A Treatise on Data Mining

Dr. N.P. Gopalan
Professor
Department of Computer Applications
National Institute of Technology Tiruchirapalli-15.

Dr.B.Siva Selvan
Assistant Professor
Department of Computer Science & Engineering
SSN College of Engineering, Chennai-603110.

Contents

Preface

The recent years have witnessed an exponential growth trend in terms of data generation and manipulation. The amount of data available these days as a result of widespread automation is tremendous and the potential to extract information and useful knowledge from such data is imminent and challenging. In this cut throat age of competitive marketing and decision making, most users are interested in analysing the current data and inferring hidden information that would be of help in future prediction and decision making. Thus the need to extract hidden and potentially useful information from large databases, referred to as data mining is only justified and imminent. This book explores the concept of data mining discussing various techniques supported by the same in the process of knowledge extraction.

Data Mining, the non trivial process of extraction of hidden and useful information from large databases is introduced in the first chapter. The various phases of the data mining process, architecture of a data mining system, types of knowledge that can be mined from databases are discussed in detail. Data mining as a technique has become a popular and essential tool primarily as a result of the extremely voluminous size of databases. It is not manually feasible to extract information from such large databases and hence the need and popularity of data mining. Different types of knowledge such as association rules, classification rules, etc. are briefly introduced in the first chapter. Infact the entire book is on the different types of knowledge that can be mined from databases and techniques for the same.

Real life data is often fraught with inconsistencies and noise and if not appropriately processed could lead to patterns or knowledge involving such noisy and inconsistent data. Data needs to be preprocessed to eliminate noise and corrupted data and be in a format that is suitable for further mining operations. The second chapter on data preprocessing primarily concentrates on this precursor step to data mining exploring various techniques such as cleaning, integration, transformation and reduction. Preprocessing techniques helps in identifying task relevant data from large databases and also reduces the data size to be processed by a data mining system by eliminating redundant and unnecessary data.

Techniques in data mining are explored from Chapter 3 onwards. Association Rule Mining, one of the interesting knowledge types, more from business domain point of view is discussed in the third chapter. An in depth discussion on the

theory and mathematics behind association rules and techniques for mining association rules is given. Frequent item-set mining phase of association rule mining, one of the most important phases is explained in detail and several algorithms for the same are considered. The concept of correlations among data base items and correlation rules that identify both positive and negative impacts of attributes on one and another is also explained. Association and Correlation rules find extensive applications in market and business oriented decision making strategies and the chapter tries to do justice by concentrating on this aspect of data mining at length.

More often than not real life applications and human beings are used to the process of assigning characteristics to instances or values. For example people in the age group 13-19 are referred to as 'teenagers',youth, senior citizens, etc. Careful observation reveals that this is essentially the process of classifying or grouping data based on certain characteristics. This is what is referred to as Data Classification in data mining terminologies and is considered for discussion in Chapter 4. Various data classification methodologies such as decision tree based algorithms, probability based techniques, etc. are explored. Another aspect of real life that is similar to classification is clustering where objects are grouped together based on similarity of certain characteristics. Real life objects are clustered or grouped according to similarity of objects and this technique in data mining is called as Clustering. Techniques for clustering, difference from classification is discussed at length.

Data prediction, outlier analysis are discussed in Chapter 6 and algorithms for the same are also explained. Descriptive data mining tasks such as characterization, generalization and comparison are explained. The last chapter concentrates on the recent trend in data mining, namely multimedia data mining. The internet and information technology revolution in the recent years has resulted in enormous amount of multimedia data such as text, audio, video, images, etc. being available. Information and Knowledge extraction from such multimedia data is what is referred to as Multimedia Data Mining (MDM).

MDM is a recent trend in relation to conventional data mining and hence most of the topics considered are recent publications. The chapter aims to introduce the readers to the recent issues in data mining, to be more precise, multimedia data mining. Interested readers must follow up this discussion with dedicated learning on each of the recently evolved data mining techniques. Data mining is a multi disciplinary field, using concepts from various areas ranging from statistics to neural networks to machine learning. A data mining system should be efficient from the data organization point of view, efficiency of algorithms for data mining, presentation of extracted knowledge in a user friendly format. We have primarily concentrated on the data mining techniques and algorithms for the same with precursor discussions on preprocessing. The book should serve as an excellent starting point material for beginners on data mining and is always best to be followed up by dedicated resource materials reference.

Chapter 1

Introduction to Data Mining

Computers today are a part and parcel of everyday life and are no longer a luxury as they once used to be. The information technology growth and revolution has resulted in wide-spread usage of computers ranging from scientific, business, engineering design, geographic applications, etc. This varied and continued usage of computer systems have resulted in enormous amount of data being available. Data is input that exists in its raw form resulting in information on further processing. With enormous amount of data available, the mankind was faced with the challenge of extracting useful and meaningful information from them, resulting in the concept of data mining.

Data Mining is formally defined as the nontrivial process of extraction of hidden, previously unknown and potentially useful information from large databases. In computer science terminologies the process of data mining operates on raw input data (input) resulting in processed information (output). The entire crux about data mining is the set of processes that achieve this transformation much like most computer science specializations.

The notion of information differs across varied domains and in this aspect the conventional database applications that manipulate input data resulting in information could be misleading. Data mining systems differ from their database application counterparts in the fact it extracts information that is not explicitly stored or available in the database. It is from this viewpoint that data mining is best described as knowledge extraction or knowledge discovery in databases. The information that is extracted by data mining systems are not explicitly available in the database whereas database applications on the other hand only project out information that are available in the database with a restricted manipulation capability. A certain degree of intelligence (knowledge) is incorporated in a data mining system, a feature that differentiates it from database applications. On these lines, a data mining system could also be treated as an intelligent database manipulation system and hence data mining is also viewed

as one of the major classifications under the field of Artificial Intelligence.

Data mining is a diverse field and encompasses concepts from statistics, database systems, machine learning, neural networks, information science, etc. The book addresses the major issues in data mining and discusses the various methodologies involved primarily from the intelligence or knowledge extraction point of view. As we go through the various data mining techniques readers will be able to appreciate the varied concepts from the above mentioned diverse fields that are employed in the process of knowledge extraction. To kick-start a debate, data mining could also be viewed as a misnomer! A better coinage would have been knowledge or information mining/extraction. But then the naming is from the viewpoint of giving more emphasis to the input (data). In a way the term is a balanced one, as it addresses both the input and output aspects of the system, namely data for the input part while mining (extract or discover) for the output part.

1.1 Phases of Data Mining-The KDD Process

Data Mining is also viewed as an essential step in the overall process of knowledge discovery, which is composed of the various phases listed in Figure 1.1. The first four phases perform data preprocessing, where in the data is prepared to be in a format that is suitable for further mining operations. Data Cleaning removes noise and other inconsistent data that is prevalent in the input database. Since the input database could be composed of data that arrives from multiple sources, data integration is employed to integrate or merge data from the various sources. Data Selection phase identifies the specific data mining task relevant data in the input database. The input database is transformed to a data mining suitable format in the data transformation phase. Next, the specific data mining task that employs intelligent methods is carried out. The resulting knowledge or patterns are evaluated for their interestingness or usability in the pattern evaluation phase. The last step of the KDD process is the presentation of the discovered knowledge in a user friendly format, referred to as the Knowledge Presentation phase.

1.2 Architecture of a Data Mining System

A data mining system is generally composed of the various components shown in Figure 1.2. A database or data warehouse unit houses the input data that is to be mined for knowledge. Given the diverse and inconsistent nature of input data, data cleaning and integration have to be performed on this component of the system. The second major component of the system is a knowledge base that is used to evaluate the resulting patterns. It helps in guiding the data mining process or pruning the search space.

Data mining engine component forms the core of the system and implements the specific data mining technique such as Classification, Clustering, etc.

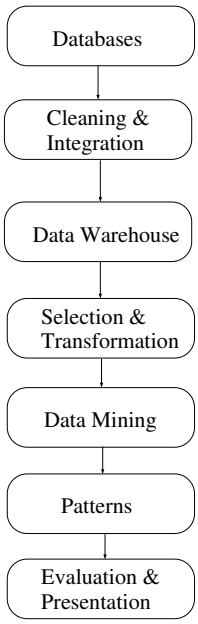

Figure 1.1: Steps in the KDD Process

It incorporates efficient algorithms to manipulate the transformed data obtained from earlier components and results in knowledge or patterns. The pattern evaluation component of the system incorporates various interestingness measures to differentiate interesting and the not so interesting patterns. Most data mining systems have this component as a part of the data mining engine itself. Finally the system supports a Graphical User Interface component that presents the extracted interesting patterns in different forms. Users are allowed to interact with the system, evaluate extracted patterns and visualize the knowledge from different perspectives.

1.3 Data Types that can be Mined

This section presents a list of different kinds of data that can be mined. As an immediate application on DBMS, data mining could be employed over relational or transactional databases to discover interesting trends or patterns. An example could be the sales database of a company or a super market that could be subject to data mining to discover interesting customer behaviours and trends.

Spatial databases that contain spatial related information such as geographic or map databases, medical and satellite images could be mined to discover patterns that detail on the features of houses located at a specific location or unearth patterns based on some distance measure. In continuation to the spatial

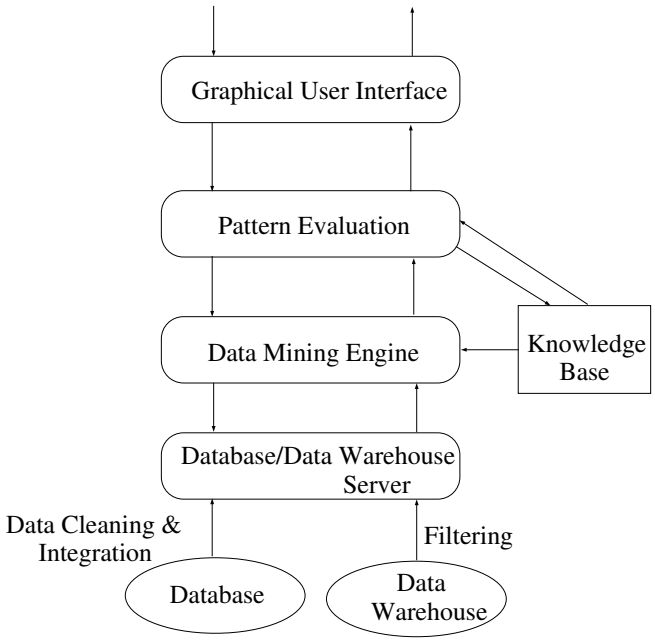

Figure 1.2: Data Mining System Architecture

discussion, another aspect of real life that is reflected in databases is temporal or time related properties. Temporal databases, whose data changes over a period of time such as bank databases could be mined to unearth knowledge that would help in decision making and strategy planning on investment options.

Text databases that contain words could be mined to discover knowledge such as the overall area or domain that the data relates to or present a short and concise summary of the otherwise long input text data. The www and the internet have a primary role in the popularity or need of data mining and the various data types that are prevalent today. The web and its usage (web data) by itself could be subject to data mining to identify patterns such as user's access behaviour, path traversal patterns, etc. Another marvel of the internet and information technology revolution is multimedia data such as images, audio and video.

The world wide web has had a major role to play in the widespread dissemination of information and usage of computers. The number of users of the internet is on the rise and there is the driving requirement to understand the behaviour of users who visit the internet, in terms of the webpages and the order in which they are visited. This technique of data mining is referred as mining path traversal patterns and helps in arriving at marketing decisions such as identifying frequently visited pages (documents) and hence place advertisements in them. Web data mining as it is referred to has also been a major

area of research in data mining resulting in several efficient techniques and varied knowledge ranging from user behaviour understanding to building semantic based search engines as opposed to the conventional and naive keyword (syntax) based ones.

The amount of multimedia data available these days over the internet or otherwise is huge and there is an imminent need to discover patterns or knowledge from them. One example could be generating the concise summary of an hour long video data so as to present the user a concise and short abstract of the input data. The recent data mining trend concentrates on multimedia databases and has resulted in a subarea of Multimedia Data Mining or MDM. MDM is the process of performing data mining on multimedia databases and forms the research area explored by the thesis. Sections to follow and the third chapter discuss MDM in greater detail and depth. The following section discusses the various kinds of knowledge or patterns that could be mined from input databases.

1.4 Knowledge Types that can be Mined

Data Mining techniques or tasks can be broadly classified as being either descriptive or predictive. Descriptive tasks are those that perform description or characterization of properties of the input database. Predictive data mining tasks are those that perform inference on input data to arrive at hidden knowledge and make interesting and useful predictions. A detailed list of the various data mining techniques and their objectives are presented in the following subsections.

1.4.1 Data Classification

It is the task of formulation of model(s) or function(s) that best describe and distinguish data classes. These models are used to predict or generate class information for data where class labels are not known. The first phase of model construction is also referred to as training phase, where a model is built or created based on the feature of the training data. It is this model that is used to predict class labels for test data, where class information is not available.

Decision trees are commonly used to represent classification models. Other representations include classification or if-then rules, mathematical formulae, neural networks, etc. A decision tree is similar to a flow chart where every node represents a test on attribute value, branch denotes a test's outcome and tree leaves represent classes. SLIQ, IC, ID3, etc are some of the existing decision tree based classifiers.

1.4.2 Prediction

It is the data mining technique that is used to predict missing or unavailable data. In a way, classification that is used to predict class labels could be treated

as prediction when numerical data are predicted. Prediction differs from classification in the fact that it is used only for numerical data prediction as opposed to classification that predicts class labels.

1.4.3 Data Clustering

Clustering is an unsupervised data mining technique as opposed to the supervised leaning approach followed by classification. The objective of clustering is to group similar objects or data and associate cluster labels with them. It differs from classification in the absence of any training data or class labels. It is based on the principle of maximizing intra-cluster and minimizing inter cluster similarity, resulting in clusters where objects within it have high similarity while across clusters objects have the least similarity amongst themselves.

1.4.4 Outlier Analysis

Data objects that deviate or do not comply with the generic behaviour of the data model are referred to as outliers. The data mining technique that deals with identification of outliers is referred to as Outlier Analysis. An application is the identification of fraudulent usage of credit, ATM cards where one unusual behaviour or outlier could be withdrawal of a huge amount as opposed to the general withdrawal trend of the customer.

1.4.5 Association Rule Mining (ARM)

It is the process of discovering interesting associations or relationships amongst data items. ARM is the process of discovering association rules that identify frequent related data items. An application is in Market Basket Analysis to identify frequently related items in a transaction. An association rule is an expression of the form $X \rightarrow Y$, where X and Y are item-sets. The rule is interpreted as the presence of items in X implying the presence of items in Y or set of database tuples satisfying the conditions in X that are likely to satisfy the conditions in Y as well.

An example association rule could be bread,jam→butter, implying the fact that transactions involving sales of items bread and jam are likely to be followed by that of purchase of a butter item. Association rules find extensive application in marketing decision making such as effective shelf space management strategies and demand projection based production. Shelf space management is about placing related items in a supermarket together or nearby and thereby increase the likelihood of a certain set of items that are preceded by the sales of another related set of items. The above class of rules are also referred to as boolean association rules, indicating the fact that they emphasize the presence or absence of items in rules.

