nonmetric similarities in various areas demonstrates that the nonmetric paradigm is a viable and desired generalization of the classic metric approach. As the similarity search problems have started to originate from more complex domains than before, the database community will have to take into consideration the nonmetric similarity search approach. In the following, we outline five directions for future research in nonmetric similarity indexing:

5.1.1 *Scalability.* Today, the most common mean of nonmetric searching is the sequential scan of the database. As such a trivial way of retrieval is sufficient only for a low number of descriptors in the database, it is not scalable with database size. In the future, the lack of efficient access methods could be a bottleneck for the domain experts because the database sizes tend to increase even in domains that used to be not data intensive (e.g., protein engineering). Thus, it is necessary to design

new indexing techniques specifically designed to tackle with nonmetric spaces. In particular, the streaming databases could benefit from nonmetric similarity search. Here a concept similar to index-free metric search [Skopal and Bustos 2009] could be applied.

5.1.2 *Indexability.* In Section 4.3 we have discussed the indexability concepts related to metric space. Although nonmetric problem can be turned into metric one and a metric indexability concept used, this might not be the best solution of the nonmetric space analysis. However, due to lack of information provided by black-box nonmetric function a native indexability concept for nonmetric problems is not possible. A way how to better analyze a nonmetric space could be a combination of multiple indexability concepts, each applicable to different mapping of the original problem (e.g., to metric space, fuzzy logic, or some future one). Of course, a particular indexability model for specific nonmetric function and/or nonmetric access method could be based on the specific properties of the problem.

5.1.3 *Implementation Specificity.* One way how to achieve an efficient and scalable nonmetric search is the design of made-to-measure access methods, where each method is designed specifically for a particular similarity function. The inverted file and the cosine measure could be an example (see Section 4.7.2). While specific access methods cannot be reused for other similarity functions (becoming thus single-purpose), it might turn out that this is the only viable solution within the restrictions given by the nonmetric similarity search approach. In other words, although a general nonmetric access method would be more universal (in terms of its reusability in various problems), its employment might not speedup the search enough, making the whole similarity search problem infeasible.

5.1.4 *Efficiency vs. Effectiveness.* In the area of similarity search (metric and nonmetric), one must not forget that the main objective is to retrieve *relevant* objects according to the query specifications made by the user. Indeed, it makes no sense to have a very efficient similarity search system if it will only return "garbage". Moreover, the effectiveness of the search system can only be improved with better search algorithms (as opposed to efficiency, which may be improved with better hardware). Thus, it is absolutely relevant to design algorithms that are not only performing queries fast, but also are highly effective (depending of course on the specific application domains).

Given the characteristics of searching in nonmetric spaces, this may imply focusing the research on approximate and probabilistic techniques. These techniques, at least in the case of metric spaces, have been shown to give a good trade-off between efficiency and effectiveness regarding to similarity search. If one does not have any information about the similarity function (like in black-box nonmetric similarity), then every technique designed for discarding objects may produce false negatives, thus approximation cannot be avoided. This may be enough for many application where the similarity function is already an approximation of the human notion of similarity. However, if one needs 100% recall with the given similarity function, then the only available solution is the sequential scan. Thus, for this sceneario, the proposal of an efficient sequential scan algorithm (like the VA-File [Weber et al. 1998] for the case of vector spaces) would be very useful.

5.1.5 *Extensibility.* Because of the simple assumptions on the syntax of pairwise similarity function, the mechanisms of similarity search might be applied also in contexts where the (dis)similarity function is interchangeable with another "syntactically compatible" aggregating/scoring function. In fact, any function that accepts two data descriptors and returns a number could be interpreted as a pairwise nonmetric (dis)similarity function. Thus, the existing techniques of nonmetric similarity search might be reused in different retrieval scenarios, where a scoring function determines the relevance of database objects to a query. Conversely, the area of similarity search might profit from existing or future retrieval approaches developed under different circumstances. Actually, a seeming difference between similarity search and other kinds of retrieval that use pairwise scoring could reside just in different terminology. In particular, we name some terms to be possibly treated as pairwise similarity, like distance, correlation, transformation/unification cost, probability, matching, and alignment.

