Metric Databases

Edgar Chávez
Universidad Michoacana, Mexico

Gonzalo Navarro
University of Chile, Chile

INTRODUCTION

The advent of the digital age creates an increasing interest to search for information in large, unstructured repositories containing textual and multimedia data. Retrieval from those repositories requires more powerful query languages, which exceed the querying capabilities of traditional database technologies. The metric space model, described herein, is an extension of the exact searching paradigm aiming to cope with the new requirements.

Traditional databases are built around the concept of exact searching. The database is divided into records, each record having a fully comparable key. Queries to the database return all the records whose keys match the search key exactly. More sophisticated searches, such as range queries on numerical keys or prefix searching on alphabetical keys, still rely on the concept of exact comparison. Recently, database manager systems (DBMS) incorporate the ability of storing so-called multimedia objects, which nevertheless can rarely be searched by content.

The most distinguishing feature of multimedia objects is that there is no point in comparing them by exact equality. Rather, users are interested in the similarity between two objects. The metric space model gives a theoretical foundation to define a meaningful way to retrieve multimedia data. A metric database is a collection of digital objects (of any kind) with a perceived similarity and a formal way to compute this perceived similarity as a metric. The perceived similarity is provided by everyday experience or by experts.

Theoretical foundations for metric databases are well established. However, metric databases are not yet mature. The end users do not have the equivalent of a full DBMS for multimedia indexing. There is, however, at least one rather robust, open source implementation of an index for metric spaces (Ciaccia, Patella, & Zezula, 1997).

APPLICATIONS OF METRIC DATABASES

Although retrieving multimedia data is one of the main applications for metric databases, their applications extend well beyond it. The following sections describe several of those applications (Chávez, Navarro, Baeza-Yates, & Marroquin, 2001).

Query by Content in Multimedia Objects

New data types such as images, fingerprints, audio and video (i.e., multimedia data types) cannot be meaningfully queried in the classical sense. Not only cannot they be ordered, but they cannot even be compared by equality. No application will be interested in searching an audio segment exactly equal to a given one. The probability that two different images are pixel-wise equal is negligible unless they are digital copies of the same source. In multimedia applications, all the queries ask for objects similar to a given one. Some example applications are face recognition, fingerprint matching, voice recognition, and image retrieval.

These approaches are based on the definition of a similarity function among objects. Those functions are provided by an expert, but they pose no assumptions on the type of queries that can be answered. In many cases, the distance is obtained via a set of $k$ features that are extracted from the object (e.g., in an image, a useful feature is the color histogram). Then each object is represented as its $k$ features, that is, a point in a $k$-dimensional space.

Information Retrieval

Although not considered a multimedia data type, unstructured text retrieval poses problems similar to mul-
timedia retrieval. This is because textual documents are in general not structured to easily provide the desired information. Text documents may be searched for strings that are present or not, but in many cases they are searched for semantic concepts.

The problem is solved by retrieving documents similar to a given query. The user can even present a document as a query, so that the system finds similar documents. Some similarity approaches are based on mapping a document to a vector of integer values, so that each dimension is a vocabulary word and the number of times the word appears in the document is the coordinate of the document in that dimension. Similarity functions are then defined in that space. Notice, however, that the dimensionality of the space is very high (thousands of dimensions).

**Text Retrieval**

Another problem related to text retrieval is spelling. Because huge text databases with low quality control are emerging (e.g., the Web), and typing, spelling, or OCR (optical character recognition) errors are commonplace in the text and the query, documents that contain a misspelled word are no longer retrievable by a correctly written query. Models of similarity among words exist (variants of the “edit distance”) that capture those kinds of errors. In this case, one gives a word and wants to retrieve the words close to it.

**Computational Biology**

ADN and protein sequences are the basic objects of study in molecular biology. Because they can be modeled as text, there is the problem of finding a given sequence of characters inside a longer sequence. However, an exact match is unlikely, and computational biologists are more interested in finding parts of a longer sequence that are similar to a given short sequence. That the search is not exact is because of minor differences in the genetic streams that describe sequences of similar functionality, evolution, experimental errors, and so on. The measure of similarity used is related to the type of differences one is interested in.

**Pattern Recognition and Function Approximation**

A simplified definition of pattern recognition is the construction of a function approximator. One has a finite sample of the data, and each data sample is labeled as belonging to a certain class. When a fresh data sample is provided, one must label this new sample as belonging to one of the known classes.

If the objects are $m$-dimensional vectors of real numbers, then a natural choice is neural nets and fuzzy function approximators. There is also another popular universal function approximator: the $k$ nearest neighbor classifier. It consists of finding the $k$ nearest objects to the unlabeled sample and assigning to this object the label having majority among the $k$ nearest neighbors. Opposed to neural nets and fuzzy classifiers, the $k$ nearest neighbor rule has zero training time, but if no indexing algorithm is used, it has linear complexity. Other applications of this universal function approximator are density estimation and reinforcement learning. In general, in any application in which one wants to infer a function based on a finite set of samples, there is a potential application.

**Audio and Video Compression**

Audio and video transmission over a narrow-band channel is a serious problem in, for example, Internet-based audio and video conferencing. A frame (i.e., a static picture in a video, or a fragment of the audio) can be thought of as formed by a number of (possibly overlapped) subframes (e.g., 16×16 subimages in a video). In a very succinct description, the problem can be solved by sending the first frame as is and, for the next frames, sending only the subframes having a large difference from the previously sent subframes. This description encompasses the MPEG standard. This poses the need to efficiently find subframes similar to the one that is to be sent among a repository of recently sent subframes.

**THE METRIC SPACE MODEL**

Proximity queries are those extensions of the exact searching from which one wants to retrieve objects from a database that are close to a given query object (Chávez, Navarro, Baeza-Yates, & Marroquin, 2001). The query object is not necessarily a database element. The concept can be formalized using the metric space model, in which a distance function $d(x, y)$ is defined for every site in a set $X$. The distance function $d$ has metric properties, that is, it satisfies $d(x, y) \geq 0$ (positiveness), $d(x, y) = d(y, x)$ (symmetry), $d(x, y) = 0$ iff $x = y$ (strict positiveness), and $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality). The latter is the key property permitting solutions better than brute force.

The database is a set $U \subseteq X$, and the query object, $q$, is an arbitrary element of $X$. A similarity query can be of two basic types: A metric range query $(q, r)d = \{ u \in U :
Related Content

A Database Interface for Link Analysis  
[www.igi-global.com/article/database-interface-link-analysis/3327?camid=4v1a](www.igi-global.com/article/database-interface-link-analysis/3327?camid=4v1a)

A Meta-Analysis Comparing Relational and Semantic Models  
[www.igi-global.com/chapter/meta-analysis-comparing-relational-semantic/74401?camid=4v1a](www.igi-global.com/chapter/meta-analysis-comparing-relational-semantic/74401?camid=4v1a)

A Formal Verification and Approach for Real-Time Databases  
[www.igi-global.com/chapter/formal-verification-approach-real-time/28557?camid=4v1a](www.igi-global.com/chapter/formal-verification-approach-real-time/28557?camid=4v1a)

Querying Multidimensional Data  
[www.igi-global.com/chapter/querying-multidimensional-data/26971?camid=4v1a](www.igi-global.com/chapter/querying-multidimensional-data/26971?camid=4v1a)