Two statistical measures that govern ARM are Support and Confidence. Support is the percentage of transactions in a database that satisfy the rule. For an association rule $X \rightarrow Y$, support is the probability $P(X \cup Y)$, where

$X \cup Y$ indicates a transaction that contains both X and Y. Confidence is the conditional probability of Y being true subject to X or $P(Y|X)$. Confidence is the degree of certainty of detected associations while support is a measure of the statistical significance of the rule.

Two phases of ARM are (i) Frequent item-set or Set Mining (FSM) and (ii) Rules Generation. FSM is the process of generating frequent item-sets that satisfy the support threshold. The second step of rules generation is relatively less complex and straightforward compared to FSM and is the process of generating rules such as $X \rightarrow Y$.

1.4.6 Characterization cum Discrimination

Characterization, also referred to as class description describe classes or concepts in a summarized and concise form. The features that describe a class or concept is referred to as a class concept description. Class description is achieved either via data characterization that summarizes the data of the class of interest or by data discrimination that compares the target class (class of interest) with other classes, also referred to as contrasting classes.

Data Characterization is a summarization of the general features of the target data class. Examples of characterization include roll up and roll down operations possible with online analytical processing systems or OLAP that presents database information in an aggregated (roll up) or detailed manner(roll down) along a specific dimension.

Data discrimination is a comparison of the features of target class data with other contrasting data classes. An example would be to compare the sales of a particular class of items such as colour tv's in the current quarter as opposed to the previous quarter. Data characterization cum discrimination is similar to classification in the overall concept. However objective wise, classification concentrates on class labels generation based on a supervised model while data characterization and discrimination concentrates on generic features identification of class labels in the absence of any training data. Each of the above mentioned techniques are discussed in greater in greater detail and depth in the chapters to follow.

1.5 Pattern Evaluation Measures

Data Mining generates patterns or knowledge that are extracted from the input database. A data mining system would generate all possible patterns, some of which might be useful and interesting to the user while others which the user is not interested in. Hence there is a need to evaluate the generated patterns for interestingness or usability according to the specific data mining task and application. A pattern is termed interesting if it is potentially useful and valid on new test data with a fair degree of certainty. An interesting pattern is what is referred to as knowledge. Support and confidence measures discussed

earlier with respect to association rule mining are examples for interestingness measures.

Measures based on statistics and governed by user specified thresholds are infact referred to as objective measures. Subjective measures are those that either confirm user beliefs and hypothesis or rejects them. Capability of a data mining system to generate all possible interesting patterns is referred to as completeness of the system and it might not always be possible for a system to generate only interesting patterns. Completeness is generally ensured by constraints and interestingness measures such as support and confidence for ARM. The issue of generating only interesting patterns is a classical optimization problem and is generally addressed by avoiding or pruning search/pattern spaces that are likely to result in uninteresting patterns.

1.6 Data Mining System - Types

Based on our earlier discussions on data mining, there is a host of possible classification of data mining systems. One possible classification is on the basis of knowledge types that can be mined such as a Data Classification, Clustering, Association Rule Mining System etc. Data Mining is a diverse field that employs techniques from statistics, machine learning, neural networks, etc. The basic underlying technique employed can be used to classify data mining systems. Also the type of databases that are being mined such as transactional, relational, object oriented etc. is another possible line of classification. Last, the specific application for which the data mining system is evolved could be a basis for classification. Examples include financial, stock market, educational, governance data mining systems.

1.6.1 Issues to be addressed by a Data Mining System

Key research issues that have dominated the data mining area are performance, mining methodology, user interaction and data diversity. Data mining algorithms and methodologies must be efficient and scalable. The execution time of the algorithms to extract knowledge from databases must be predictable and realistic. The algorithms must scale well to the size of databases and their execution times must be acceptable for scaled databases as well.

Other performance related issues are support of parallel and distributed computing. The huge database sizes, the varied or multiple sources of data and the computational complexity necessitate the support for parallel and distributed data mining algorithms. Data from various sources or huge databases are split as partitions. It is these data partitions that are subjected to data mining in a parallel fashion and results(knowledge) from various partitions are finally merged or integrated. Another issue to be addressed by a data mining system is to support incremental knowledge extraction or mining. As real time databases keep evolving, the system should be capable of extracting new knowledge or patterns (due to the recent data updates) in an incremental fashion, without

having to mine the database all over again, starting from scratch (already mined portion).

As a result of the diverse types of knowledge that can be mined, based on user and application specific requirements, the system should support a range of data mining techniques such as association analysis to classification and clustering. Database applications that support retrieval based operations are popular as a result of friendly database query languages, on lines similar to programming languages. Database query languages such as SQL help end users (in a way, programmers) to efficiently and easily query, manipulate data in databases. On these lines, though data mining system differs from a retrieval system in the overall objective or mode of operation, a certain degree of querying facilities such as knowledge types to be mined, data mining task specific constraints and interestingness measures must be user specifiable. This would only enhance the user interaction of the system and thereby cater to the specific situation needs.

The data mining system should avoid the situation of over-fitting, wherein the model might be too specific for a database and might not fit for futuristic databases. Over-fitting primarily results due to the data specific assumptions made in the model or miniature size of the training data. Databases and their scope for varied data support suffer from the dimensionality curse. A conventional database might be composed of several attributes, not all of which could be relevant to the specific data mining task. Apart from not contributing to the data mining tasks, these unnecessary attributes might also interfere with the performance of the data mining system. In effect, the system should support methods to identify task relevant attributes or dimensionality reduction techniques. One of the phases in KDD is data selection that primarily concentrates on clearly establishing the data mining task relevant attributes.

The system must support features that present the mined results in a user understandable format. The system as a whole must be capable of handling noisy and inconsistent data using data preprocessing that filters out such overhead and unnecessary data. The system should be capable of handling diverse database types ranging from transactional to multimedia and other complex and heterogeneous data that can come from varied and distributed data sources.

Summary

The chapter introduced the readers to the concept of data mining justifying the need and benefits of the same. The prototype of a data mining system, types of data and knowledge that can be mined have been introduced briefly, each of which are considered in greater detail in the chapters ahead. The major performance issues related to a data mining system and features to be addressed were also discussed. By the end of this chapter, readers should be able to appreciate the need for data mining, interpret the notion of knowledge and have a basic idea on the various patterns that can be mined from databases. The next chapter concentrates on data preprocessing, one of the essential phases of data mining.

Review Questions

1. Justify the need for data mining.
2. Define Support and Confidence.
3. Discuss the major research issues in data mining.
4. List two applications for association and outlier analysis data mining techniques.
5. Differentiate Characterization from Classification.
6. How does data mining differ from conventional retrieval tasks?
7. Identify the various data mining tasks that you would possibly encounter from your work domain point of view.

Chapter 2

Data Preprocessing Techniques

Data preprocessing is a collection of techniques applied over input databases to eliminate noisy, missing and inconsistent data and thereby enhance the efficiency of the data mining process. A few of the preprocessing techniques mentioned in the earlier chapter are data cleaning, integration, transformation and reduction. Data cleaning aims at removing noise and inconsistency from the input data. Data reduction reduces the size of the data by eliminating redundant features and aggregation. Transformation aims at transforming the input data to a format that is suitable for further mining operations, such as normalization. The chapter discusses the various preprocessing techniques that could be applied prior to the actual data mining process.

The need for data preprocessing stems from the fact that more often than not real time data are either incomplete or noise corrupted or inconsistent. For example, several attribute values for the various tuples in the input database could be missing (not recorded or incomplete data), noisy data as a result of incorrect attribute values or errors in data transmission or manual data entry errors and inconsistent data due to deviations from the other data. All these situations require that the data be processed before the actual mining process is applied. In effect prior to the process of knowledge extraction the input data is processed (pre, i.e before the mining process) so that the mined knowledge are accurate.

Techniques such as cleaning concentrate on filling missing values, removing noise and inconsistencies in the input data. Integration takes care of assembling data from multiple database and sources avoiding inconsistencies and redundancies. Reduction aims at reducing the size of data considerably yet producing the same knowledge. In effect reduction techniques concentrate on eliminating irrelevant attributes from the input data that do not contribute to the effective knowledge base. A detailed discussion on the various preprocessing techniques is given in the sections ahead.

2.1 Data Cleaning Techniques

As discussed earlier, cleaning aims to remove noisy, inconsistent and incomplete data from the input database. One possible solution is to ignore the tuple(s) that are noisy or incomplete or inconsistent. However this might not be an attractive solution except for situations when the number of missing attributes within a tuple is large. Since databases involve a data entry at some stage of their operation, another alternative is to have the missing values filled manually with the obvious time consumption limitation.

A simple and realistic mathematical way out would be to average out the missing values by computing the mean from other valid data (attribute values for other tuples). As an extension, class (eg. manager, labourer, etc.) specific averages could also be used to fill in the missing values of the matching class. Alternatively statistics concepts such as regression could also be used to predict the missing values and forms the most effective approach to eliminate noisy and missing data situation. Most of the alternatives discussed above also handle the noisy and inconsistent input data situation with additional solutions such as clustering that eliminates outliers or data not belonging to any of the clusters identified.

2.2 Data Integration Techniques

Most of the real life data are accquired from multiple sources at varied locations with diverse specifications. One of the key issues to be addressed is to avoid duplication of attributes such as EmployeeNo, EmployeeId, etc. from the final integrated version. To be precise the integration technique aims at merging data from multiple sources eliminating duplicates in the respective schema specifications. In addition to immediate or direct duplication, there can also be the situation of can be-inferred or derived attributes.

Attributes such as grosssal can be inferred from the other basic attributes and hence can be avoided. In this process of eliminating secondary redundancy, correlation analysis is also employed. Attributes are measured for the statistical parameter of correlation with possible outcomes of positively or negatively or non correlated. Correlation between any two attributes X & Y is measured using the formula in 2.1, where n denotes the number of tuples in the input database and σ_X denotes the standard deviation of X.

$$corr(X,Y) = \frac{\sum(X-\bar{X})(B-\bar{B})}{(n-1)\sigma_X\sigma_Y} \tag{2.1}$$

Positive correlation indicates that the relationships between the attributes is high and can be handled by eliminating the redundant ones. Cases where the attributes are not correlated (measure value=0), the attributes are deemed independent and included in the integrated version. Negative correlation indicates that the attributes have negative impact on one another, i.e if the value of one attribute increased, then the negatively correlated attribute's value decreases.

In other negative correlation equates to inverse proportion. Another key issue to be addressed as a part of integration is the detection and resolution of conflicts in data values from different sources where the units used to measure could differ such as currency differences.

2.3 Data Transformation Techniques

Transformation is the process of converting the original input data into alternate forms that are suitable for further mining. Techniques such as smoothing that removes noise from data, aggregation, normalization are all used for transformation. Normalization scales the data to specified and desired ranges and helps in faster data mining algorithms. Types of normalization are decimal scaling, z-score and min-max. Decimal scaling normalization divides attribute values by powers of 10, depending upon the scale one requires.

Zero score normalization uses man and standard deviation of attribute values to normalize based on the expression $val' = \frac{value - \bar{X}}{\sigma_X}$, where val is the value of Attribute X and val' is the normalized value. Minmax normalization does linear transformation using the minimum and maximum values of attributes and the specified range maximum and minimum using the expression in 2.2. Another transformation technique is also to insert new attributes (inferred ones) in situations where the computed values are frequently used in the data mining algorithms. An example would be to include TaxPaid attribute which can be added to the basic structure and hence contribute to better efficiency of the data mining algorithm.

$$val' = \frac{val - min_X}{max_X - min_Y}(newmax_X - newmax_Y) + newmin_X \tag{2.2}$$

2.4 Data Reduction Techniques

Reduction results in reduced(size) representation of data and hence contribute to efficient data mining. It aims at eliminating irrelevant, redundant attributes and employs encoding mechanisms to reduce the data set size. The technique where aggregate operations (monthly sales as opposed to weekly) are applied resulting in the construction of a data cube (a condensed representation of the original database) is referred as data cube aggregation.

A data cube is basically a model that allows the data to be represented in multiple dimensions. The technique of eliminating irrelevant and redundant attributes is referred to as dimension reduction and the methodology where encoding mechanisms such as run length encoding are employed is called as data compression. Numerosity reduction is the alternative of replacing data values with estimated smaller representations such as clusters, samples, etc. Also replacement of data values with ranges and higher conceptual values is another solution to data reduction. Concept hierarchies help in multiple level abstraction of data and knowledge extraction at the varied hierarchical levels.

2.4.1 Data Cube Approach

Data cube aggregation employs aggregate operations such as summarization in generating a integrated view of the data. An example would be the construction of a data cube sales data area aggregated on a monthly basis as opposed to the weekly basis on which they have been recorded in the database. Every cell of the cube stores an aggregate value, in addition to which concept hierarchies can be incorporated for each of the attributes thereby allowing multiple level abstraction of the data. They provide a faster alternative to store precomputed and aggregated data. The data cube that reflects abstraction at the lowest level is referred to as the base cuboid and the one at the highest level as the apex.

2.4.2 Dimension Reduction Approach

The input database could be ideally composed of several attributes many of which are irrelevant to the potential data mining process. Dimensionality reduction primarily concentrates on identifying such irrelevant attributes and hence reduce the volume and improve the efficiency of the data mining process. Attribute subset selection strategies are employed to identify the set of relevant attributes. Dimensionality reduction techniques help in reducing the data volume and also make the knowledge (patterns) extracted from the data mining process easier to interpret, as they will not involve the relevant attributes and will be reduced in nature.

Given that for n attributes there are 2^n possible attribute subsets, an exhaustive searching would be time consuming and expensive. Greedy approaches that identify best and worst attributes based on statistical significance measures such as information gain, gini index are used in the process of identifying relevant attributes. Stepwise forward selection, backward elimination and combination of both are some of the solutions to attribute subset selection strategies.

In forward selection, new attributes (relevant) are added to the initial empty set of relevant attributes in a relevant fashion. In backward elimination approach, the original set of attributes constitutes the relevant attributes set and irrelevant attributes are identified and eliminated as before in a step wise fashion. In the combined approach, at each step the best attribute is added and the worst is eliminated from the set of relevant attributes. Classification algorithms such as ID3, SLIQ, etc. make use of attribute selection measures such as gini index to identify the classification determining attributes (attributes that would make up the decision tree.

Techniques where the data mining process by itself employs attribute selection methods such as classification are referred to as wrapper (attribute subset selection strategy is wrapped or enclosed within the data mining engine) while others are referred to as filter systems in the fact that prior to data mining process irrelevant attributes are filtered (removed).

2.4.3 The Compression Approach

Data compression techniques also aid in reducing the dimensionality of the data. Compression techniques are broadly classified as being lossless or lossy. Cases where the original data can be reconstructed entirely from the condensed form is referred as lossless and lossy otherwise where only an approximate of the original data can be reconstructed. Lossless compression techniques such as run length encoding permit only basic and minimal reduction of data size. On the other hand lossy techniques such as principal component analysis, independent component analysis and wavelet transforms can be employed.