From the implementation point of view, the domain-specific retrieval techniques (such as BLAST, see Section 3.4.1, or retrieval of 3D models, see Section 3.2.2) are often designed as one complex "monolithic" algorithm designed by a domain expert, where the similarity function is not explicitly declared as an independent module. For such a domain-specific technique, it may emerge a requirement on improving its efficiency (the speed of retrieval) in the future, especially when the algorithm involves sequential scan of the database. In such case it might be beneficial to separate the actual similarity function from the existing retrieval algorithm. The retrieval algorithm could be consecutively re-developed to take advantage of an existing (non)metric access method, or to design a new domain-specific access method, which might lead to a more efficient search technique.

REFERENCES

Ackermann, M., Blömer, J., and Sohler, C. 2008. Clustering for metric and non-metric distance measures. In *Proc. 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'08)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 799–808.

Aggarwal, C., Hinneburg, A., and Keim, D. 2001. On the surprising behavior of distance metrics in high dimensional spaces. In *Proc. 8th International Conference on Database Theory (ICDT'01)*. Springer-Verlag, London, UK.

Aggarwal, C. and Yu, P. 2000. The IGrid index: reversing the dimensionality curse for similarity indexing in high dimensional space. In *Proc. 6th ACM International Conference on Knowledge Discovery and Data Mining (KDD'00)*. ACM Press, New York, NY, USA, 119–129.

Altschul, S., Gish, W., Miller, W., Myers, E., and Lipman, D. 1990. Basic local alignment search tool. *Journal of Molecular Biology 215,* 3 (Oct), 403–410.

Angeles-Yreta, A., Solis-Estrella, H., Landassuri-Moreno, V., and Figueroa-Nazuno, J. 2004. Similarity search in seismological signals. In *Proc. 6th Mexican International Conference in Computer Science (ENC'04))*. IEEE Computer Society, Washington, DC, USA, 50–56.

Antani, S., Lee, D., Long, L., and Thoma, G. 2004. Evaluation of shape similarity measurement methods for spine X-ray images. *Journal of Visual Communications and Image Representation 15,* 3, 285–302.

Ashby, F. and Perrin, N. 1988. Toward a unified theory of similarity and recognition. *Psychological Review 95,* 1, 124–150.

Ashby, F. G., Ed. 1992. *Multidimensional models of perception and cognition.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Athitsos, V., Alon, J., Sclaroff, S., and Kollios, G. 2004. Boostmap: A method for efficient

approximate similarity rankings. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*. IEEE, 486–493.

Athitsos, V., Hadjieleftheriou, M., Kollios, G., and Sclaroff, S. 2005. Query-sensitive embeddings. In *Proc. ACM International Conference on Management of Data (SIGMOD'05)*. ACM Press, New York, NY, USA, 706–717.

Baeza-Yates, R. and Ribeiro-Neto, B. 1999. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Bartolini, I., Ciaccia, P., and Patella, M. 2005. WARP: Accurate retrieval of shapes using phase of fourier descriptors and time warping distance. *IEEE Pattern Analysis and Machine Intelligence 27,* 1, 142–147.

Barton, G. J. 2002. SCANPS Version 2.3.9 User guide. University of Dundee, UK.

Becker, G. and Potts, M. 2007. Non-metric biometric clustering. In *Proc. Biometrics Symposium*. 1–6.

Beckmann, N., Kriegel, H.-P., Schneider, R., and Seeger, B. 1990. The R*-tree: an efficient and robust access method for points and rectangles. In *Proceedings of the 1990 ACM SIGMOD international conference on Management of data*. ACM Press, 322–331.

Berndt, D. and Clifford, J. 1994. Using dynamic time warping to find patterns in time series. In *Proc. AAAI Workshop On Knowledge Discovery in Databases*. 229–248.

Berry, M. and Browne, M. 1999. *Understanding Search Engines, Mathematical Modeling and Text Retrieval*. Siam.

Berry, M., Dumais, S., and Letsche, T. 1995. Computation methods for intelligent information access. In *Proc. ACM/IEEE Supercomputing Conference*.

Blanken, H. M., de Vries, A. P., Blok, H. E., and Feng, L. 2007. *Multimedia Retrieval*. Springer.

Böhm, C., Berchtold, S., and Keim, D. 2001. Searching in high-dimensional spaces – Index structures for improving the performance of multimedia databases. *ACM Computing Surveys 33,* 3, 322–373.

