CHAPTER-25
Mining Multimedia Databases

25.Mining Multimedia Databases

25.1 Introduction

A multimedia database system stores and manages a large collection of multimedia objects, such as audio data, image data, video data, sequence data, and hypertext data, which contain text, text markups and linkages. Multimedia database systems are increasingly common owing to the popular use of audio-video equipment, CD-ROMs and the Internet. Typical multimedia database systems include NASA's EOS (Earth Observation System), various kinds of image and audio-video databases, human genome databases, and Internet databases.

In this section, our study of multimedia data mining focuses on image data mining. Mining sequence data is studied with respect to data mining applications in bioinformatics. Mining hypertext data is studied on mining the World-Wide Web. Here we introduce multi-media data mining methods, including similarity search in Multimedia data, multidimensional analysis, classification and prediction analysis, and mining associations in multimedia data.

25.2 Similarity Search in Multimedia Data

"When searching/or similarities in multimedia data, I suppose we can search based on either the data description or the data content?" That is correct. For similarity searching in multimedia data, we consider two main families of multimedia indexing and retrieval systems:

(1) description-based retrieval systems, which build indices and perform object retrieval based on image descriptions, such as keywords, captions, size, and time of creation; and

(2) Content-based retrieval systems, which support retrieval based on the image content, such as color histogram, texture, shape, objects, and wavelet transforms. Description-based retrieval is labor-intensive if performed manually. If automated, the results are typically of poor quality; for example, the assignment of keywords to images can be a tricky and arbitrary task. Content-based retrieval uses: visual features to index images and promotes object retrieval based on feature similarity, which is highly desirable in many applications.

In a content-based retrieval system, there are often two kinds of queries: image-sample-based queries and image-feature specification queries. Image sample-based queries find all of the images that are similar to the given image sample. This search compares the feature vector (or signature) extracted from the sample with the feature vectors of images that have already been extracted and indexed in the image database. Based on this comparison, images that are close to the sample image are returned. Image feature specification queries specify or sketch image features like color, texture, or shape, which are translated into a feature vector, to be matched with the feature vectors of the images in the database. Content-based retrieval has wide applications, including medical diagnosis, weather prediction, TV production, Web search engines for images, and e-commerce. Some systems, such as QBIC (Query By Image Content),

support both sample-based and image feature specification queries. There are also systems that support both content- based and description-based retrieval.

Several approaches have been proposed and studied for similarity-based retrieval in image databases, based on image signature:

- Color histogram-based signature: In-this approach, the signature of an image includes color histograms based on the color composition of an image regardless of its scale or orientation. Since this method does not contain any information about shape, location, or texture, two images with similar color composition may contain very different shapes or textures, and thus could be completely unrelated in semantics.
- Multifeature composed signature: In this approach, the signature of an image includes a composition of multiple features: color histogram, shape, location, and texture. Often, separate distance functions can be defined for each feature and subsequently combined to derive the overall results. Multiple dimensional content-based searches often use one or a few probe features to search for images containing such (similar) features. It can therefore be used to search for similar images.
- Wavelet-based signature: This approach uses the dominant wavelet coefficients of an image as its signature. Wavelets capture shape, texture, and location information in a single unified framework. This improves efficiency and reduces the need for providing multiple search primitives (unlike the second method above). However, since this method computes a single, signature for an entire image, it may fail to identify images containing similar objects where the objects differ in location or size.
- Wavelet-based signature with region-based granularity: In this approach, the computation and comparison of signatures are at the granularity of regions, not the entire image. This is based on the observation that similar images may contain similar regions, but a region in one image could be a translation or scaling of a matching region in the other. Therefore, a similarity measure between the query image Q and a target images T can be defined in terms of the fraction of the area of the two images covered by matching pairs of regions from Q and T. Such a region-based similarity search can find images containing similar objects, but where these objects may be translated or scaled.

25.3 Multidimensional Analysis of Multimedia Data

To facilitate the multidimensional analysis of large multimedia databases, multimedia data cubes can be designed and constructed in a manner similar to that for traditional data cubes from relational data. A multimedia data cube can contain additional-dimensions and measures for multimedia information, such as color, texture, and shape.