Principal component analysis aims at represent data using orthogonal vectors so that the original voluminous data is projected on a smaller space. It differs from attribute selection strategies in the fact that it creates new and alternative set of variables (attributes) that capture the meaning of the database. PCA employs normalization to avoid domination in the process of data compression. The orthogonal vectors (principal components) are derived on the basis of the eigen vectors of the covariance matrix obtained for the normalized dataset. The wavelet transform approach much like Fourier transforms is a signal processing approach where representatives of the strongest coefficients are used to create a condensed representation of the original data.

2.4.4 Principal Component Analysis (PCA)

PCA is a way of identifying patterns in data and a data representation format that highlights the similarities and differences in the data. It serves as a powerful tool for analysing data and offers the advantage of data compression by reducing the number of dimensions without much loss of information. A stepwise algorithm for PCA is given in Figure 2.1. We have only given a short introduction to the compression approach and interested readers must pursue each of the varied techniques in further detail as it would be beyond the scope of a text on data mining.

2.4.5 Independent Component Analysis (ICA)

Independent Component Analysis aims at identifying a set of independent components that best explain the data given a set of observations. It has its basis from the cocktail party problem where the objective is to hear the voice of one person from that of several persons voices. The problem is often referred to as the Blind Source Separation problem where the objective is to separate the observed signal into its original components or sources, without any knowledge (blind) about the mixing of the source signals.

ICA is based on the assumption that given a vector of observations ($x = (x_1, \ldots x_n)$), each of these observations can be derived from a set of n independent components, i.e $x_i = a_{i1}s_1 + \ldots + a_{in}s_n$, which can alternatively be represented as x=As in matrix notation. Here s represents the random vector which is the collection of the independent components and A is the mixing ma-

1. The input data to be analysed is read.
2. The data is transformed(in a way normalization) by subtracting the mean from each of the data dimensions. If we consider data across two dimensions x & y, then each of the data values of x & y are subtracted from their respective means namely $\bar{x}\&\bar{y}$, resulting in a dataset whose mean is zero.
3. The covariance matrix of the transformed data is obtained, where the matrix is a square matrix across the number of data dimensions.
4. Eigenvectors and eigenvalues of the covariance matrix is computed. Eigen vectors which are perpendicular to each other reveal information about patterns in the data.
5. From the computed eigen vectors and eigen values, a feature vector is constructed. Feature vector is a list of eigen vectors of significance stored as columns in the feature vector matrix. Eigen vector with the highest eigen value is the principal component of the data set and eigen vectors with larger values convey the most significant relationship between the data dimensions.
6. Then the final data set is derived from the feature vector and the original transformed data set. As a part of PCA computation the the feature vector and the transformed data are transposed and then multiplied resulting in the derived data set where data are represented in terms of the orthogonal vectors determined.

Figure 2.1: Stepwise Computation of PCA

trix. Components S_i are independent of each other,i.e $P(S_i, S_j) = P(S_i)P(S_j)$. ICA assumes that the distributions of the components are far from normal as possible. Limitations of ICA are that variances and order of independent components cannot be determined resulting in the situation where one cannot evaluate the relative importance of the components. Independence of components is measured using kurtosis, entropy, mutual information content.

2.4.6 Singular Value Decomposition (SVD)

Singular value decomposition is yet another powerful dimensionality reduction technique and finds application in Latent Semantic Indexing. Given a original data matrix A (n*p), the decomposition is represented as $A_{n*p} = U_{n*n}S_{n*p}V^T_{p*p}$, where U and V are orthogonal matrices ($U^TU = I$), I being the identity matrix. SVD computation involves the generation of eigenvalues and eigen vectors of AA^T and A^TA. Eigen vectors of A^TA constitute the columns of V and that of AA^T make up the columns of U. Singular values stored in matrix S are the square roots of the eigen values obtained from AA^T or A^TA.

2.4.7 Factor Analysis

Factor Analysis, a model evaluation technique is a collection of methods used to examine the underlying constructs's influence on the responses on variables. Two types of factor analysis are exploratory and confirmatory. Exploratory factor analysis (EF concentrates on discovering the nature of the constructs that influence a set of responses. On the other hand confirmatory factor analysis (CFA) tests whether a specified set of constructs is influencing the responses in a predicted manner. Observed responses are partially influenced by the underlying common and unique factors. Factor analysis is performed by measuring the correlation between the observed measures. Positively or highly correlated measures are likely to be influenced by the same factors while negative correlated measures by different factors.

EFA aims to determine the number of common factors that influence a set of measures and the strength of relationship between each factor and observed measure. EFA finds application in determining the important features in classification, identifying related set of items that are governed by similar factors, etc. EFA consists of the following steps:

1. Measurements Collection: All the variables (data) are measured along the same experimental units.

2. Correlation matrix computation: The relationship or correlation between the variables is computed and represented as a matrix.

3. Identifying number of factors for inclusion: Optimal number of factors criterion such as number of eigen values of the correlation matrix that are greater than 1 (kaiser), number of eigen values prior to major dip

in magnitude (eigen values of correlation matrix arranged in descending order).

4. Initial factor extraction and manipulation: The set of initial factors are extracted using likelihood estimation and manipulate using orthogonal and oblique rotations resulting in uncorrelated and correlated factors.

5. Factor structure Interpretation: Strength of the relationship between measure and factors is controlled by the factor loading parameter which is used in factor score analysis to evaluate the various factors.

Confirmatory factor analysis concentrates on testing the ability of a predefined factor model to fit an observed data and finds application in testing the validity of factor models, compare more than one model for the same data, factor correlationship, etc. The model to be validated is selected and then the variables to tested are measured along the same experimental units. Correlation matrix is generated and then the model is fit for the observed data. Model adequacy is ensured by comparing the correlation matrix implied by the model and the actual observed matrix. Tests such as chi-square goodness of fit test are employed to confirm or reject the null hypothesis that the model adequately fits the data.

CFA is generally used when model is available and cases where information about the responses to measures is not available, EFA is used. EFA and CFA are used in combination wherein after establishing a model using CFA, EFA can be used to locate fine inconsistencies between the model and data and result in a further detailed model.

2.4.8 Numerosity Reduction

Numerosity reduction which aims at replacing existing data values with much smaller ones can be achieved using the regression concept of statistics. Linear regression systems where a random variable is represented as linear function of the the predictor variable. An example linear regression function would be B=x+yA, where B is the response variable, A is the predictor variable and x,y are the regression coefficients, evaluated using the method of least squares. Linear regression is also extended to support response variable prediction based on multi dimensional feature vectors, referred to as multiple regression.

Histogram representation of data is also another alternative to numerosity reduction where individual attribute values/ranges are represented on the X axis and their corresponding counts on the Y axis. Types of histograms are equiwidth, equidepth and variance optimal. If ranges selected are uniform it is referred to as equiwidth histogram and if the frequency of values within a selected range/individual value are approximately the same, it is referred to as equidepth histograms. Variance optimal histogram concentrates on identifying the least variant from the set of all possible histograms for a given number of ranges.

Clustering could also be used as an alternative to numerosity reduction where the principle of grouping similar objects within a cluster is exploited to replace actual data with cluster representatives. Cluster representatives are selected on the basis of a distance measure. Sampling techniques with and without replacement could also be employed to replace original data with condensed representation drawing random samples from the original data set.

Summary

This chapter discussed the various data preprocessing techniques namely cleaning, transformation, integration and reduction techniques. Preprocessing phase constructs data to be in a format suitable for further mining operations eliminating noise, redundant and inconsistent data from databases. Dimensionality reduction techniques help in cutting down on data size by dropping redundant and unnecessary data. The chapter has introduced readers to the various data and numerosity reduction techniques. The following chapter explains Association Rule Mining, one of the fundamental and widely used data mining technique in business applications.

Review Questions

1. Explain the various data cleaning alternatives.
2. Justify the need for data integration techniques.
3. Define normalization and explain the various types and apply the same for a data set of your choice.
4. Define dimension reduction and explain in detail the attribute selection approach.
5. Compare and contrast the various dimensionality reduction techniques. (refer external sources as well).

Chapter 3

Association Rule Mining

Association Rule Mining is the process of discovering interesting relationships and associations amongst the various items in the database. It primarily concentrates on transactional databases with the objective of establishing associations amongst the various items involved in a transaction. This chapter concentrates on Association Rule Mining exploring the fundamentals and algorithms for the same. It is one of the basic and interesting techniques in data mining. Association Rule Mining as a data mining field has been extensively researched resulting in several efficient algorithms and methodologies. The widespread usage and popularity of association mining is its immediate application scope mainly from business perspectives. It is also used a precursor to other data mining techniques such as classification, clustering, prediction. Hence we start our exploration on data mining on this basic, interesting and important branch of the data mining tree.

3.1 Introduction

In this modern age of aggressive marketing and business practices, organizations are all the more interested in analysing the current purchasing trend or patterns of customers and hence engage in demand oriented production and increased sales strategies. Association rule mining or association mining as defined earlier is the process of discovering associations or relationships between the database items. It finds application in market basket analysis, also referred to as MBA. Market analysts would be interested in identifying the frequently purchased items, so that the organization can adopt effective shelf space management and efficient sales strategies.

As an example, an association involving items such as milk, butter, jam could help the analyst in arriving at an association (knowledge or pattern) such as sales of milk and bread together is likely to be followed by the sales of the item jam. The immediate advantage of such associations would be in shelf space management, which is to place the related (associated or frequently purchased)

items on near by racks rather than far apart or in random fashion. In this manner, the likelihood of the item being seen by the customer and the effective sales of the commodity is increased. On the contrary, placing related items far apart could indirectly contribute to the sales of other unrelated/related items that lie in between them (assuming the customer traverses the market from $item_1$ one end to $item_2$ on the other end.)

Two statistical measures of significance that control the process of association rule mining are support and confidence. Support is statistical significance of a rule while confidence is the degree of certainty of the detected associations. The entire process of association mining is controlled by user specified parameters namely minimum support and confidence. Discovered association rules are deemed to be interesting only if their support and confidence satisfy the minimum threshold specified by the user.

3.2 Association Rule Mining Fundamentals

An Association Rule is basically an expression of the form $X \rightarrow Y$, where X & Y are item-sets. An item-set is nothing but a collection of database items. The above rule is interpreted as the set of tuples in the database that satisfy the conditions in X are likely to satisfy the conditions in Y as well. In simple terms, the presence of items in X implying the presence of items in Y as well. On these lines association rules are also referred to as boolean rules as they indicate the presence or absence of items, in which case the associations are referred to as negative associations.

Two statistical measures that govern ARM are Support and Confidence. Support is the percentage of transactions in a database that satisfy the rule. For an association rule $X \rightarrow Y$, support is the probability $P(X \cup Y)$, where $X \cup Y$ indicates a transaction that contains both X and Y. Confidence is the conditional probability of Y being true subject to X or $P(Y|X)$. Confidence is the degree of certainty of detected associations while support is a measure of the statistical significance of the rule. For an association rule $X \rightarrow Y$, support and confidence are mathematically represented as shown in equations 3.1 and 3.2.

$$Support = \frac{Number\ of\ Transactions\ Containing\ both\ X\ \&\ Y}{Total\ Number\ of\ Transactions\ in\ the\ database} \tag{3.1}$$

$$Confidence = \frac{Number\ of\ Transactions\ Containing\ both\ X\ \&\ Y}{Number\ of\ Transactions\ containing\ X} \tag{3.2}$$

Association Rule Mining as a data mining process has been traditionally viewed as two phase approach, consisting of frequent item-set mining and rules generation phases. The first phase of frequent item-set construction or mining is the major and crucial step of association mining in relation to the rules generation phase. Much of the research over the years has primarily concentrated on this aspect of association rule mining resulting in several efficient approaches and algorithms for the same.

An item-set that occurs in the database the user specified minimum support number of times is said to be a frequent item-set. In other words an item-set I whose support count (number of times it appears in the database (primarily a transactional database)) is termed a frequent item-set. Frequent item-sets are generally represented by L denoting the fact that they are large item sets. A k item-set is nothing but an item-set composed of k items(elements). In effect the process of frequent item-set mining concentrates on generating all possible frequent item-sets whose length ranges from 1 to k, k being the number of distinct frequent item-sets of length 1, represented as L_1.

The second step of rules generation concentrates on generating association rules of the form $X \rightarrow Y$ using the generated frequent item-sets, X & Y being collection of item-sets. Finally only rules that satisfy the user specified minimum support and confidence thresholds are retained and referred to as strong association rules. It is these rules that satisfy the user parameters that are used in decision making and management strategies. This step is relatively less complex in comparison to the frequent item-set mining phase and involves a straightforward manipulation of the generated frequent item-sets and their subsets. This chapter discusses the various algorithm in the literature for frequent item-set mining and the rules generation phase.

3.3 Apriori Algorithm for Association Mining

One of the first algorithms to evolve for frequent item-set and association rule mining was Apriori proposed by Agrawal et.al. Apriori is a level-wise algorithm and is based on the antimonotonic property of set theory which states that every subset of a frequent item-set is also frequent, which is to say given that an item-set is frequent, all possible subsets of the same are also deemed to be frequent. Apriori is a candidate generation algorithm and proceeds in a level wise fashion.

Two major steps of the apriori algorithm are the join and prune steps. The join step is used to construct new candidate sets. A candidate item-set is basically an item-set that could either be frequent or infrequent with respect to the support threshold. Higher level candidate item-sets (C_i) are generated by joining previous level frequent item-sets or L_{i-1} with itself. It is in this aspect that Apriori is treated as a level wise algorithm.

The prune step helps in filtering out candidate item-sets whose subsets (prior level) are not frequent. This is based on the anti monotonic property as a result of which every subset of a frequent item-set is also frequent. Thus a candidate item-set which is composed of one or more infrequent item-sets of prior level is filtered (pruned) from the process of frequent item-set and association mining. As an example if set 1,2 is infrequent then candidate item-set 1,2,3 will also be infrequent and hence is pruned. The algorithm is given in Figure 3.1 and an illustration of the same for the sample input in Table 3.1 is shown in Figure 3.2 with minimum support threshold =2(20%).

1. $L_1 = Frequent\ items\ of\ length\ 1$.
2. for $(k = 1; L_k! = \phi; k++)$ do
3. C_{k+1}=Candidates generated from L_k;
4. for each transaction t in database do
5. increment the count of all candidates in C_{k+1} that are contained in t.
6. L_{k+1}=Candidates in C_{k+1} with minimum support.
7. end do
8. return the set L_k as the set of all possible frequent item-sets.