Brin, S. 1995. Near neighbor search in large metric spaces. In *Proc. 21st Conference on Very Large Databases (VLDB'95)*. Morgan Kaufmann, 574–584.

Bustos, B., Navarro, G., and Chávez, E. 2003. Pivot selection techniques for proximity searching in metric spaces. *Pattern Recognition Letters 24,* 14, 2357–2366.

Bustos, B. and Skopal, T. 2006. Dynamic similarity search in multi-metric spaces. In *Proc. 8th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR'06)*. ACM Press, 137–146.

Buttler, D. 2004. A short survey of document structure similarity algorithms. In *Proc. 5th International Conference on Internet Computing (IC'04)*. CSREA Press, 3–9.

Cha, G.-H. 2006. Non-metric similarity ranking for image retrieval. In *Proc. 17th International Conference on Database and Expert Systems Applications (DEXA'06)*. LNCS 4080. Springer, 853–862.

Cha, S.-H., Yoon, S., and Tappert, C. 2005. On binary similarity measures for handwritten character recognition. In *Proc. 8th International Conference on Document Analysis and Recognition (ICDAR'05)*. IEEE Computer Society, 4–8.

Chávez, E. and Navarro, G. 2001. A Probabilistic Spell for the Curse of Dimensionality. In *ALENEX'01, LNCS 2153*. Springer, 147–160.

Chávez, E., Navarro, G., Baeza-Yates, R., and Marroquín, J. 2001. Searching in metric spaces. *ACM Computing Surveys 33,* 3, 273–321.

Chen, L. and Lian, X. 2008. Dynamic skyline queries in metric spaces. In *EDBT '08: Proceedings of the 11th international conference on Extending database technology*. ACM, New York, NY, USA, 333–343.

Chen, L., zsu, M. T., and Oria, V. 2005. Robust and fast similarity search for moving object trajectories. In *PROC. ACM International Conference on Managment of Data (SIGMOD'05)*. ACM, 491–502.

Ciaccia, P. and Patella, M. 2002. Searching in metric spaces with user-defined and approximate distances. *ACM Database Systems 27,* 4, 398–437.

Ciaccia, P. and Patella, M. 2009. Principles of Information Filtering in Metric Spaces. In *Proc. 2nd International Workshop on Similarity Search and Applications (SISAP'09)*. IEEE Computer Society, 99–106.

Ciaccia, P., Patella, M., and Zezula, P. 1997. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In *VLDB'97*. 426–435.

Corazza, P. 1999. Introduction to metric-preserving functions. *American Mathematical Monthly 104,* 4, 309–23.

Cormen, T., Stein, C., Rivest, R., and Leiserson, C. 2001. *Introduction to Algorithms, Second Edition*. The MIT Press.

Corral, A., Manolopoulos, Y., Theodoridis, Y., and Vassilakopoulos, M. 2000. Closest pair queries in spatial databases. *SIGMOD Rec. 29,* 2, 189–200.

Datta, R., Joshi, D., Li, J., and Wang, J. Z. 2008. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys 40,* 2, 1–60.

Dayhoff, M. O., Schwartz, R. M., and Orcutt, B. C. 1978. A model for evolutionary change in proteins. *Atlas of Protein Sequence and Structure 5*, 345–352.

Deb, S. 2004. *Multimedia Systems and Content-Based Image Retrieval*. Information Science Publ.

Dixon, S. L. and Koehler, R. T. 1999. The hidden component of size in two-dimensional fragment descriptors: Side effects on sampling in bioactive libraries. *Journal of Medicinal Chemistry 42,* 15, 2887–2900.

Dohnal, V., Gennaro, C., Savino, P., and Zezula, P. 2003. D-index: Distance searching index for metric data sets. *Multimedia Tools and Applications 21,* 1, 9–33.

Donahue, M., Geiger, D., Liu, T., and Hummel, R. 1996. Sparse representations for image decomposition with occlusions. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR'96)*. IEEE Computer Society, Washington, DC, USA, 7–12.

Dubuisson, M.-P. and Jain, A. 1994. A modified hausdorff distance for object matching. In *Proc. 12th IAPR Conference on Computer Vision and Image Processing*. 566–568.

Eckhardt, A., Skopal, T., and Vojtáš, P. 2009. On fuzzy vs. metric similarity search in complex databases. In *Proc. 8th Conference on Flexible Query Answering Systems (FQAS'09)*. LNAI, vol. 5822. Springer, 64–75.