Let's examine a multimedia data mining system prototype called MultiMediaMiner, which extends the DBMiner system by handling multimedia data. The example database tested in the MultiMediaMiner system is constructed as follows: Each image contains two descriptors: a feature descriptor and a layout descriptor. The original image is not stored directly in the database; only its descriptors are stored. The description information encompasses fields like image file name, image URL, image type (e.g., gif, jpeg, bmp, avi, mpeg, etc.), a list of all known Web pages referring to the image (i.e., parent URLs), a list of keywords, and a thumbnail used by the user interface for image and video browsing. The feature

descriptor is a set of vectors for each visual characteristic. The main vectors are a color vector containing the color histogram quantized to 512 colors (8 x 8 x 8 for R x G x B), a MFC (Most Frequent Color) vector, and a MFO (Most Frequent Orientation) vector. The MFC and MFO contain five color centroids and five edge orientation centroids for the five most frequent colors and five most frequent orientations, respectively- The edge orientations used are 0°, 22.5°, 45°, 67.5°, 90°, and so on. The layout descriptor contains a color layout vector and an edge layout vector. Regardless of their original size, all images are assigned an 8 X 8 grid. The most frequent color for each of the 64 cells is stored in the color layout vector, and the number of edges for each orientation in each of the cells is stored in the edge layout vector. Other sizes of grids, like 4 x 4, 2 x 2 and 1x1, can easily be derived.

The Image Excavator component of MultiMediaMiner uses image contextual information, like HTML tags in Web pages, to derive keywords. By traversing on-line directory structures, like the Yahoo! directory, it is possible to create hierarchies of keywords mapped onto the directories in which the image was found. These graphs are used as concept hierarchies for the dimension keyword in the multimedia data cube.

A multimedia data cube can have many dimensions. The following are some examples: the size of the image or video in bytes; the width and height of the frames (or picture), constituting two dimensions; the date on which the image or video was created (or last modified); the format type of the image or video; the frame sequence duration in seconds; the image or video Internet domain; the Internet domain o f pages referencing the image or video (parent URL); the keywords; a color dimension; an edge-orientation dimension; and so on. Concept hierarchies for many numerical dimensions maybe automatically defined. For other dimensions, such as for Internet domains or color, predefined hierarchies may be used.

The construction of a multimedia data cube will facilitate multiple dimensional analyses of multimedia data primarily based on visual content, and the mining of multiple kinds of knowledge, including summarization, comparison, classification, association, and clustering.

The multimedia data cube seems to be an interesting model for multidimensional analysis of multimedia data. However, we should note that it is difficult to implement a data cube efficiently given a large number of dimensions. This curse of dimensionality is especially serious in the case of multimedia data cubes. We may like to model color, orientation, texture, keywords, and so on, as multiple dimensions in a multimedia data cube. However, many of these attributes are set-oriented instead of single-valued. For example, one image may correspond to a set of keywords. It may contain a set of objects, each associated with a set of colors. If we use each keyword as a dimension or each detailed color as a dimension in the design of the data cube, it will create a huge number of dimensions. On the other hand, not doing so may lead to the modeling of an image at a rather rough, limited, and imprecise scale. More research is needed on how to design a multimedia data cube that may strike a balance between efficiency and the power of representation.

25.4 Classification and Prediction Analysis of Multimedia Data

Classification and predictive modeling have been used for mining multimedia data, especially in scientific research, such as astronomy, seismology, and geo-scientific research. Decision tree classification is an essential data mining method in reported image data mining applications.

Example Taking sky images that have been carefully classified by astronomers as the training set, we can construct models for the recognition of galaxies, stars, and other stellar objects, based on properties like magnitudes, areas, intensity, image moments, and orientation. A large number of sky images taken by telescopes or space probes can then be tested against the constructed models in order to identify new celestial bodies. Similar studies have successfully been performed to identify volcanoes on Venus.