Figure 3.1: Apriori Algorithm

Trans_id	Items in Transaction
1	3,4
2	2,3,5
3	1,2,3,5
4	2,5
5	1,2,5
6	1,3
7	2,3
8	1,3
9	2,3
10	3,5

Table 3.1: Sample Input Database

C_1	L_1	C_2	L_2	C_3	L_3
		12(2)	12(2)		
1(4)	1(4)	13(3)	13(3)	123(1)	
2(6)	2(6)	15(2)	15(2)	125(2)	125(2)
3(8)	3(8)	23(4)	23(4)	135(1)	235(2)
4(1)	5(5)	25(4)	25(4)	235(2)	
5(5)		35(3)	35(3)		

Figure 3.2: Apriori Algorithm Illustration

3.4 Working of the Algorithm

Given the input database in Table 3.1, Apriori first identifies frequent item-sets of length 1 or L_1. In the example above, excepting item 4 all the four items (1, 2, 3 & 5) satisfy the support threshold and hence L_1=1,2,3,5. Successive stages of the algorithm generate candidate item-sets from previous level frequent item-sets. Thus C_2 is generated by self joining L_1 with itself resulting in the item-sets {12,13,15,23,25,35} or column 3 in the Table 3.2. Since the entire C_2 set satisfies the minimum support threshold, L_2=C_2. L_2 when self joined with itself results in {123,125,135,235} (column 5), out of which only item-sets 125(2) and 235(2) are frequent constituting the set L_3. Further candidate item-set generation (and frequent item-set generation) is not possible because of the antimontone property and hence the algorithm terminates with frequent item-sets of length 3

3.5 Association Rules Generation

This is the second phase of association rule mining and concentrates on mining rules of the form $X \rightarrow Y$, where X & Y are item-sets. For each frequent item-set (f) all possible non-empty subsets (s) are generated and rules of the form $s \rightarrow f - s$ are constructed. The generated association rules are retained only if supportcount(f)/supportcount(s) is greater than the minimum confidence threshold specified by the user. Note that confidence is the conditional probability expressed in terms of the itemset support count and for a rule $X \rightarrow Y$ is expressed as $\frac{support\ count(X \cup Y)}{support\ count(X)}$.

3.6 Limitation of Apriori

Apriori despite its simple logic and inherent pruning advantage suffers from the limitation of huge number of repeated input scans. Since it is a level wise algorithm it requires separate scan of the input database and over the entire frequent item-set mining process, this becomes tedious and is a serious limitation. Much of the early research in association rule mining concentrated on eliminating this repeated input scans setback resulting in several efficient algorithms such as Frequent Pattern (FP) tree growth, Dynamic Item-set Counting (DIC), etc.

However variations of Apriori exists which minimize the overhead of repeated input scans. One of the variation is Algorithm AprioriTid (Apriori Transaction id), emphasizing the fact that transactions in the database are replaced by candidate itemsets that occur in that transaction. This variation of Apriori performs well at higher iterations or levels whereas the conventional Apriori performs better at lower levels since the entry might be longer than the corresponding transaction. Hence evolved the concept of AprioriHybrid which combined the advantageous features of Apriori and AprioriTid. Subsequent sections present the other algorithms and approaches to frequent itemset mining.

Another limitation of Apriori based approach is the generation of candidate sets which can become cumbersome and time consuming when the number of frequent 1 item-sets is large. FP growth algorithm is a pattern growth approach that generates patterns on the fly and avoids the candidate pattern logic of Apriori. Only those patterns or item-sets that are encountered in the input database are maintained by the algorithm thereby eliminating the need to maintain large number of patterns. Thus it offers significant improvements in both time and space complexity and is explained in the sections to follow.

3.6.1 Dynamic Item-set Counting Algorithm

The Dynamic Item-set Counting (DIC) algorithm proposed by Brin et.al aimed at reducing the number of database scans by dividing the database into intervals of specific sizes. Similar to Apriori, all candidate item-sets of length 1 is generated. L_1 is generated by the end of the first scan, midway of which when the interval specified is reached, counting of candidate item-sets of length 2 is started. The approach is similar to a train running over the data with stops at intervals M transactions apart.

Assuming the database contains 20,000 transactions and M=5000, item-sets of length 1 are counted over the entire transaction set. However midway the counting, after the first 5000 transactions counting of item-sets of length 2 and after the first 1000 transactions counting of item-sets of length 3 is commenced. Momentarily assume that only frequent or candidate item-sets of length 3 are possible. Thus when the end of the database (first scan is over) is reached, L_1 generation is complete and the algorithm resumes to count item-sets of length 2 and 3. After the first 5000 transactions, counting of item-sets of length 2 is also complete and after the first 10,000 transactions counting of item-sets of length 3 is completed. In effect the algorithm reduces the number of database scans to 1.5 as opposed to 3 scans required by a level wise Apriori approach. The algorithm is based on the overall principle of counting for item-sets whenever it is optimal rather than having to wait for the completion of the previous pass.

3.6.2 CARMA Algorithm

The performance of the DIC algorithm is heavily dependent on the actual distribution of data in the database. The CARMA (Continuous Association Rule Mining Algorithm) proposed by Hidber et.al adopts an approach similar to DIC with interval size(M) as 1. Thus candidate item-sets are generated on the fly from every transaction that is processed. Once a transaction is processed, support of all candidate item-sets contained in that specific transaction is incremented and also new candidate item-sets in that transaction are generated, subject to the condition that subsets of it are frequent with respect to the number of processed transactions.

CARMA results in more candidate sets compared to Apriori and DIC despite offering the flexibility of allowing the user to change minimum support threshold values during the algorithm's execution. Superset of all frequent item-sets is

guaranteed by the end of the first scan and by the end of the second scan exact counts of frequent item-sets is generated.

3.6.3 The Sampling Approach

The Sampling approach proposed by Toivonen et.al require a maximum of two database scans. It first selects a random sample from the input database and finds all frequent item-sets in the sample. The results are then verified with the rest of the database. In situations of actual frequent patterns being missed out, a second scan of the database is used to generate the counts of left out frequent patterns. The overall logic is heavily dependent on the failure rate which in turn is dependent on the minimum support threshold. The threshold requires to be decreased drastically in cases of failure, resulting in the possibility of candidate set explosion.

3.6.4 Partitioning Approach

The partitioning algorithm proposed by Savasere et.al adopts a divide and conquer approach to the process of frequent item-set mining. The input database is partitioned into several disjoint parts (partitions) and frequent items within the partitions are generated. Later the frequent item-sets obtained from various partitions are merged, resulting in a superset of all frequent item-sets in the complete database. This is based on the reasoning that an item-set that is frequent in the complete database must be relatively frequent in one of the partitions. Actual counts of the frequent item-sets are computed during a second scan of the database. The partitioning approach is again dependent on data distribution and heterogenity of the database. There can be situations where not all candidate item-sets within a specific partition can be fit in the memory.

3.6.5 FP growth Algorithm

The previously discussed approaches have primarily adopted a breadth first strategy towards the issue of frequent item-set mining. The size of the candidate item-sets can be reduced by adopting a depth first logic as done by FPgrowth algorithm. The FPgrowth(Frequent Pattern) algorithm stores the transactions of the database in a tree format and every item has a linked list that goes through all transactions that contain that item. Initially the database is scanned once to generate the set L_1. Later a modified transactional database that includes only those items that are in L_1 and in its descending order is created for the purpose of the tree construction. In fact this tree data structure is referred to as the frequent pattern, referring to the fact that it contains all frequent patterns.

A tree is created with empty root node and then branches are populated with the processing of every transaction (modified ones). Also the nodes in the tree maintain a count of the number of transactions that share that specific node. As and when branches are added, count of a node in the tree is suitably incremented. An item head table that points to the various occurrences of

1. The first phase of Frequent Pattern tree approach is construction of the frequent pattern tree. The tree is constructed as follows:
   (a) Scan the original database once to identify set L_1 and then rearrange L_1 in the descending order of support count (count).
   (b) Grow a tree with empty root node (null). During the second scan, starting from the null root node, add paths to the tree corresponding to the reordered transactions (transaction whose items have been reordered according to the descending order property of L_1), updating the node's count. For transactions that share common ancestor nodes use the existing nodes in the tree and update the node's count suitably. Cases where a common ancestor node is not present, new nodes are added from the null root node and the procedure repeated as mentioned above.
2. The second phase of FP tree algorithm is to mine or generate the frequent patterns from the constructed FP tree (earlier step output).
   (a) For every element (reverse order of descending order L_1), now locate positions in the tree where the nodes appear in the tree and list out the paths as the conditional pattern bases, with the considered element treated as a common suffix.
   (b) From each of the conditional pattern bases determine the cumulative frequent pattern tree counts.
   (c) Finally mine the frequent patterns by suffixing L_1's element with the frequent pattern tree values obtained in the earlier step.

Figure 3.3: Frequent Pattern Tree Algorithm

items in the tree is also created. FP growth approach is based on the principle of reducing the size of the database representation by maintaining the more frequently occurring patterns near the root and hence increase the likelihood of sharing nodes in the tree structure. The created FP tree is mined to generate the various frequent patterns subject to the minimum support threshold.

3.6.6 The FP growth Algorithm

A detailed illustration of the algorithm for the sample input database considered earlier with Apriori algorithm is given in the next section.

Trans_id	Items in Transaction
1	3
2	3,2,5
3	3,2,5,1
4	2,5
5	2,5,1
6	3,1
7	3,2
8	3,1
9	3,2
10	3,5

Table 3.2: Reordered Input Database

3.6.7 FP tree Algorithm Illustration

The Frequent Pattern growth approach requires two overall scans of the input database to generate all possible frequent item-sets. The first scan much like Apriori helps in finding out frequent and infrequent 1 item-sets (L_1). This step basically helps in filtering out the infrequent 1 items at a much earlier stage of the algorithm rather than proceeding with too deep with the infrequent item-sets. The set L_1 or frequent 1 item-sets is sorted in descending order of their support. For the example considered earlier for the explanation of Apriori, the descending ordered set L_1 is as follows: 3(8), 2(6), 5(5), 1(4). Note that the item 4 is infrequent since its support count(1) is less than that of the minimum support threshold (2).

During the second scan of the database a tree data structure (FP tree) is grown starting from an empty root node (null). Prior to construction of the FP tree, the original database is reordered(items within transactions are reordered) in the order of L_1 (descending order). It is these transactions that are added as paths to the tree. For the sake of reader convenience we give the reordered database (shown in Figure 3.2). Note that this step is primarily a result of the tree growing phase and hence no separate scan of the database would be required as is likely to be misinterpreted.

As can be seen from the table, the first transaction which was originally 3,4 is reordered as 3 (note that item 4 is infrequent and hence left out). Similarly the second transaction 2,3,5 is reordered as 3,2,5 and so on. It is these reordered transactions that are added as paths of the FP tree structure. Thus for the first transaction a path from the root node (null) is added with its count set to 1. For the second transaction 3,2,5, the already existing path 3 is updated with count 2 and new paths with nodes 2 and 5 and counts set to 1 are branched from the existing node 3.

For the third transaction the entire path $3 \rightarrow 2 \rightarrow 5$ is reused with their counts suitably incremented to 3, 2 and 2 with an new node 1 added as child of node 5 and count set to 1. Thus the overall logic of tree growth adopted in FP tree approach is that existing paths are reused to the extent possible. However

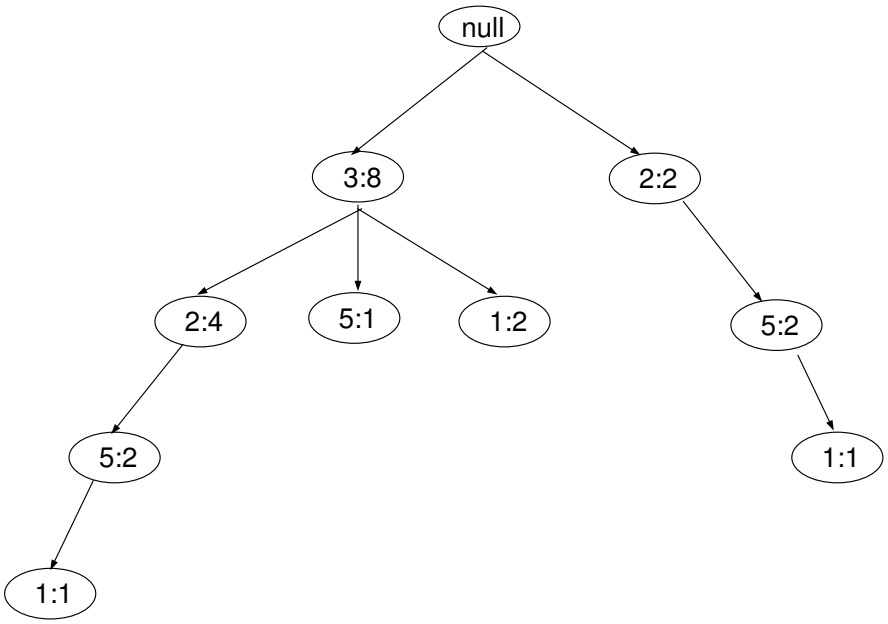

Figure 3.4: Final FP tree Structure

if the starting label (node) of a transaction is other than the currently existing child node from the root, as is the case for the fourth transaction 2,5 a new node 2 is added from the root of the tree and paths grown suitably. The final FP tree structure by the end of the second scan is as shown in Figure 3.4.

The FP tree algorithm also maintains an item header table to aid the process of tree traversal and pattern extraction. The item header table contains the reorder items in L_1 with pointers to the various locations in the tree where the respective item appears in the tree (pointers maintained on cumulative basis). The item-header table and the respective pointers to the FP tree are as shown in Figure 3.5. The second major phase of FP tree algorithm is the mining of the FP tree to generate the frequent patterns.

The FP tree constructed is mined as follows. We start from each frequent 1 item-set or pattern (from the last element of reordered L_1) and constructs its conditional pattern base which is nothing but all those paths in the tree that finally lead to the considered element of L_1. This suffix is not added to the conditional pattern base and included in the final frequent patterns. From the conditional pattern base, patterns (FP tree) are constructed recursively satisfying the support threshold retaining those that satisfy the minimum support threshold. For the example considered earlier, the conditional pattern base, FP trees and frequent patterns mined are as shown in Figure 3.6.

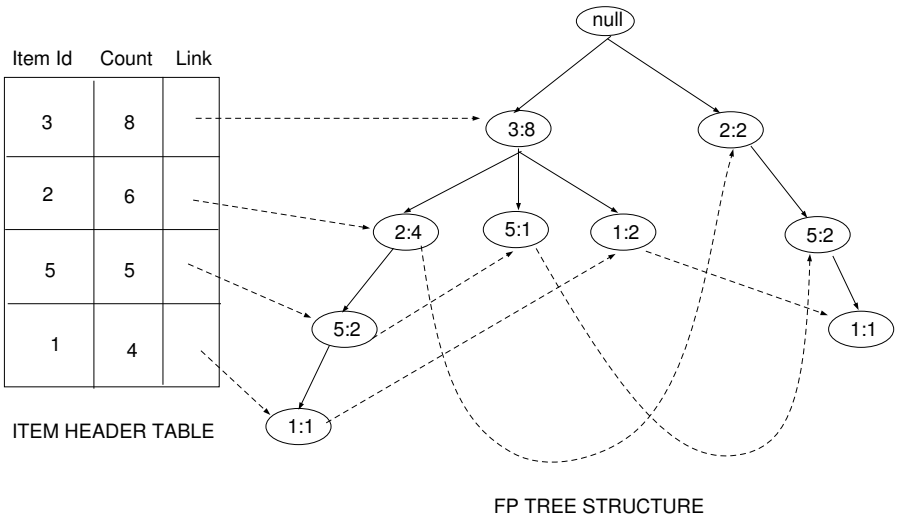

Figure 3.5: Item Header table with pointers to the FP tree

Item	Conditional pattern base	Conditional FP tree	Frequent Patterns
1	{(3 2 5:1),(3:2),(2 5:1)}	(2 5:2),(3:3)	{251(2),21(2),51(2),31(3)}
5	{(3 2:2),(3:1),(2:2)}	(2:4 3:3 :2)	{25(4),35(3),235(2)}
2	{(3:4)}	(3:4)	{32(4)}

Figure 3.6: FP Tree Mining

	Tea	$\overline{Tea}$	Row_Sum
Coffee	300	200	500
$\overline{Coffee}$	400	100	500
Col_Sum	700	300	1000

Table 3.3: Sample Transaction Database (contingency table)

3.7 Correlation Rules

Association rules help in identifying relationships between item-sets based on the occurrence of items and using the support, confidence parameters. Another class of rules that helps in identifying the negative associations such as occurrence of item A does not imply the occurrence of B are referred to as correlation rules. Consider the transactional database where among the 1000 transactions, 300 involved the purchase of both tea and coffee, 200 involved only coffee, 400 only tea and 100 involved none of the two items. An association rule such as tea $\rightarrow$ coffee would have support of 30% and confidence of 43%. As should be clear, this is one instance where association rules might not always convey the actual relationship or effect of one item over the other.