Faloutsos, C. 1996. *Searching Multimedia Databases by Content*. Kluwer Academic Publishers, Norwell, MA, USA.

Faloutsos, C. and Kamel, I. 1994. Beyond uniformity and independence: analysis of r-trees using the concept of fractal dimension. In *Proc. 13th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'94)*. ACM Press, New York, NY, USA, 4–13.

Faloutsos, C. and Lin, K.-I. 1995. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proc. ACM International Conference on Management of Data (SIGMOD'95)*. ACM, New York, NY, USA, 163–174.

Faloutsos, C., Ranganathan, M., and Manolopoulos, Y. 1994. Fast subsequence matching in time-series databases. *SIGMOD Rec. 23,* 2, 419–429.

Farago, A., Linder, T., and Lugosi, G. 1993. Fast nearest-neighbor search in dissimilarity spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15,* 9, 957–962.

Foote, J. 1997. A similarity measure for automatic audio classification. In *Proc. AAAI Spring Symposium on Intelligent Integration and Use of Text, Image, Video, and Audio Corpora*.

Foote, J. and Silverman, H. 1994. A model distance measure for talker clustering and identification. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'94)*. IEEE, New York, NY, USA, 317–320.

Freeman, M. 2006. Evaluating dataflow and pipelined vector processing architectures for fpga co-processors. In *DSD '06: Proceedings of the 9th EUROMICRO Conference on Digital System Design*. IEEE Computer Society, Washington, DC, USA, 127–130.

Freeman, M., Weeks, M., and Austin, J. 2005. Hardware implementation of similarity functions. In *IADIS International Conference on Applied Computing*.

Fu, A. Y., Wenyin, L., and Deng, X. 2006. Detecting phishing web pages with visual similarity assessment based on earth mover's distance (emd). *IEEE Transactions on Dependable and Secure Computing 3,* 4, 301–311.

Gerstein, M. and Levitt, M. 1998. Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins. *Protein Science 7,* 2, 445–456.

Giannopoulos, P. and Veltkamp, R. C. 2002. A pseudo-metric for weighted point sets. In *Proc. 7th European Conference on Computer Vision-Part III (ECCV'02)*. Springer-Verlag, London, UK, 715–730.

Goh, K.-S., Li, B., and Chang, E. 2002. DynDex: A dynamic and non-metric space indexer. In *Proc. 10th ACM International Conference on Multimedia (MM'01)*. ACM Press, 466–475.

Goyal, N., Lifshits, Y., and Schtze, H. 2008. Disorder inequality: A combinatorial approach to nearest neighbor search. In *Proc. 1st ACM International Conference on Web Search and Data Mining (WSDM'08)*. ACM, 25–32.

Green, P. 1993. SWAT, http://www.phrap.org/phredphrap/swat.html.

Guo, A. and Siegelmann, H. 2004. Time-warped longest common subsequence algorithm for music retrieval. In *5th International Conference on Music Information Retrieval*.

Hart, P. 1968. The condensed nearest neighbour rule. *IEEE Transactions on Information Theory 14,* 3, 515–516.

He, H. and Singh, A. 2006. Closure-tree: An index structure for graph queries. In *Proc. 22nd International Conference on Data Engineering (ICDE'06)*. IEEE Computer Society, 38.

Henikoff, S. and Henikoff, J. 1992. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America 89,* 22, 10915–10919.

Higgins, A., Bahler, L., and Porter, J. 1993. Voice identification using nearest-neighbor distance measure. *IEEE International Conference on Acoustics, Speech, and Signal Processing 2*, 375–378.

Hjaltason, G. and Samet, H. 1995. Ranking in spatial databases. In *Proc. 4th International Symposium on Advances in Spatial Databases (SSD'95)*. LNCS 951. Springer, 83–95.

Holm, L. and Sander, C. 1993. Protein structure comparison by alignment of distance matrices. *Journal of Molecular Biology 233,* 1 (September), 123–138.

Howarth, P. and Ruger, S. 2005. Fractional distance measures for content-based image retrieval. In *Proc. 27th European Conference on Information Retrieval Research (ECIR'05)*. LNCS 3408. Springer-Verlag, 447–456.