Data preprocessing is important when mining such image data and can include data cleaning, data focusing, and feature extraction. Aside from standard methods used in pattern recognition such as edge detection and Hough transformations, techniques can be explored such as the decomposition of images to eigenvectors or the adoption of probabilistic models to deal with uncertainty. Since the image data are often in huge volumes and may require substantial processing power, parallel and distributed processing are useful. Image data mining classification and clustering are closely linked to image analysis and scientific data mining, and thus many image analysis techniques and scientific data analysis methods can be applied to image data mining.

25.5 Mining Associations in Multimedia Data

Association rules involving multimedia objects can be mined in image and video databases- At least three categories can be observed: - .

Associations between image content and non-image content features: A rule like “if at least 50%of the upper part of the picture is blue, it is likely to represent sky" belongs to this category since it links the image content to the keyword sky.

Associations among image contents that are not related to spatial relationships: A rule like "a picture contains two blue squares, it is likely to contain one red circle as well" belongs to this category since the associations are all regarding image contents.

Associations among image contents related to spatial relationships: A rule like "a red triangle is in between two yellow squares, it is likely there is a big oval-shaped object underneath" belongs to this category since it associates objects in the image with spatial relationships.

To mine associations among multimedia objects, we can treat each image as a transaction and find frequently occurring patterns among different images.

First, an image may contain multiple objects, each with many features such as color, shape, texture, keyword, and spatial location, so that there could be a large number of possible associations. In many cases, a feature may be considered as the same in, two images at a certain level of resolution, but different at a finer resolution level. Therefore, it is essential to promote a progressive resolution refinement approach. That is, we can first mine frequently occurring patterns at a relatively rough resolution level, and then focus only on those that have passed the minimum support threshold when mining at a finer resolution level. This is because the patterns that are not frequent at a rough level cannot be frequent at

finer resolution levels. Such a multi-resolution mining strategy substantially reduces the overall data mining cost without loss of the quality and completeness of data mining results. This leads to an efficient methodology for mining frequent itemsets and associations in large multimedia databases.

Second, since a picture containing multiple recurrent objects is an important feature in image analysis, recurrence of the same objects should not be ignored in association analysis. For example, a picture containing two golden circles is treated quite differently from that containing only one. This is quite different from that in a transaction database, where the fact that a person buys one gallon of milk or two may often be treated the same as "buys_milk". Therefore, the definition of multimedia association and its measurements, such as support and confidence, should be adjusted accordingly.

Third, there often exist important relative spatial relationships among multimedia objects, such as above, beneath, between, nearby, left-of, and so on. These features are very useful for exploring object associations and correlations. Spatial relationships together with other content-based multimedia features, such as color, shape, texture, and keywords, may form interesting associations. Thus, spatial data mining methods and properties of topological spatial relationships become quite important for multimedia mining.

25.6 Mining Time-Series and Sequence Data

"What is a time-series database? What is a sequence database?" A time-series database consists of sequences of values or events changing with time. The values are typically measured at equal time intervals. Time-series databases are popular in many applications, such as studying daily fluctuations of a stock market, traces of a dynamic production process, scientific experiments, medical treatments, and so on. A time-series database is also a sequence database. However, a sequence database is any database consists of sequences of ordered events, with or without concrete notions of time. For example, Web page traversal sequences are sequence data, but may not be time-series data.

In this section, we examine several important aspects of mining time-series databases and sequence databases, including trend analysis, similarity search, and the mining of sequential patterns and periodic patterns in time-related data.

Trend Analysis

A time series involving a variable V, representing, say, the daily closing price of a share in a stock market, can be viewed as a function of time f, that is, Y=-F(t), Such a function can be illustrated as a time-series graph, which describes a point moving with the passage of time.

"How can we study time-series data?" There are four major components or movements that are used to characterize time-series data:

Long-term or trend movements: These indicate the general direction in which a time-series graph is moving over a long interval of time. A trend curve or a trend line displays this movement. Typical methods for determining a trend curve or trend line include the weighted moving average method and the least squares method, discussed further below.