Realistically speaking coffee and tea are competing products (appropriate assumptions) and the rule currently mined could be misleading. Correlation rules employ the mathematical concept of correlation analysis between variables (items) and based on the computed measure, item relationships are identified. Correlation between the occurrence of A and B is measured as $\frac{P(A \cup B)}{P(A)P(B)}$. If the measured correlation is less than 1 then A & B are interpreted to be negatively correlated while on the other hand a computed value of ¿1 is treated as A & B being positively correlated. If the computed value is exactly equal to 1, then A & B are completely independent of one another. Assume the following transactional database (represented as a contingency table) as shown in Figure 3.3.

From the table one can infer the fact that probability of a customer buying tea is 0.7, probability of a coffee purchase is 0.5 and the probability of both the purchase is 0.3. Now the correlation measure between tea and coffee as defined by the earlier equation is 0.3/0.7*0.5=0.85. Since the correlation measure is less than 1, the two items tea and coffee are negatively correlated. Such negative correlations or the negative impact of an item over another can be mined only via correlation analysis and not by conventional association rule mining or analysis. Thus correlation analysis would be very helpful in situations where negative impact among transactional items have to be established.

Correlation rules exhibit the property of upward closedness in the sense that given a set of items is correlated, all supersets of it are also correlated. Thus addition of items to a correlated set does not modify correlation. The chi-square test can also be used to determine negative implications and exhibits upward

	i_1	$\overline{i_1}$	$\sum_{row}$
i_2	300	200	500
$\overline{i_2}$	400	100	500
$\sum_{col}$	700	300	1000

Table 3.4: Sample contingency table

closedness at each level of significance. The chisquare test of significance starts out with an initial empty set of correlated items and add one by one.

The objective is to determine the minimally correlated itemsets or item-sets that are correlated but none of their subsets are correlated. Such item-sets are referred to as border item-sets and since all supersets of a minimal correlated item-set would be correlated, upward searching could be avoided. However chisquare tests depend heavily on the contingency table and could result in inaccuracy when the table is sparse.

3.7.1 Algorithm based on Chisquare test for mining correlated item-sets

The chisquare significance level based test for correlation is given in Figure 3.7. An item-set is termed SIGNIFICANT if it is supported and minimally correlated. Item-set at level i+1 can be significant only if all its subsets at level i have support and none of its subsets at level i are correlated. For a level i+1 the objective is to find all supported and uncorrelated sets from level i stored in NOTSIGNIFICANT. The set SIGNIFICANT contains the supported and correlated item-sets and forms the output of the algorithm. CANDIDATE contains candidate sets for level i+1 from NOTSIGNIFICANT sets at level i.

Candidate item-sets are then moved to the SIGNIFICANT and INSIGNIFICANT category based on the computed chi-square value. Computed chi-square is expressed as $\sum_i (O(i) - E(i))^2/E(i)$, where O(i) is the observed frequency or count and E(i) is the expected frequency or count of the considered item-set i. Contingency table between two item-sets say i_1 and i_2 is constructed involving their compliments and is as shown in Table 3.4. However the limitation of chisquared tests for independence and correlation is that it is suited for situations where all cells in the contingency table have expected value more than 1 and 80% of the cells have value more than 5.

Correlated item-sets satisfy the criterion of upward closure where if an item-set i is correlated, then all of its super-sets are also correlated. This is a constructive strategy unlike the downward closure based support measure which is a pruning strategy. Upward closure property helps avoiding false positives (an item-set that is frequent at a prior can always become infrequent at the next level), whereas upward closure results in false negative, where certain correlated (i+1) item-sets can be ignored. Correlation rules concentrate on minimally correlated item-sets wherein all supersets of the same are correlated and none of its subsets are.

1. For every item-set generate the count of it (O(i)), which will also be used in expected value computation.
2. Initialize CANDIATE, SIGNIFICANT and NOTSIGNIFICANT sets to empty.
3. For every pair of items i_1 & i_2 satisfying the condition O(i_1) and O(i_2) are > s(the support threshold), the pair of item-sets (i_1,i_2) are added to CANDIDATE.
4. Set NOTSIGNIFICANT to empty.
5. If CANDIDATE set is empty, return SIGNIFICANT and exit.
6. For every item-set in CANDIDATE set, contingency table for the itemset is constructed and if less than p (support fraction) % of the cells satisfy the support count(s), go to step 8.
7. If the computed chi-square value (χ^2) is >= the tabulated chi-square value (χ^2_α), then the itemset is moved to the SIGNIFICANT category (added) or else to the NOTSIGNIFICANT category.
8. Repeat the process with the next CANDIDATE item-set which when empty is set to K such that every subset of size $|k|$ -1 of K is in NOT-SIGNIFICANT and go to step 4.

Figure 3.7: Chi-square based correlated item-sets algorithm

3.8 Advanced Concepts in Association Mining

3.8.1 Multilevel Association Rules

Associations that satisfy user specified minimum support and confidence thresholds are referred to as Strong associations. In a database associations might not be strong at lowest levels of abstraction as compared to higher levels. As an example associations between bread and jam would be more strong compared to those between specific brands such as breadX and jamY. Data mining algorithms in generally must support varied levels of abstraction or operational levels and leave it to the end user to mine knowledge at the required level of abstraction. Frequent item-sets are mined at each conceptual hierarchy level such as HCLCOMP, COMPUTERS, etc. Frequent 1 item-sets at the first level are mined and then frequent 2 item-sets at higher levels are mined. One approach is to use a uniform support threshold for all levels while another alternative is to have lowered support thresholds at lower levels of abstraction as compared to the higher levels.

The uniform support approach offers the advantage of a simplified search procedure and can also be optimized to avoid items whose ancestors do not satisfy the minimum support. However the disadvantage of such a uniform support logic would be that if the value is too high then several meaningful associations at lower levels could be ignored. The alternative to this is to have lowered support thresholds at lower levels of abstraction and then gradually increase it at the higher levels. Also level wise independent, filtering by item(set) across levels are employed to mine multilevel associations.

Level cross filtering strategies consider an item-set at the i^{th} level only if its parent at the $(i-1)^{th}$ level is frequent. Filtering based on k item-sets where a k item-set at level is considered only if its corresponding parent k item-set at level i-1 is frequent. This filtering might prove to be too restrictive and result in many associations getting pruned out. Filtering by an item considers an item at level i only if its parent node at level i-1 is frequent. This approach might again miss out on associations between low level items at reduced support factors.

A level passage threshold that allows frequent items to be passed down is used, thereby permitting child items that satisfy the passage threshold to be examined and result in more meaningful associations. This allows mining associations such as bread $\rightarrow$ jam Y or inter level associations to be mined. Such rules are also referred to as cross association rules. The system should ensure that specific rules convey new information not available from generic rules. Examples of such situations are specific rules breadX$\rightarrow$jamY and generic rules such as bread$\rightarrow$jam. The specific rule does not convey any new information and is best treated as redundant and eliminated from further consideration.

3.8.2 Multidimensional and Quantitative Association Rules

Association rules are classified as single and multidimensional rules. Single dimensional rules are those which involve only one predicate such as purchase of

items (bread, jam, etc.). These rules are also referred to as intra dimension association rules, where the relationship is restricted to from within the dimension (predicate). Rules that involve multiple predicates or dimensions from more than one table are referred to as multidimensional rules. Examples could be rules involving dimensions age and salary of an employee (age, salary) with the type of computers purchased (purchase predicate).

Association rules in general and multidimensional rules in specific could involve both numeric(integer valued) as well as categorical(values from a set) attributes. Numeric attributes are introduced in the rules either in the form of ranges based on predefined hierarchies or using binning (quantitative association rules) or distance based methods (distance based association rules). Distance based rules aim to capture the semantic of the range representation and then approximate. A representative attribute from the original database is selected and the tuples are projected along it. Close tuples based on euclidean distance measure (discussed in greater detail in the Chapter on clustering) are used to cluster or group the tuples into various clusters. Clusters are optimized using both density as well as a frequency threshold. Then associations are mined from clusters across the various projected attributes.

3.8.3 Guided Association Rule Mining

The entire process of association rule mining could be made more accurate with knowledge or background information guiding the process. Meta (template) rules could be used to generate specific association rules that correlate to the database attributes and values. These metarules are like templates or a syntax of the to be generated rules. Examples include metarules such as $Predicate_1 \wedge Predicate_2 \wedge \ldots \wedge Predicate_n \rightarrow Purchase(pname)$ and specific instantiations such as age(X,40-50)$\wedge$ salary(X,20-30K)$\rightarrow$FlatDisplays.

Association mining could also be guided by conditions to be satisfied by the mined rules. SQL format based conditions as a part of the where clause checking for support and confidence thresholds, groupby and having clauses are a few examples of constraint guided association mining. This effectively helps in reducing the search space

Summary

The chapter has provided an indepth discussion on association mining and the various algorithms in the literature for the frequent item-set mining phase. The basic Apriori and advanced algorithms such as FP tree, DIC and other methodologies have been addressed. Impact of correlation analysis in association rule mining has also been explored leading to the domain of correlation rules. The field of association rule mining has been extensively researched since its inception and what has been given in this chapter is just a short introduction and is best followed up with a detailed survey of the latest techniques referring to research publications.

Review Questions

1. Correlate the application of association rules in your work domain.
2. Compare and contrast the various frequent item-set mining methodologies.
3. For the input database given below generate association rules with minimum support=2 (22%) using Apriori and FP tree algorithms.

Tid	Items
1	1,2,5
2	2,4
3	2,3
4	1,2,4
5	1,3
6	2,3
7	1,3
8	1,2,3,5
9	1,2,3

4. Explain in detail the limitations of Apriori algorithm.
5. Define support and confidence.
6. Generate association rules for a realistic market database and test the significance of the generated knowledge. (assignment).
7. Differentiate association from correlation rules.
8. Explain the algorithm used to compute correlated item-sets.
9. Differentiate upward and downward closure property.
10. Implement the Apriori and FP tree based association rule mining algorithms in a programming language of your choice. (assignment)
11. Generate a survey of the various frequent item-set and association mining algorithms.(refer external source/assignment)

Chapter 4

Data Classification Techniques

Data Classification is the data mining technique of generating models for input databases and classify database records or tuples. In other words it is the process of classifying the database records into one of the predefined classes. An example application could be an employee database to classify employees as belonging to upper, middle and lower classes. Several techniques exist in the literature to perform data classification. ID3, SLIQ, IC, etc. are a few of the existing classifiers or classification algorithms. This chapter elaborates on classification, exploring the various data classification algorithms and concepts.

4.1 Data Classification Fundamentals

Data Classification in terms of Artificial Intelligence is a supervised learning technique, involving two major phases of Training and Testing. It is supervised in the sense that the process of assigning class labels is based on a model or prototype created from the existing class information available in the training data. In other words, the training data is already classified data where the individual records have classes (labels) assigned to them and the objective of the training phase is to build a model from this data. It is this classification that is used to perform classification on the test data. Test data is one where the records are devoid of class label information. In terms of a real life organization, training data could be viewed as past data (history) of the organization, while test data could be the list of new entrants.

The training data tuples are classified from among the set of predefined classes (say upper, middle, lower, etc.) with the employee database example. Records in the training data are also referred to as training samples or objects, as these tuples serve as model data for the further process of classification and model creation. Since class label information is made available from the training data point of view, the approach is referred to as being supervised (by the

training data). In other words the training data guides the process of data classification of test data. Data Mining techniques where an explicit demarcation as training and test data are not available, or where the data mining task does not have a model data to proceed from are referred to as Unsupervised techniques. Examples include clustering which aims at grouping related or similar records or tuples. The groups are referred to as clusters. Clustering concentrates on grouping records in the input database without any background information on the number of groups and the grouping process.

Accuracy of the data mining process is generally tested using a hold out method that uses a test set purely for the purpose of testing the validity of the developed classification model. The test set is made up of tuples randomly drawn and independent of the training samples, to avoid the optimistic situation of the model tending to overfit the test set. Predefined classes of the test set are compared with those generated via the classification model. Depending on the result of comparison, the model is either put to use for classifying future test data(data where class labels are not known) or it is refined for further accuracy. Classification finds extensive application in medical diagnosis such as classifying cancer related records as being malignant, abnormal or normal, etc. It can also be employed in selective marketing by communicating new promo materials only to those new customers who are likely to purchase the product than to all of the existing customers.

4.1.1 Data Preprocessing for Classification

Data cleaning, selection or relevance analysis and transformation are the preprocessing techniques employed prior to the actual classification process. As explained in the earlier chapter on preprocessing techniques, data cleaning removes noise employing smoothing and replaces missing values using averaging or other statistical approaches. Wrapper techniques which incorporate cleaning as a part of the data mining task are also common, but an explicit cleaning phase clears the way for further learning, eliminating corrupt data at the earliest stage.

Data selection phase performs relevance analysis by evaluating the impact of attributes for the specific data mining tasks. With respect to classification, this step identifies attributes that determine the class labels for database records (training data). Relevance analysis helps in eliminating irrelevant (attributes which do not determine or have any role to play in deciding the class labels). With respect to the employee database setup, class labels are likely to be determined based on attributes such as age, salary drawn, investment, etc. rather than attributes such as name, sex, address, etc. This is also referred to as feature selection, where key features for the data mining task are identified. Data selection should contribute significantly to the learning process, wherein the time for learning from the original data set as compared to that from the reduced relevant set should be more.

Normalization techniques can also be employed to avoid a wide range and large magnitude attribute values. Also categorical data such as address attribute

could be normalized using concept hierarchies to a condensed representation format. Addresses could be manipulated (transformed) on city level as opposed to the low level street values.