Hristescu, G. and Farach-Colton, M. 1999. Cluster-preserving embedding of proteins. Tech. rep., 99–50, Department of Computer Science, Rutgers University.

Hu, N., Dannenberg, R., and Tzanetakis, G. 2003. Polyphonic audio matching and alignment for music retrieval. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. 185–188.

Huang, B. and Kinsner, W. 2002. ECG frame classification using dynamic time warping. In *Proc. Canadian Conference on Electrical and Computer Engineering*. Vol. 2. IEEE Computer Society, Los Alamitos, CA, USA, 1105–1110.

Huttenlocher, D., Klanderman, G., and Rucklidge, W. 1993. Comparing images using the hausdorff distance. *IEEE Pattern Analysis and Machine Intelligence 15,* 9, 850–863.

Jacobs, D., Weinshall, D., and Gdalyahu, Y. 2000. Classification with nonmetric distances: Image retrieval and class representation. *IEEE Pattern Analysis and Machine Intelligence 22,* 6, 583–600.

Jagadish, H. V., Ooi, B. C., Tan, K.-L., Yu, C., and Zhang, R. 2005. iDistance: An adaptive B+-tree based indexing method for nearest neighbor search. *ACM Trans. Database Syst. 30,* 2, 364–397.

Jäkela, F., Schölkopf, B., and Wichmanna, F. A. 2008. Similarity, kernels, and the triangle inequality. *Journal of Mathematical Psychology 52,* 5, 297–303.

Jesorsky, O., Kirchberg, K. J., and Frischholz, R. 2001. Robust face detection using the hausdorff distance. In *Proc. 3rd International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA'01)*. LNCS 2091. Springer-Verlag, 90–95.

Kabsch, W. 1976. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A 32,* 5 (September), 922–923.

Kao, D., Bergeron, R., and Sparr, T. 1997. Mapping metric data to multidimensional spaces. Tech. rep., TR 97-13, Department of Computer Science, University of New Hampshire.

Ke, Y., Cheng, J., and Ng, W. 2008. Efficient correlation search from graph databases. *IEEE Transactions on Data Knowledge and Engineering 20,* 12, 1601–1615.

Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. 2000. Dimensionality reduction for fast similarity search in large time series databases. *JOURNAL OF KNOWLEDGE AND INFORMATION SYSTEMS 3*, 263–286.

Keogh, E. and Ratanamahatana, C. 2005. Exact indexing of dynamic time warping. *Knowledge and Information Systems 7,* 3, 358–386.

Khamsi, M. A. and Kirk, W. A., Eds. 2001. *An Introduction to Metric Spaces and Fixed Point Theory.* Wiley-Interscience.

Kim, S.-W., Park, S., and Chu, W. 2001. An index-based approach for similarity search supporting time warping in large sequence databases. In *Proc. 17th International Conference on Data Engineering (ICDE'01)*. IEEE Computer Society, Washington, DC, USA, 607–614.

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

Kruskal, J. B. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29,* 1, 1–27.

Lackner, P., Koppensteiner, W. A., Sippl, M. J., and Domingues, F. S. 2000. Prosup: a refined tool for protein structure alignment. *Protein Engineering 13,* 11 (November), 745–752.

Lee, C.-H. and Lin, M.-F. 2008. Adaptive similarity measurement using relevance feedback. In *Proc. IEEE 8th International Conference on Computer and Information Technology Workshops (CITWORKSHOPS'08)*. IEEE Computer Society, Washington, DC, USA, 314–318.

Levenshtein, I. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics-Doklady 10*, 707–710.

Lifshits, Y. 2009. Combinatorial framework for similarity search. In *Proc. 2nd International Workshop on Similarity Search and Applications (SISAP'09)*. IEEE Computer Society, 11–17.

Lipman, D. and Pearson, W. 1985. Rapid and sensitive protein similarity searches. *Science 227*, 1435–1441.

Logan, B. and Salomon, A. 2001. A music similarity function based on signal analysis. In *Proc. IEEE International Conference on Multimedia and Expo (ICME'01)*. 745–748.

Lu, X. and Jain, A. 2008. Deformation modeling for robust 3d face matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30,* 8 (Aug.), 1346–1357.

Mandl, T. 1998. Learning similarity functions in information retrieval. In *EUFIT*.