Cyclic movements or cyclic variations: These refer to the cycles, that is, the long-term oscillations about a trend line or curve, which may or may not be periodic: That is, the cycles need not necessarily follow exactly, similar patterns after equal intervals of time.

Seasonal movements or seasonal variations: These movements are due to events that recur annually, such as the sudden increase in sales of chocolates and flowers before Valentine's Day or of department store items before Christmas. In other words, seasonal movements are the identical or nearly identical patterns that a time series appears to follow during corresponding months of successive years. -

Irregular or random movements: These characterize the sporadic motion of time series due to random or chance events, such as labor disputes, floods, or announced personnel changes within companies.

The above trend, cyclic, seasonal and irregular movements are represented by variables **T, C, S, I**, respectively. Time-series analysis is also referred to as the decomposition of a time series into these four basic movements. The time- series variable Y can be modeled as either the product of the four variables **(i.e., Y =T x C x S x I)** or their sum. This choice is typically empirical.

A moving average loses the data at the beginning and ends of a series may sometimes generate cycles or other movements that are not present in the Original data, and may be strongly affected by the presence of extreme values.

By using a weighted moving average of appropriate orders, the cyclic, seasonal, and irregular patterns in the data can be eliminated, resulting in only the trend movement.

“Is there any way to adjust the data for seasonal fluctuations?" In many business transactions, there are expected regular seasonal fluctuations, such as higher sales volumes during the Christmas season. Therefore, it is important to identify such seasonal variations and "deseasonalize" the data for trend and cyclic data analysis. For this purpose, the concept of seasonal index is introduced, as a set of numbers showing the relative values of variable during the months of a year. For example, if the sales during October, November, and December are 80%, 120%, and 140% of the average monthly sales for the whole year, respectively, then 80, 120, and 140 are' the seasonal index numbers for the year. If the original monthly data are divided by the corresponding seasonal index numbers, the resulting data are said to be deseasonalized, or adjusted for seasonal variations. Such data still include trend, cyclic, and irregular movements.

The deseasonalized data can be adjusted for trend by dividing the data by their corresponding trend values- Furthermore; an appropriate moving average will smooth out the irregular variations and leave only cyclic variations for further analysis. If periodicity or approximate periodicity of cycles occurs, cyclic; indexes can be constructed in a manner similar to that for seasonal indexes.

Finally, irregular or random movements can be estimated by adjusting data for the trend, seasonal, and cyclic variations. In general, small deviations tend to occur with large frequency, whereas large deviations tend to occur with small frequency, following normal distribution.

In practice, it is often beneficial to first graph the time series and qualitatively estimates the presence of long-term trends, seasonal variations, and cyclic Variation. This may help in selecting a suitable method for analysis and in comprehending its results.

With the systematic analysis of the movements of trend, cyclic, seasonal and irregular components, it is possible to make long-term or short-term predictions (forecasting the time series) with reasonable quality.

25.7 Similarity Search in Time-Series Analysis

"What is a similarity search?" Unlike normal database queries, which find data that match the given query exactly, a similarity search finds data sequences that differ only slightly from the given query sequence. Given a set of time-series sequences, there are two types of similarity search. Subsequence matching find all of the data sequences that are similar to the given sequence, while whole sequence matching finds those sequences that are similar to one other. Similarity search in time-series analysis is useful for the analysis of financial markets (e.g., stock data analysis), medical diagnosis (e.g., cardiogram analysis), and in scientific or engineering databases (e.g., power consumption analysis).

Data Transformation: From Time Domain to Frequency Domain

For similarity analysis of time-series data, Euclidean distance is typically used as a similarity measure.

Many techniques for signal analysis require the data to be in the frequency domain. Therefore, distance-preserving orthonormal transformations are often used to transform the data from the time domain to the frequency domain. Usually, data-independent transformation is applied where the transformation matrix is determined a priori, independent of the in-put data. Two popular data-independent transformations are the discrete Fourier transform (DFT) and the discrete wavelet transform (DWT). Since the distance between two signals in the time domain is the same as their Euclidean distance in the frequency domain, the DFT does a good job of preserving essentials in the first few coefficients. By keeping only the first few (i.e., "strongest" coefficients of the DFT) we can compute the lower bounds of the actual distance.