4.1.2 Evaluating Classification Models

Data classification systems must support key characteristics of accuracy, speed, robustness, scalability and interpretability. Accuracy refers to the ability of the model to correctly predict class labels for unknown or test data. Misclassified labels or classes must be to the extent minimal possible. Capability of the model to correctly predict class labels even in the presence of noise and inconsistent data is referred to as its robustness. Ability of the system to have as fast a response time as possible is referred to as its speed and Scalability in data mining refers to the ability of the system to respond well to increasing database sizes. The classification model proposed should provide a better understanding and interpretation of the data and this trait is referred to as interpretability of the system.

4.2 Decision Tree Model based Classifiers

Decision trees have been prevalent in the computer science literature for a long time with varied applications ranging from search techniques to control structures. A decision tree is nothing but a flow chart that predicts the future course of actions depending on certain conditions being satisfied. A classical example is the if else construct supported in most programming languages that chooses a particular set of actions (block) based on condition(s) specified being satisfied or not.

Decision trees have been correlated in the domain of data classification resulting in algorithm such as SLIQ(Supervised Learning in Quest), ID3(Induction Decision), IC(Interval Classifier), etc. Decision trees in the context of classification are composes of internal nodes, leaf nodes and branches. Internal nodes correspond to attributes and tests on them, leaf nodes denote the final classes (class labels) and branches denote the outcome of tests on attributes (branches arise from internal nodes as a result of the outcome). As an example a node such as age<=40 tests for the age attribute of a record being less than or equal to 40 or not and depending upon the outcome either further nodes are grown or leaf nodes (class labels) are added to the tree.

Ideally a classification decision tree or a classifier cannot be composed of just one node since classes are likely to be characterized by more number of attributes such as age, salary, investment, etc. Consider that in an employee database classes such as upper, middle and lower are based on two attributes namely age and salary (for the sake of simplicity). A sample decision tree could be as shown in Figure4.1.

In the above decision tree, the three classes Upper, Lower and Middle are characterized by two attributes namely age and salary. The tree is interpreted

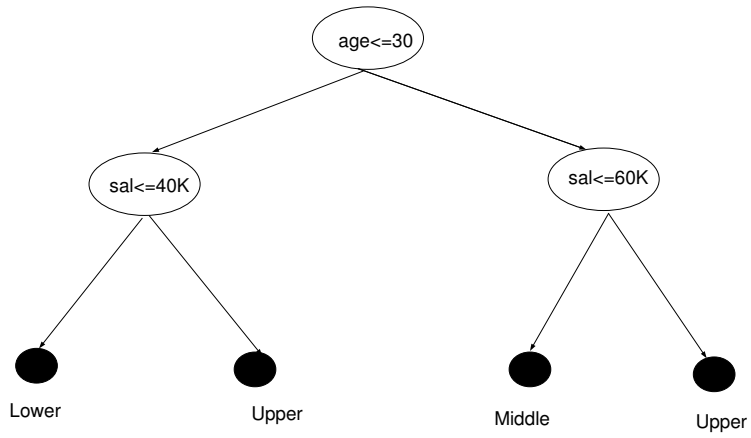

Figure 4.1: Example Decision Tree Classifier

as records whose age is <=30 and salary is <=40K belong to Lower class. For records with age <=30 but salary>40K, the class is Upper. On similar lines, age being > 30 and salary either <=60K or else the records are classified appropriately as Middle or Upper. Leaf nodes denoting the class labels are represented as darkened circles. Note that this is only an illustrative example purely for the purpose of understanding the notion of a decision tree and its functioning in the domain of a classifier. Further examples in the chapter will consider an actual input and perform classification.

Decision tree model thus classifies records from the test data by traversing the nodes in it which actually perform the various condition tests and checks on the attributes characterizing the classification process. As a result of their structure, decision tree model can be alternatively represented in the form of if then rules, also referred to as classification rules, which would be just a syntactical representation of the tree paths and outcomes. The following section presents one of the basic decision tree classifiers, namely inductive decision tree classifier or decision tree constructed by induction.

4.3 Decision Tree Induction based Classifier

Decision tree based on induction employs the divide and conquer approach and constructs decision trees in a top down manner. From among the various classification determining attributes, the one with the best attribute selection measure is selected and made the root node of the tree. Attribute selection measures employed could be Gini Index, Information gain, etc. and are explained in detail in sections to follow. These measures basically help in ranking the classification determining attributes and the tree is populated with nodes according to the ranking obtained.

1. Create a Node (root node) in the Tree.
2. If all records or samples in the training database belong to the same class label C, then a leaf node is created with label C and terminated.
3. If attribute value list is empty, create a leaf node labelled with majority class label and terminate.
4. Determine the test attribute based on the attribute selection measure rank.
5. For every value of the chosen attribute or based on n-ary model
6. Add a tree branch with the test condition specified
7. Assume s is the number of records in the training db which satisfy the test condition.
8. If s is empty then create a leaf node labelled with the majority class label.
9. else grow the decision tree inductively (call the decision tree build algorithm again with the new set of attributes).

Figure 4.2: Inductive Decision Tree Classifier Algorithm

To start with the root node of the tree represents the entire training database. If the input is such that all records belong to the same class, say upper, then the algorithm terminates with the class label (leaf node) added as a child to the root node. However with realistic inputs this is hardly the case. The training database when composed of n classes, branches are created for every known value of attribute and the records are partitioned accordingly. Alternatively a binary or n-ary tree decision model could also be incorporated to avoid the situation of too much overfitting. The tree is thus grown in a top down manner (huge database to relatively smaller set of records belonging to the same class) with internal nodes being added.

Inductive growth of the decision tree and recursive partitioning of the training database is terminated only when all the leaf level nodes are composed of records belonging to the same category or class. However when a node runs out of values for tests, then the leaf node is created and labelled with the class label of majority of the samples in it. The inductive decision tree algorithm is shown in Figure 4.2.

4.3.1 Attribute Selection Measures

One of the key phases of decision tree induction classification model is to rank the classification determining attributes based on some parameter. Information gain is one such m measure that is used to evaluate attributes which describes

the amount of information contained in the attribute. The attribute that reflects the maximum or highest information gain or entropy is chosen as the first decision tree node. Such a measure results in a decision tree model that requires the minimal information required to classify training samples and pure partitions. Information measure primarily aims at reducing the number of tests that are required to classify records and thereby contribute to faster classification. Number of paths traversed in the decision tree is directly proportional to the time required for classification. Gini index is also another attribute measure that aims to minimize impurity of a decision tree node and is employed in the SLIQ classifier.

Let S represent the number of training data records and n denote the number of class labels. Let s_i represent the number of samples in S that belong to class C_i. Expected information required to classify a given sample is as shown in Equation 4.1, where p_i is the probability that a random sample is of type C_i. Since information is represented as bits log 2 base is used.

$$I = \sum_{i=1}^{n} p_i log_2(p_i) \tag{4.1}$$

Assume that an Attribute A has n distinct values, that could be used to partition a training data set S into n partitions, with each partition being defined by the attribute value a. If s_{ij} denotes number of samples of a particular class C_i of a subset S_j, the expected information or entropy based on partitioning by subsets is as shown in Equation 4.2.

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + s_{2j} + \cdots + s_{nj}}{s} I(s_{1j}, \ldots, s_{nj}) \tag{4.2}$$

The first part of the summation acts as the weight of the j^{th} subset. Information gain aims to minimize the entropy value to the extent possible resulting in less number of impure partitions. Information gain is represented as the difference between the expected information to classify and based on the partitions, or difference between equations 1 & 2. The following section presents an illustration for the decision tree induction algorithm and the attribute selection measure information gain.

4.3.2 Illustration for Decision Tree Induction

Consider the training database shown in Figure 4.3. There are three distinct classes namely Upper, Lower and Middle. EmployeeID is more of a record identifier field and ideally cannot form a classification relevant attribute. Similarly City of an employee also has the least relevance to classification. Thus the relevant attributes in the sample database are age and salary. However if one wishes to filter lists entirely on the information gain or other attribute selection measures, one can always do. With respect to this example, we shall compute the information gain measures for age and salary and then construct the decision

EmpID	Age	Salary	City	Class
550	<= 20	50K	Chennai	Upper
600	<= 20	60K	Trichy	Upper
200	21 – 30	40k	Bangalore	Middle
400	> 30	50K	Mumbai	Middle
100	> 30	60k	Delhi	Upper
300	> 30	40K	Mumbai	Lower
250	21 – 30	60K	Chennai	Upper
225	21 – 30	50K	Delhi	Upper
475	<= 20	10K	Chennai	Lower
675	<= 20	20K	Delhi	Lower

Figure 4.3: Sample Training Database

tree model using the induction algorithm. Let the three classes Upper, Middle and Lower carry equivalent representations in numeric order (1, 2 & 3). Thus for Age¡=20 s_{11}=2, s_{21}=0 and s_{31}=2. On similar lines for Age:21-30, s_{12}=2, s_{22}=1 and s_{32}=0 and for Age¿30, s_{13}=1, s_{23}=1 and s_{33}=1. Thus the expected information to classify a record based on age partition is
$E(Age) = \frac{4}{10}(I(s_{11}, s_{21}, s_{31})) + \frac{3}{10}(I(s_{12}, s_{22}, s_{32})) + \frac{3}{10}(I(s_{13}, s_{23}, s_{33}))$. The expected information to satisfy a given sample is as follows:
$-\frac{5}{10}log_2\frac{5}{10} - \frac{2}{10}log_2\frac{2}{10} - \frac{3}{10}log_2\frac{3}{10}$, which is equal to 1.48.
$Gain(Age) = I(S_1, S_2, S_3) - E(Age)$
Computations for E(age):
$I(s_{11}, s_{21}, s_{31}) = -\sum_{i=1}^{m} P_{ij}log_2(P_{ij})$
The various information gain factors for age attribute works out as 0.92,1.06 and 1.38 respectively for age attribute with values of $<= 20, 21-30 \& > 40$. The the net information gain of age attribute as 1.48-1.1= 0.38. A similar computation for salary computation results in gain of 0.727. Thus the order of significance of attributes in the decision tree construction process would be salary followed by age. The decision tree resulting from the execution of the algorithm in Figure 4.2 with classification determining attributes in the order of salary followed by age is as shown in Figure 4.4. Note that leaf nodes in the tree are represented by darkened circles denoting the various possible classes Upper(U), Middle(M) and Lower(L) respectively. Class abbreviations are used for clarity of the decision tree representation. Assuming that the unknown (input) record from a test database has characterizing attribute values of Salary=40 and Age=40, the class label generated by a simple traversal of the appropriate branches of the tree is Lower(L).

4.3.3 Decision Tree Pruning

Pruning aims at fine tuning the constructed decision tree to result in optimal classifiers. Pre and post pruning are the approaches to avoid overfitting of the data. In prepruning, pruning is performed during the process of tree growth.

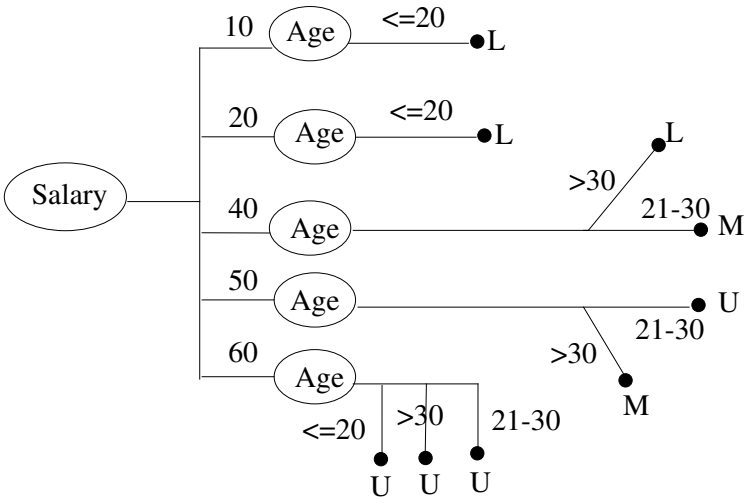

Figure 4.4: Decision Tree for the Sample Database in Figure 4.3

Further splits or partitions are avoided and the node at which prepruning is being attempted is converted to a leaf node with label of the majority class in the respective partition.

Goodness of fit tests such as chisquare are used to assess whether further split at a node exceeds the minimum threshold or not. Splits are avoided if the computed value falls below the threshold or otherwise continued. Higher thresholds correlated to specific decision trees while lower threshold refers to generic trees. Postpruning performs pruning on the completely constructed decision tree. This strategy computes the error between removing or pruning a subtree and the case when the subtree is not pruned. If pruning results in increased error, then the subtree is retained or else the subtree or partition is avoided.

Another principle for pruning is to employ the minimum description length strategy to minimize the number of encoding bits resulting in the best pruned tree. Post pruning results in far more reliable trees compared to prepruning at the cost of increased computation.

4.3.4 Performance Related Issues in Decision Tree Induction

The basic decision tree induction algorithm requires the attributes to be categorical or discrete resulting in enormously wide decision trees and branches. One alternative to this exhaustive approach is to adopt a binary partitioning approach as done in SLIQ. For an attribute with n values, there would be n-1 possible split values and generally at successive partitions, midpoints of adjacent

values is considered. Alternatives to the information gain attribute selection measure, such as gain ratio, gini index are also used.

Decision tree node splitting could result in to the problem of fragmentation when the number of samples at a given branch becomes negligible. Fragmentation is generally handled by grouping categorical attributes. Binary decision trees also reduces fragmentation resulting in more accurate classifiers. The case of an attribute repeatedly being tested along a tree branch is referred to as repetition. The case of subtrees being duplicated within the tree is also a common issue in decision tree modelling. All the three issues are handled by attribute construction where new attributes are created from the existing ones. Decision tree algorithms must also support the feature of incremental update where for incremental data versions, the algorithm should process only the new data rather than having to learn from the scratch.

4.4 Other Decision Tree Classifiers

4.4.1 The SLIQ Classifier: Supervised Learning In Quest

SLIQ[1] is a decision tree based classifier that can handle both numeric as well as categorical attributes and also scales well to the database sizes. It employs a novel presorting technique during the tree building phase and performs classification of disk resident data sets. It improves classification learning time without much loss in accuracy. In relation to other classifiers, SLIQ executes faster and results in compact trees. Two phases of SLIQ much like CART(Classification And Regression Trees), C4.5 are building and tree pruning. The tree building algorithm consists of the steps described in Figure 4.5.

The tree pruning phase attempts to prune out noisy and inconsistent branches in the tree. Two major components of the tree building phase are evaluation of splits for attribute, selection of best split and partitions creation using the best split. Partitions are nothing but application of splitting criterion on the data. Attributes are evaluated for splitting using the gini index, given by $gini(T) = 1 - \sum p_j^2$, p_j being the relative frequency of class j in the data.

Numeric attributes are handled using a binary split of the form $A \leq v$, where A is the attribute name and v is the value or real number for the attribute. The training data set is presorted, sorting based on the values being considered for splitting. Assume $v_1, v_2, \ldots, v_n$ are the sorted values of a numeric attribute. The midpoint of an interval is typically chosen as the split point. Categorical attribute splits are handled using split conditions of the form $A \in S'$, where S' is a subset of S, the set of possible values of the categorical attribute.