Marcu, D. 2004. A study on metrics and statistical analysis. *Studia Univ. BABESBOLYAI, Mathematica XLIX,* 3, 43–74.

Marzal, A. and Vidal, E. 1993. Computation of normalized edit distance and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15,* 9, 926–932.

Micó, M. L., Oncina, J., and Vidal, E. 1994. A new version of the nearest-neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing time and memory requirements. *Pattern Recognition Letters 15,* 1, 9–17.

Morse, M. and Patel, J. 2007. An efficient and accurate method for evaluating time series similarity. In *PROC. ACM International Conference on Managment of Data (SIGMOD'07)*. ACM, 569–580.

Mukherjee, A. 1989. Hardware algorithms for determining similarity between two strings. *IEEE Trans. Comput. 38,* 4, 600–603.

Needleman, S. and Wunsch, C. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology 48,* 3, 443–453.

Nikolova, N. and Jaworska, J. 2003. Approaches to measure chemical similarity: a review. *SAR & Combinatorial Science 22,* 10, 1006–1026.

Ortiz, A. R., Strauss, C. E., and Olmea, O. 2002. Mammoth (matching molecular models obtained from theory): An automated method for model comparison. *Protein Science 11,* 11 (November), 2606–2621.

Pampalk, E. 2006. Audio-based music similarity and retrieval: Combining a spectral similarity model with information extracted from fluctuation patterns. implementation submitted to the 3rd Annual Music Information Retrieval eXchange (MIREX'06). In *Proc. International Symposium on Music Information Retrieval*.

Papadimitriou, C. H., Tamaki, H., Raghavan, P., and Vempala, S. 1998. Latent semantic indexing: A probabilistic analysis. In *Proc. ACM SIGACT-SIGMOD-SIGART Conference on Principles of Database Systems (PODS'98)*. ACM, New York, NY, USA, 159–168.

Parker, C., Fern, A., and Tadepalli, P. 2007. Learning for efficient retrieval of structured data with noisy queries. In *Proc. 24th International Conference on Machine Learning (ICML'07)*. ACM Press, 729–736.

Perera, D. G. and Li, K. F. 2008. Parallel computation of similarity measures using an fpga-based processor array. In *AINA '08: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications*. IEEE Computer Society, Washington, DC, USA, 955–962.

Pu, J., Kalyanaraman, Y., Jayanti, S., Ramani, K., and Pizlo, Z. 2007. Navigation and discovery in 3D cad repositories. *IEEE Computer Graphics and Applications 27,* 4, 38–47.

Ratanamahatana, C. A. and Tohlong, P. 2006. Speech audio retrieval using voice query. In *Proc. 9th International Conference on Asian Digital Libraries (ICADL'06)*. LNCS 4312. Springer-Verlag, 494–497.

Robinson, D. D., Lyne, P. D., and Richards, W. G. 2000. Partial molecular alignment via local structure analysis. *Journal of Chemical Information and Computer Sciences 40,* 2, 503–512.

Rosch, E. 1975. Cognitive reference points. *Cognitive Psychology 7*, 532–547.

Roth, V., Laub, J., Buhmann, J. M., and Müller, K. R. 2002. Going metric: Denoising pairwise data. In *Proc. International Conference on Neural Information Processing Systems (NIPS'02)*. 817–824.

Rothkopf, E. 1957. A measure of stimulus similarity and errors in some paired-associate learning tasks. *Journal of Experimental Psychology 53,* 2, 94–101.

Rubner, Y., Puzicha, J., Tomasi, C., and Buhmann, J. 2001. Emipirical evaluation of dissimilarity measures for color and texture. *Computer Vision and Image Understanding 84,* 1, 25–43.

Rubner, Y. and Tomasi, C. 2001. *Perceptual Metrics for Image Database Navigation*. Kluwer.

Rubner, Y., Tomasi, C., and Guibas, L. 1998. A metric for distributions with applications to image databases. In *Proc. 6th International Conference on Computer Vision (ICCV'98)*. 59–66.

Saha, S. and Bandyopadhyay, S. 2007. MRI brain image segmentation by fuzzy symmetry based genetic clustering technique. In *Proc. IEEE Congress on Evolutionary Computation (CEC'07)*. IEEE, 4417–4424.

Samet, H. 2006. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann.