For efficient accessing, a multidimensional index can be constructed using the first few Fourier coefficients. When a similarity query is submitted to the system, the index can he used to retrieve the sequences that are almost a certain small distance away from the query sequence. Post-processing is then performed by computing the actual distance between sequences in tie time domain and discarding any false matches.

For sub sequence matching, each sequence is first broken down into a set of "pieces" of window with length w. The features of the subsequence inside each window are then extracted. Each sequence is mapped to a "trail" in the feature space. For subsequence analysis, we divide the trail of each sequence into "subtrails", each represented by a minimum bounding rectangle. A multipiece assembly algorithm can then be used to search for longer sequence matches.

25.8 Cases and Parameters for Sequential Pattern Mining

Most studies of sequential pattern mining concentrate on symbolic patterns, since numerical curve patterns usually belong to the scope of trend analysis and prediction in statistical time-series analysis.

There are several parameters whose setting may strongly influence the results of sequential pattern mining. The first parameter is the duration of a time sequence T. The duration may be the entire available sequence in the database, or a user- selected subsequence such as that corresponding to the year 1999- Sequential pattern mining, can then be confined to the data within a specified duration. Duration may also be defined as sets of partitioned sequences, such as every year, or every week after stock crashes, or every two weeks before and after a volcano eruption. In such cases, periodic patterns can be discovered.

The second parameter is the event folding window, w. A set of events occurring within a specified period of time can be viewed as occurring together in certain analyses. If w is set to be as long as the duration T, it finds time-insensitive frequent patterns-these are essentially association patterns, such as "In 1999, customers who bought a -PC bought a digital camera as well" (not even caring which was bought first). If w is set to 0 (i.e., no event sequence folding), sequential patterns are found where each event occurs at a distinct time instant, such as "A customer who bought a PC and then a memory chip is likely to buy a CD-ROM later on". If w is set to be something in between (e.g., for transactions occurring within the same month or within a slide window of 24 hours), then these transactions are considered as occurring within the same period, and such sequences are folded in the analysis.

The third parameter is the time interval, int, between events in the discovered pattern. This parameter may have the following settings:

- int = 0: This means that no interval gap is allowed; that is, it finds strictly consecutive sequences, such as sequential patterns like a_{i-1} , a_i , a_{i+1} where a, is an event occurring at time i. The event folding window, w, can be taken into consideration in such a case. For example, if the event folding window is set to a week, this will find frequent patterns occurring in consecutive weeks. DNA analysis often requires the discovery of consecutive sequences without any interval gap.
- min_interval = int = max_interval: This means that we want to find patterns that are separated by at least min_interval but at most max_interval. For example, the pattern "If a person rents movie A, it is likely she will rent movie B within 30 days" implies int < 30 (days).
- int = c !=0 : Users may like to find patterns carrying an exact interval, int. For example, the query "Every time the Dow Jones drops more than 5%, what will happen exactly two days later?" will search for sequential patterns with int == 2 (days).

The user can specify constraints on the kinds of sequential patterns to be mined by providing "pattern templates" in the form of serial episodes and parallel episodes, or regular expressions. A serial episode is a set of events that occurs in a total order, whereas a parallel episode is a set of events whose occurrence ordering is trivial. Let the notation (E, t) represent event type E at time t. Consider the data (A,1), (C,2), and (B,5) with an event folding window width of 2, where the serial episode A->B and the parallel episode A & C both occur in the data. The user can also specify constraints in the form of a regular expression, such as (A | K) C * (D | E), which indicates that the user would like to find patterns where event-A and B first occur (but their relative ordering is unimportant), followed by one or a set of events C, followed by the events D and E (where D can occur either before or after E). Notice that other events can occur in between those specified in the regular expression.