SLIQ eliminates the need to sort the data at each node of the decision tree by sorting the training data just once at the beginning of the tree building phase. Separate lists for the classification relevant attribute are created and a class list that associates class labels with the samples is created. Attribute list entries contain attribute values and an index to the class list. Class lists contain the class label and the reference to the decision tree leaf node. Partitions are

1. maketree (Training Data T)
2. begin
3. Call Partition (T)
4. end
5. Partition (Data S)
6. begin
7. if (all points in S are of the same class) then return
8. evaluate splits for each attribute A_i
9. use best split to create partitions S_1 and S_2
10. Call Partition(S_1) and Partition(S_2)
11. end

Figure 4.5: Generic Tree Building Algorithm

identified by predicate conjunctions of the decision tree paths and the class list identifies the partition to which an example belongs to. To start with leaf reference fields of all class list entries point to the decision tree root. Attribute values are tagged with class list index. Split tests on attributes are evaluated using the algorithm given in Figure 4.6.

4.4.2 Interval Classifier

Interval classifier for data mining applications[2] is yet another decision tree classifier that handles numeric attributes values by generating nodes in the form of class intervals or ranges and categorical attributes have individual branches created for the various values. IC compares well in classification accuracy and speed in comparison to inductive decision tree algorithm. IC classifier consists of two steps namely tree building and traversal to generate classification functions for the various classes or groups. The tree building algorithm consists of the steps shown in Figure 4.7.Goodness functions such as information gain are used to evaluate attributes for splitting criterion. Generation of classification functions requires a simple traversal of the decision tree with attributes along the path till a leaf node (class label) is reached are expressed in a conjunct representation.

1. for every attribute
2. begin
3. traverse the attribute list of A
4. for every attribute value v in the attribute list
5. find the corresponding entry in the class list, class node and leaf node, say l
6. update the class histogram in the leaf l
7. if A is a numeric attribute then compute gini index for leaf l
8. end
9. if A is a categorical attribute
10. begin
11. for each leaf of the tree
12. find subset of A with best split
13. end
14. end

Figure 4.6: Algorithm for Evaluating Attribute Splits

1. maketree (Training data T)
2. begin
3. Partition the training data according to the groups.
4. for all groups and attributes
5. begin
6. generate histograms for the varying groups over the entire domain of the attributes
7. apply goodness functions to identify winner attribute A
8. Partition domain of A into weak and strong intervals
9. Each strong interval is assigned the winner group
10. end
11. end
12. for the weak intervals of A with subset data T_l
13. call the make tree procedure again.

Figure 4.7: Tree building phase of Interval Classifier

4.5 Bayesian Classification

Another class of classification systems that are not based on actual attribute values but on statistical parameters are referred to as Bayesian Classifiers. As the name suggests, the Bayes Theorem in probability forms the basis for Bayesian Classifiers. These classifiers in relation to their decision tree counterparts offer the advantage of high speed and accuracy. The naive bayesian classifier works on the assumption that the impact of an attribute value on a specific class is independent of the values of the other attributes. This property is referred to as the class conditional independence. Bayesian belief networks unlike naive bayesian classifiers allow representation of dependencies among subsets of attributes and are also used for classification

Assume that D is the test data sample for which class labels have to be predicted and that A is an hypothesis or assumption that D belongs to a particular class C, the problem of classification is interpreted as determining the probability that the assumption A or hypothesis holds given the observation that D is of class C type. It is also represented by conditional probability P(A|D), referred to as the posterior probability of A conditioned on D. P(A) is the a priori (prior) probability of A. In other words posterior probability is based on background information such as D in the earlier example, as opposed to prior probability that is independent of X. P(D|A) is the probability of the data sample being true given that it belongs to class C. These probabilities P(A|D), P(D), P(D|A) are employed in the bayes theorem to predict the class label given a data sample as follows: $P(A|D) = \frac{P(D|A)P(H)}{P(D)}$.

The bayes theorem based naive bayesian classifier based on bayes theorem employs the following steps:

1. All the data samples or records are represented by an n dimensional feature vector such as $V = (v_1, v_2, \ldots, v_n)$ which represent the n observations of the records with attributes $A_1, A_2, \ldots, A_n$.

2. Assume that there are m classes such as $C_1, C_2 \ldots, C_m$. Given a test data D with unknown class label, objective of classification is to predict the class for D which has the highest posterior probability conditioned on D. Thus the bayesian classifier assigns an unknown sample D to a class C_i if and only if $P(C_i|D) > P(C_j|D), for 1 \leq j \leq m, j \neq i$. The objective is to maximise the probability of a class C_i being the assigned class given a sample D. As explained earlier, according to bayes theorem,probability of the predicted class label is as shown below.
$P(C_i|D) = \frac{P(D|C_i)P(C_i)}{P(D)}$.

3. It can be observed that P(D) is constant for all classes and hence the maximization process oriented term is $P(D|C_i)P(C_i)$. Also since the class prior probabilities are equally likely, the term to be maximized is $P(D|C_i)$. Cases where all classes do not have the same probability contain the $P(C_i)$ term, which can be computed from the number of

$P(Age = 40\|Upper) = \frac{1}{5} = 0.2$
$P(Age = 40\|Middle) = \frac{1}{5} = 0.2$
$P(Age = 40\|Lower) = \frac{1}{5} = 0.2$
$P(Salary = 40\|Upper) = \frac{0}{5} = 0$
$P(Salary = 40\|Middle) = \frac{1}{5} = 0.2$
$P(Age = 40\|Lower) = \frac{1}{5} = 0.2$
$P(D\|Upper) = 0.2 * 0 = 0$
$P(D\|Middle) = 0.2 * 0.2 = 0.04$
$P(D\|Lower) = 0.2 * 0.2 = 0.04$

Table 4.1: Bayesian Classification Computed Probabilities

samples of the particular class divided by total number of samples, i.e $P(C_i) = \frac{No.\ of\ Training\ samples\ with\ class\ C_i}{Total\ Number\ of\ Training\ Samples}$.

4. The property of class conditional independence helps in treating the attribute values as being conditionally independent of one another. Thus $P(D|C_i)$ is expressed as product of $P(d_1|C_1)P(d_2|C_2)\ldots P(d_m|C_m)$, where $P(d_m|C_i)$ is nothing but the number of training samples of class C_i with attribute value for D as d_m divided by the total number of training samples of class C_i. With respect to continuous valued attributes, normal distributions are used to predict the probability values.

5. Finally to classify an unknown data sample, $P(D|C_i)P(C_i)$ is evaluated for each class C_i and the sample D is assigned to the class C_i such that $P(D|C_i)P(C_i) > P(D|C_j)P(C_j)$, for $for 1 \leq j \leq m, j \neq i$

4.5.1 Illustration for Bayesian Classifiers

Consider the sample database considered earlier for the explanation of decision tree based classifiers shown in Figure 4.3 for bayesian classifiers as well. Assume that the test data record to be classified (D) has the following values for age, salary fields 40, 40. The objective is to generate a class label (Upper|Lower|Middle) based on training data information available and employing bayesian classification model. The various probabilities required for the model are as in Table 4.1. The probabilities for the individual classes are 0.5, 0.2 and 0.3 for Upper, Middle and Lower respectively. Now the maximization term evaluates to 0, 0.008 and 0.012 respectively for the Upper, Middle and Lower classes respectively. As can be observed 0.012 is the maximum value and hence the unclassified record is assigned the class label Lower. Note that this is a random illustration and values considered might not model a real time distribution of data.

The bayesian classifier makes the assumption that the effect of an attribute value on a given class is independent of the values of other attributes or what is referred to as conditional class independence. However real time database entries are likely to encounter situations of dependency between variables. Thus

conditional class independence need not be always true and hence classifiers must be in place that handle this situation as well. Bayesian belief networks based on the principle of joint conditional probability distribution provide a way out. They define conditional independency between a subset of variables and depict causal relationships between the variables.

Belief networks are basically composed of a directed acyclic graph and a conditional probability table. The acyclic graph is made up of the variables as nodes and the conditional or probabilistic dependence between them is denoted by arcs. An arc from node X to Y indicates X to be the parent or immediate predecessor of Y. Given a variable , it is treated to be conditionally independent of its non descendants (children) nodes in the graph.

The conditional probability of a variable specifies the conditional distribution of the variable given its parents. In other words it is the probability of the variable occurring subject to the parents carrying the different possible values. The joint probability of a collection of variables that defines the class label for a record is expressed as the products of the individual conditional probabilities. Thus $P(x_1, x_2, \ldots, x_n) = \Pi_{i=1}^{n} P(x_i|Parents(z_i))$. The classification algorithm returns as output one or more nodes of the network along with a probability for each. Thus the network actually returns class labels and associated probabilities of the prediction.

Bayesian belief networks training is lot more easier when the network structure and the observable variables are available, in which case the conditional probability table values have to be computed, much like the naive bayesian classifier approach. However in the case of the network structure with hidden variables, gradient descent is used to learn values for the conditional probability table. Values are predicted using a weight updation process using the gradient descent strategy to find the local optimum based on the hill climbing approach. Initially random values are chosen and then the weights are gradually updated. To start with the gradient values are computed, then the weights are updated by a small value that takes it towards the solution and the weights are normalized to have values in the range of 0 and 1.

4.6 Classification Based on Neural Networks

Neural networks that simulate the working of neurons or functional units of the human brain inherently support long training times and hence lend extremely well to the issue of classification and model training. A neural network is a collection of input and output units that are connected with associated weights. These weights are updated or adjusted in the process of predicting the type of input records or samples. Neural nets as a result of their inbuilt tolerance to noise and support for performance under non trained data is yet another alternative to the data mining task of classification.

Back propagation is technique in neural network models where training involves a back propagation phase of error computation and weight adjustment to result in correct classification. It performs learning over multi layer feed

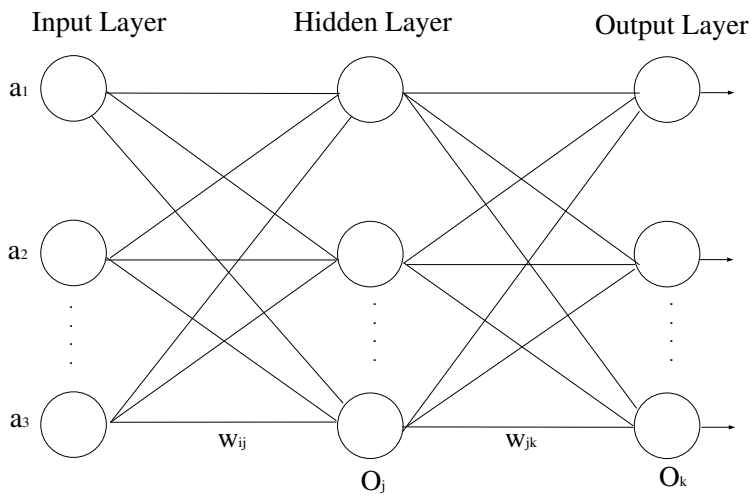

Figure 4.8: An Example Multilayer Feed Forward Neural Network

forward neural networks. A multi layer network is one which is composed of several layers such as the input, hidden and output layers. Inputs related to the attributes of the training data records or samples.

The inputs (training data attribute values) are fed simultaneously to the different units that make up the input layer. Outputs of the input layer units are fed to the hidden layer, which can be more than one depending on the application. Outputs of the hidden layer (last in case of multiple hidden layers) is fed to the output layer which finally yields the model's class prediction. It is a feed forward network in the sense that weights are propagated only in the forward direction from input to hidden to output layers and never fed back to a previous layer in the network. An example multilayer feed forward neural network composed of one input, hidden and output layers is depicted in Figure 4.8.

The neural net structure is best described by the number of units that makeup the input and output layer and the number of hidden layers. Attribute values are best normalized between 0 and 1 and encoded as one unit per domain value. Thus if an attribute has a domain of x_1, x_2, x_3, then the input layer is made up of 3 units, one for each domain value and the units are set(1) or reset depending on the occurrence of the value in the input record. One unit of the output layer can correspond to two classes and cases of multiple classes can associate each unit with a class labels. The number of hidden layers is entirely a learning issue and is best arrived at only after learning and subsequent accuracy evaluation.

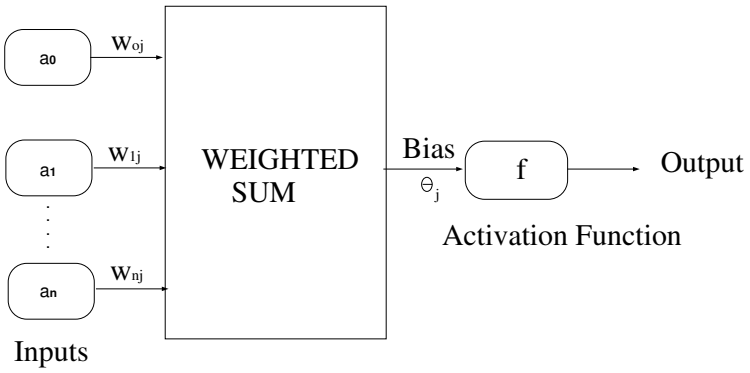

Figure 4.9: Example Neural Network

4.6.1 The Back propagation Algorithm

A back propagation neural net differs from its feed forward counterpart in the fact, that there is a certain degree of error correction involved. Input samples are fed to the network and the network output is compared with the actual class labels. The weights are then adjusted so as to minimize the error between the actual and predicted class labels. Thus weights are back propagated starting from the output to the hidden layer(s) to the input layer in the reverse order of network topology.

The algorithm essentially consists of four major phases namely: (i) Weights initialization, (ii) Propagation of inputs through the network in the forward direction (iii) Error back propagation and finally (iv) terminate the learning when the weights converge. To start with all the connections in the networks have their weights initialized to small starting point values. Next the inputs are fed to the network resulting in outputs similar to the input units ($O_j = I_j$)).

Output to the hidden and output layers are computed as a linear combination of the respective layer's inputs. For the network in Figure 4.9 inputs are the outputs to which a unit is connected in the prior layer multiplied by the corresponding connection weight. For a unit j in the hidden or output layer, input to the respective unit is $I_j = \sum_i w_{ij} O_i + \theta_j$, where w_{ij} is the weight of the connection from unit in the previous layer to unit j, O_i is the output of unit i from the previous layer and θ_j is the bias of the unit which serves as a threshold to alter the unit's activities Every unit in the hidden and output layer applies an activation function to its net input to scale the input onto a smaller domain, in effect normalizing the values. One such function is the squashing function whose output is expressed as $O_j = \frac{1}{1+e^{-I_j}}$. The feed forward part of the algorithm ends here and from now on the network does the tasks of computing error and back propagating it, adjusting weights to result in a further accurate prediction.