Santini, S. and Jain, R. 1999. Similarity measures. *IEEE Pattern Analysis and Machine Intelligence 21,* 9, 871–883.

Sanz, I., Mesiti, M., Guerrini, G., and Berlanga, R. 2008. Fragment-based approximate retrieval in highly heterogeneous xml collections. *Data & Knowledge Engineering 64,* 1, 266–293.

Schaefer, G., Zhu, S., and Ruszala, S. 2005. Visualization of medical infrared image databases. In *Proc. 27th IEEE Annual Conference on Engineering in Medicine and Biology*. IEEE, 634–637.

Shepard, R. N. 1962. The analysis of proximities: Multidimensional scaling with an unknown distance function I. *Psychometrika 27,* 2, 125–140.

Skopal, T. 2004. Pivoting M-tree: A Metric Access Method for Efficient Similarity Search. In *Proceedings of the 4th annual workshop DATESO, Desná, Czech Republic, ISBN 80-248-0457-3, also available at CEUR, Volume 98, ISSN 1613-0073,* http://www.ceur-ws.org/Vol-98. 21–31.

Skopal, T. 2006. On fast non-metric similarity search by metric access methods. In *Proc. 10th International Conference on Extending Database Technology (EDBT'06)*. LNCS 3896. Springer, 718–736.

Skopal, T. 2007. Unified framework for fast exact and approximate search in dissimilarity spaces. *ACM Transactions on Database Systems 32,* 4, 1–46.

Skopal, T. and Bustos, B. 2009. On index-free similarity search in metric spaces. In *Proc. 20th International Conference on Database and Expert System Applications (DEXA'09)*. LNCS, vol. 5690. Springer, 516–531.

Skopal, T. and Lokoč, J. 2008. NM-tree: Flexible approximate similarity search in metric and non-metric spaces. In *Proc. 19th International Conference on Database and Expert Systems Applications (DEXA'08)*. LNCS 5181. Springer-Verlag, 312–325.

Smeaton, A. F., Over, P., and Kraaij, W. 2006. Evaluation campaigns and TRECVid. In *Proc. 8th ACM International Workshop on Multimedia Information Retrieval (MIR'06)*. ACM Press, New York, NY, USA, 321–330.

Smith, T. and Waterman, M. 1981. Identification of common molecular subsequences. *Journal of Molecular Biology 147,* 1, 195–197.

Su, M.-C. and Chou, C.-H. 2001. A modified version of the k-means algorithm with a distance based on cluster symmetry. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23,* 6, 674–680.

Suyoto, I. S. H., Uitdenbogerd, A. L., and Scholer, F. 2007. Effective retrieval of polyphonic audio with polyphonic symbolic queries. In *Proc. International Workshop on Multimedia Information Retrieval (MIR'07)*. ACM, New York, NY, USA, 105–114.

Tao, Y., Papadias, D., and Lian, X. 2004. Reverse knn search in arbitrary dimensionality. In *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*. VLDB Endowment, 744–755.

Taylor, W. and Orengo, C. 1989. Protein structure alignment. *Journal of Molecular Biology 208,* 1, 1–22.

Tekli, J., Chbeir, R., and Yetongnon, K. 2007. A hybrid approach for xml similarity. In *Proc. 33rd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'07)*. LNCS, vol. 4362. Springer, 783–795.

Tsang, W., Corboy, A., Lee, K., Raicu, D., and Furst, J. 2005. Texture-based image retrieval for computerized tomography databases. In *Proc. 18th IEEE Symposium on Computer-based Medical Systems (CBMS'05)*. IEEE Computer Society, 593–598.

Tsukada, S. and Watanabe, T. 1995. Speech recognition device for calculating a corrected similarity partially dependent on circumstances of production of input patterns, NEC Corporation. In *US Patent No. 5432886*.

Tuzcu, V. and Nas, S. 2005. Dynamic time warping as a novel tool in pattern recognition of ECG changes in heart rhythm disturbances. In *IEEE International Conference on Systems, Man and Cybernetics*. IEEE Computer Society, 182–186.

Tversky, A. 1977. Features of similarity. *Psychological Review 84,* 4, 327–352.

Tversky, A. and Gati, I. 1982. Similarity, separability, and the triangle inequality. *Psychological Review 89,* 2, 123–154.