25.9 Methods for Sequential Pattern Mining:

The Apriori property employed in association rule mining can be applied to mining sequential patterns because if a sequential pattern of length k is infrequent, its superset (of length k + 1) cannot be frequent. Therefore, most of the methods for mining sequential patterns adopt variations of Apriori-like algorithms, although they may consider different parameter settings and constraint. Another approach for mining such patterns is to explore a database projection-based sequential pattern growth technique, similar to the frequent-pattern growth (FP-growth)

Periodicity Analysis

Periodicity analysis is the mining of periodic patterns, that is, the search for recurring patterns in time-series databases. Periodicity analysis can be applied to many important areas. For exam le, seasons, tides, planet trajectories, daily power consumption, daily traffic patterns, and weekly TV programs all present certain periodic patterns.

As indicated in our discussion of the previous section, mining periodic patterns can be viewed as mining sequential patterns by taking duration as a set of' partitioned sequences, such as every year, every slot after or before the occurrence of certain events, and so on.

The problem of mining periodic patterns can be partitioned into three categories:

- Mining full periodic patterns, where every point in time contributes (precisely or approximately) to the cyclic behavior of the time series. For example, all of the days in the year approximately contribute to the season cycle of the year.
- Mining partial periodic patterns, which specify the periodic behavior of the time series at some but not all of the points in time. For example, Sandy reads the New York Times from 7:00 to 7:30 every weekday morning, but her activities at other times do not have much regularity. Partial periodicity is a looser form of periodicity than full periodicity, and it also occurs more commonly in the real world.
- Mining cyclic or periodic association rules, which are rules that associate a set of events that occur periodically. An example of a periodic association rule is "Based on day-to-day transactions, if afternoon tea is well received between 3:00-5:00 pm, dinner will sell well between 7:00-9:00 pm on weekends".

Techniques for full periodicity analysis have been studied in signal analysis and statistics. Methods like FFT (Fast Fourier Transformation) have been popularly used to transform data from the time domain to the frequency domain in order to facilitate such analysis,

The efficient mining of partial periodicity has been studied in recent data mining research. Most methods for mining full periodic patterns are either inapplicable to or prohibitively expensive for mining partial periodic patterns owing to the latter's mixture of periodic events and non-periodic events in the same period. For instance, FFT cannot be used for mining partial periodicity because it treats the time series as an inseparable flow of values. Some periodicity detection methods can uncover some partial periodic patterns, but only if the period, length, and timing of the segment in the partial patterns with certain behaviors are explicitly specified. For the newspaper reading example, we need to explicitly specify details such as "Find the regular activities of Sandy during the half-hour after 7:00 for a period of

24 hour." A naïve adaption of such methods to the partial periodic pattern-mining problem would be prohibitively expensive, requiring their application to a huge number of possible combinations of the three parameters of period, length, and timing.

Most of the studios on mining partial periodic patterns and cyclic association rules apply the Apriori property heuristic and adopt some variations of Apriori like mining methods. Constraints can also be pushed deep into the sequential pattern and periodic pattern mining process.

25.10 Review Questions

1 Explain About Similarity Search in Multimedia Data

2 Explain About Multidimensional Analysis of Multimedia Data

3 Explain About Classification and Prediction Analysis of Multimedia Data

4 Explain About Mining Associations in Multimedia Data

5 Explain About Mining Time-Series and Sequence Data

6 Explain About Similarity Search in Time-Series Analysis

7 Explain About Cases and Parameters for Sequential Pattern Mining

8 Explain About Methods for Sequential Pattern Mining**:**

25.11 References

[1]. Data Mining Techniques, Arun k pujari 1st Edition

[2] .Data warehousung,Data Mining and OLAP, Alex Berson ,smith.j. Stephen

[3].Data Mining Concepts and Techniques ,Jiawei Han and MichelineKamber

[4]Data Mining Introductory and Advanced topics, Margaret H Dunham PEA

[5] The Data Warehouse lifecycle toolkit , Ralph Kimball Wiley student Edition