Typke, R., Giannopoulos, P., Veltkamp, R., Wiering, F., and van Oostrum, R. 2003. Using transportation distances for measuring melodic similarity. In *Proc. 4th International Conference on Music Information Retrieval (ISMIR'03)*. 107–114.

Uhlmann, J. 1991. Satisfying general proximity/similarity queries with metric trees. *Information Processing Letters 40,* 4, 175–179.

Vacha, P. and Haindl, M. 2008. Illumination invariants based on Markov random fields. In *Proc. 19th International Conference on Pattern Recognition (ICPR'08)*. IEEE Computer Society.

Vidal, E. 1986. An algorithm for finding nearest neighbours in (approximately) constant average time. *Pattern Recognition Letters 4,* 3, 145–157.

Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., and Keogh, E. 2003. Indexing multidimensional time-series with support for multiple distance measures. In *Proc. 9th ACM International Conference on Knowledge Discovery and Data Mining (KDD'03)*. ACM Press, New York, NY, USA, 216–225.

Vlachos, M., Kollios, G., and Gunopulos, D. 2005. Elastic translation invariant matching of trajectories. *Machine Learning 58,* 2–3, 301–334.

Vojtáš, P. and Eckhardt, A. 2009. Using tuneable fuzzy similarity in non-metric search. In *Proc. 2nd International Workshop on Similarity Search and Applications (SISAP'09)*. IEEE, 163–164.

Wang, X., Wang, J. T. L., Lin, K. I., Shasha, D., Shapiro, B. A., and Zhang, K. 2000. An index structure for data mining and clustering. *Knowledge Information Systems 2,* 2, 161–184.

Weber, R., Schek, H.-J., and Blott, S. 1998. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB '98: Proceedings of the 24rd International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 194–205.

Wild, D. J. and Willett, P. 1996. Similarity searching in files of three-dimensional chemical structures. alignment of molecular electrostatic potential fields with a genetic algorithm. *Journal of Chemical Information and Computer Sciences 36,* 2, 159–167.

Willett, P. 1998. Structural similarity measures for database searching. In *Encyclopedia of Computational Chemistry*. John Wiley, 2748–2756.

Willett, P., Barnard, J. M., and Downs, G. M. 1998. Chemical similarity searching. *Journal of Chemical Information and Computer Sciences 38,* 6, 983–996.

Wilson, D. L. 1972. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics 2,* 3, 408–421.

Wirotius, M., Ramel, J.-Y., and Vincent, N. 2004. Improving DTW for online handwritten signature verification. In *Proc. International Conference on Image Analysis and Recognition (ICIAR'04)*. LNCS 3212. Springer-Verlag, 786–793.

Yan, X., Yu, P., and Han, J. 2005. Substructure similarity search in graph databases. In *Proc. ACM International Conference on Management of Data (SIGMOD'05)*. ACM, 766–777.

Yang, K. and Shahabi, C. 2004. A PCA-based similarity measure for multivariate time series. In *Proc. 2nd ACM International Workshop on Multimedia Databases (MMDB'04)*. ACM, New York, NY, USA, 65–74.

Yi, B.-K., Jagadish, H. V., and Faloutsos, C. 1998. Efficient retrieval of similar time sequences under time warping. In *Proc. International Conference on Data Engineering (ICDE'98)*. 201–208.

Zezula, P., Amato, G., Dohnal, V., and Batko, M. 2005. *Similarity Search: The Metric Space Approach (Advances in Database Systems)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Zhang, B. and Srihari, S. 2002. A fast algorithm for finding k-nearest neighbors with non-metric dissimilarity. In *Proc. 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*. IEEE Computer Society, 13.

Zhao, Q., Hoi, S. C. H., Liu, T.-Y., Bhowmick, S. S., Lyu, M. R., and Ma, W.-Y. 2006. Time-dependent semantic similarity measure of queries using historical click-through data. In *Proc. 15th International Conference on World Wide Web (WWW'06)*. ACM, New York, NY, USA, 543–552.

Zhou, X., Zhou, X., and Shen, H. 2007. Efficient similarity search by summarization in large video databse. In *Proc. 18th Australasian Database Conference (ADC'07)*. Australian Computer Society, 161–167.

Zuo, X. and Jin, X. 2007. General hierarchical model (ghm) to measure similarity of time series. *SIGMOD Record 36,* 1, 13–18.