Error is sent back by updating the weights and biases in such a way so as

to minimize the error. For a unit j in the output layer, error is computed as $Err_j = O_j(1 - O_j)(C_j - O_j)$, where O_j is the actual output of unit j and C_j is the correct output. Error computation of a hidden layer unit j is nothing but the weighted sum of the errors of the units connected to the unit in the next layer. Thus the error of a hidden layer unit j is $Err_j = O_j(1 - O_j) \sum_k Err_k w_{jk}$, w_{jk} being the connection weight from unit j to a unit k in the next layer and Err_j is the error of unit j. Weights are updated by a quantity Δ which is nothing but the learning rate multiplied by error and output values. Thus $\Delta w_{ij} = lErr_j O_i$ and this change in weight (delta) is added to the current weight. $(w_{ij} = w_{ij} + \Delta w_{ij})$

The learning rate parameter is a constant with values between o and 1 and generally predicts the speed of learning of the network. Too small values result in slow learning while too large values result in fluctuations and hence inaccurate classification. Ideally they are set to 1/number of iterations that have been done over the training data. Weight updation as said before is by the method of gradient descent using hill climbing and the learning rate parameter ensures global optimum is reached. The bias is also updated by a quantity $\Delta\theta_j = l.Err_j$ resulting in new bias $\theta_j = \theta_j + \Delta\theta_j$. Weight and bias updation could either be on a sample wise basis or by the end of every iteration or pass over the entire training data set, also referred to as an epoch. Thus updates are either done on a case or epoch basis. The learning is stopped or terminated when either the epochs threshold is reached or misclassified records are minimal or the weights in the previous iteration are far less than the set threshold. The back propagation based learning algorithm for data classification is shown in Figure 4.10.

Given the complex nature of network connections, weights and biases in the network, neural networks based classification model is generally difficult for human interpretation and are best expressed as rules, to be more specific if then rules. To start with even before the knowledge representation phase, the network is pruned to eliminate those connections that do not adversely affect the classification process when removed. Rules describing relationships between input and hidden layer units are extracted. Also the network is tested for sensitivity of input variables, by varying the values of one input unit while keeping the other network parameters unchanged and measuring the output.

4.7 Evaluation of Classification Models

As discussed earlier accuracy is one of the parameters for evaluating a classifier model. Classification accuracy estimates the likelihood of the classifier correctly generating the class label for unknown test data based on the model that it has learnt from the training database. Two techniques that are commonly used to estimate accuracy are hold and cross validation method. In the hold out method, the given input data is partitioned into two namely training and test, with the training data set composed of 2/3 of the original data and the rest making up the test data set.

In the k fold cross validation method, the input data is randomly partitioned into k mutually exclusive partitions, each of approximately equal size. Training

1. Initialize weights and biases of the network connections.
2. while(stopping criterion not reached)
3. begin
4. for every training database record D
5. begin
6. for every hidden or output layer unit j
7. begin
8. compute $I_j = \sum_i w_{ij} O_i + \theta_j$
9. $O_j = \frac{1}{1+e^{-I_j}}$
10. end
11. for each unit j in the output layer
12. compute $Err_j = O_j(1 - O_j)(C_j - O_j)$
13. for each unit j in the hidden or output layer
14. compute $Err_j = O_j(1 - O_j) \sum_k Err_k w_{jk}$
15. for every weight in the network
16. compute $\Delta w_{ij} = lErr_j O_i$ and $w_{ij} = w_{ij} + \Delta w_{ij}$
17. for every bias in the network
18. compute $\Delta\theta_j = l.Err_j$ and $\theta_j = \theta_j + \Delta\theta_j$
19. end
20. end

Figure 4.10: Classification Learning Algorithm using Back propagation

and testing phase are performed for k iterations and during any specific iteration P_i, partition i constitutes the test data and the rest make up the training data. Thus for iteration 3, partition 3 acts the test data while the other partitions $P_1 \ldots P_n$, excluding P_3 serve as the training data for learning.

Accuracy is estimated as number of samples correctly predicted in k iterations divided by total number of records in the original input data. Another variation to the cross validation ensures that the distribution of records in the various partitions covers all classes, approximately and is referred to as the stratified cross validation technique. Other technique such as sampling with replacement are also employed to estimate classification accuracy.

Two techniques that are generally used to increase classification accuracy of a model are boosting and bagging, which concentrate on merging different possible classifies to result in the optimal classifier. Bagging is based on the principle of majority wins where the class labels that results most from the different classifiers is the one that is chosen for the test data record. For a specific iteration t, a training data T_t is sampled with replacement from the original data. Note that since sampling is with replacement, some of the original data records might not be included in T_t and repetitions may also occur.

A classifier C_t is learned for every training data set T_t and the class which results most from the various learned classifiers is assigned as the class label for the unknown test data record. Boosting extends the concept of bagging by associating weights with the various training partitions. Classifiers are learned and later the weights are updated to improve the mis-mis classification error in subsequent iterations. As before results from the various classifiers to decide the final class, with the weight of the classifier indicating its accuracy.

4.8 Other Measures for Evaluating Classifiers

Some of the other parameters that are used to evaluate classifier are speed, robustness, scalability and readability. Scalability refers to the number of input and output operations made by the classifier on large databases while readability or interpretability could be an objective measure such as number of nodes in the tree, number of neural network layers/layer units. However there are other parameters based on which classification models can be compared and evaluated.

Sensitivity and Specificity are two measures that can contribute to better classifier evaluation. Sensitivity is defined as the ability of the classifier to correctly identify the positives or what are referred to as true positives. With regards to classification and class labels, positives and negatives represent the majority and minority class labelled records or samples. For example in medical classifier application two possible classes could be malignant and non malignant. Here malignant is treated as the positive class and non-malignant as the negative class. This classification of positive and negative samples are entirely from the majority and minority occurrence of the class labels and has got nothing to do with either the literal or medical orientation of the terms.

Sensitivity or the ability of the classifier to correctly classify the positive

samples, is expressed as $\frac{No.\ of\ correctly\ classified\ positive\ samples}{No.\ of\ Positive\ Samples}$. On similar lines specificity is expressed as $\frac{No.\ of\ Correctly\ classified\ negative\ samples}{No.\ of\ negative\ samples}$.Based on these measures, precision or the accuracy of the classifier from the entire database point of view is expressed as shown in Equation 4.3.

$$\frac{No.\ of\ correctly\ classified\ positives}{No.\ of\ correctly\ classified\ positives + No.\ of\ misclassified\ positives} \tag{4.3}$$

The terminology also treats correctly classified positive records as true positives and misclassified positive records as false positives. The overall accuracy of the classifier using the above measures is expressed as in Equation 4.4. Positives refer to the total number of positive samples and negatives relate to the total number of negative samples in the input database. Note that as often is the case assumed with many classifier it might not be always possible to generate unique class labels for every input record. In certain situations it would be ideal to return a probability based class distribution for the input records and then based on user specified thresholds assign the appropriate class labels.

$$accuracy = sensitivity(\frac{positives}{positives + negatives}) + specificity(\frac{negatives}{negatives + positives}) \tag{4.4}$$

4.9 Other Classification Techniques

Data Mining as mentioned in Chapter 1 is a culmination of concepts from several domains ranging from neural networks to statistics. Infact the concept of association rule mining discussed in Chapter 3 can also be used in the process of classification. Association rules as discussed before identify interesting relationships or rules between the various data items. These could be extended to result in class labels as a consequent or right hand side of the rule. Thus a rule of the form $A_1 \wedge A_2 \wedge \ldots A_n \rightarrow Class_i$ could be interpreted as the presence of attributes $A_1, A_2 \& A_3$(along with values/conditions) implying the class label of the sample as i. As before measures such as support and confidence are used to denote the set of records that satisfy the rule antecedent and consequent. As would be the case generally, the number of rules involving similar attributes could be many in number and in such cases the system identifies a class of rules called as possible rules that includes the rule with the highest confidence. Such association rule based classification systems are also referred to as associative classifiers.

Another extension to associative classification is to identify the various rules involving attribute value pairs and class labels and then cluster or group related and similar (adjacent from a plot point of view). Such systems are also referred to as association rule clustering systems. Another group of classifier based on association mining incorporates the concept of emerging pattern. An emerging pattern is one whose support increases gradually with transitions in database. The rate at which the support of such a pattern is referred to as growth rate

of the emerging pattern. The differentiating power of emerging patterns in correctly classifying a record is then used in assigning scores to the classified record labels. The class which has the highest such score is then assigned as the class label for the unknown test data record.

Genetic algorithms, a branch of Artificial Intelligence mimics the concepts of natural evolution in the process of learning. It starts out with an initial population of the solution (if then classification rules) and then uses operators such as selection, crossover and mutation in the process of generating new rules. Selection operator aims to select those rules from the population that are highly fit from the classification accuracy point of view. Crossover attempts to merge two rules resulting in new off springs and then passes them on to higher generations depending on their fitness. Mutation attempts to complement the conditions in the rules resulting in offspring rules which are evaluated for their fitness before passing on to the next generation. The process of learning is continued so far till the resulting population satisfies the fitness threshold or number of generations limitation is reached.

Neighbourhood based classification attempts to generate class labels for records based on neighbours to it using a distance measure such as euclidean distance. The various training database records are mapped onto a n dimensional spatial notation and then for a given unknown record, a set of neighbours that are adjacent to this record is identified. Among the m neighbours identified, the class labels that occurs the most is assigned as the class label for the unknown test data record. Such techniques result in faster learning time compared to decision tree and back propagation based models. However they suffer from the limitation of increased classification time as opposed to the neural network and tree based methods. This is due to the fact that no model is essentially built by the system and for a given unknown record, every time a set of neighbours have to be identified. In a large database, the number of neighbours for a record could be huge resulting in enhanced classification time.

Another technique could employ the concept of case based reasoning in the process of classification. To start with a set of training cases (could be if then classification rules)is created and for a new record (test case), the list of existing cases is compared for similarity. If one of the cases yield a match, then the class label pertaining to the matching case is assigned to the test data record. In situations where an exact match of the base cases is not possible, the nearest neighbourhood approach is used to identify the set of possibly similar cases and the class label of the most frequent case is assigned to the test data record. Such case based methods require the support for efficient matching and indexing of the training cases.

The approach of rough set approximation could also be correlated for classification. It is applicable for discretized attributes and is based on the concept of equivalence class creation. Set of records that are identical with respect to the data describing attributes constitute an equivalence class. Classes which are not identical to the existing set of classes are defined as rough sets and these are approximated with lower and upper bounds.

The lower bound approximation for a given class consists of all those record

that are certain to belong to the class without any doubt based on the characteristics of the data record. The upper bound approximation consists of those records that are certain not to be of the respective class type. The approximations can be represented in the form of if then decision rules and used to classify new test records. Rough set approximation is also used as a feature reduction and relevance analysis technique to eliminate unnecessary and redundant attributes and assess the significance of attributes to the data mining task. The classification systems discussed so far employ a hard and numeric based threshold which might mis-classify records that fall on either ends of the threshold. An alternative to this is to employ fuzzy threshold and boundaries where instead of specific numeric thresholds set membership and associated degrees are represented. It is almost a generic representation of the specific rules mined so far and hence there can always be more than one fuzzy rule that matches the input on hand. In such cases a majority voting is used to resolve the tie.

Summary

The chapter has introduced the readers to the concept of data classification and the various algorithms for the same. Techniques such as decision tree induction, neural network based back propagation and associated classifier have been discussed in depth. Classification forms an essential part of most real time applications with the need of having to identify the possible class to which an (new) input belongs to. Measures to evaluate classification systems such as sensitivity, specificity, accuracy, speed, etc. have been introduced and readers should be able to compare and evaluate the various classification models built for the databases of their choice.

Review Questions

1. Explain the technique of data classification, identifying its objective.
2. Compare and Contrast the performance of decision tree from probabilistic classifiers. (refer external source).
3. Discuss the need for tree pruning in classification systems.
4. Explain bayesian classifiers in detail with a sample database of your choice.
5. Discuss parameters on which classification models are compared and evaluated.
6. Explain associative rule based classification systems in detail.
7. Survey the various classifiers discussed and others in the literature from their application domain, functionality and working and performance point of view. (refer external source).

Chapter 5

Data Clustering Techniques

Clustering, as the name suggests is the technique of grouping related records or samples in databases. A cluster is interpreted as a collection of related or similar records, similarity being a measure of relationships between records. A naive clustering example with respect to the employee databases discussed earlier could be to group the records based on class label, i.e all employees belonging to a particular class treated as one cluster or group. Here similarity measure is correlated to the class labels level similarity or equality to be more precise.

Going along the lines of the above example, clustering could look similar to the data classification approach of data mining. However clustering differs from classification in the aspect of underlying learning mechanism used. Classification is more of a supervised learning approach where training data with class label information is available for the system to come up with a model that is to be put to use for classifying the records in the test database.

However clustering is an unsupervised learning technique, in the fact that there is no explicit demarcation of data as training and test data. There is only one generic database available and class labels information are not available. Thus the input database ideally is composed of attributes and the objective of clustering is to group related or similar records in the database without any background information on class labels or other information. Classification could be viewed as a case of learning by examples while clustering is the case of learning by observation (of similarity among data records).

Clustering aims to group related records by measuring similarity among the attributes or characteristics of the input database tuples or samples. Thus a key phase of any clustering technique is similarity measurement and there are a host of clustering techniques differing in the mode of operation and similarity measure. The chapter provides an indepth discussion of the different types of clustering and algorithms for the same.

5.1 Introduction to Data Clustering

Clustering as has been discussed shortly is the process of grouping a set of records or objects from the real life point of view. The various groups available at the end of the process are referred to as a cluster, a cluster being a collection of related or similar records. Remember it is a collection of similar and related records and hardly is the case that they contain equal records. Equality of attribute values which can be possibility in real life databases can hardly be a criterion for measuring similarity of records. Clustering principally aims at grouping records that are characteristically same(similar). In fact most often it is similarity than sameness of records that is sought for in clustering.

Clustering works on the principle of maximising intra cluster and minimizing inter cluster similarity. The principle optimizes the overall function of the clustering process. Intra cluster similarity criterion ensures that objects or records that are placed in a group or cluster have the highest similarity, while inter cluster similarity ensures that objects or records within a cluster have the least similarity with objects or records of another cluster. Thus the process of clustering attempts to group records ensuring the condition that data within a cluster are most similar while data across clusters are the least similar.

Applications of clustering are varied to the extent of data mining's wide application domain. It finds application in business management in the form of identifying customer groups based on purchasing patterns and other characteristics. It is used in geographical information systems and data analysis in determining land areas that are similar and grouping houses in cities based on the construction type. It also finds extensive application in web data mining by grouping together related documents on the web and hence contribute to efficient and intelligent information retrieval strategies. Clustering could also be viewed as a preprocessing technique prior to classification and other data mining techniques by discovering patterns or knowledge from the established clusters. In such cases it acts a data characterization technique that analyses the input database to suit further mining operations.

Clustering encompasses techniques from statistics, machine learning, spatial technology, etc. However statistics has been a profound contributor to the area of clustering with techniques such as k-means, mediods clustering, etc. Clustering could be based on distance similarity measure oriented such as k means or be driven by the overall concept. In the case of conceptual clustering, records are grouped only if they are describable by a concept while distance based methods groups records with minimal distance difference. As in classification, clustering techniques need to provide support for scalability, complex data types and shapes, high dimensionality and numeric cum categorical attribute support. Some of the key performance and research issues related to clustering are identified below.

1. Clustering techniques should be scalable. Scalability is an important characteristic of any database application and data mining is no different either. The clustering techniques must be capable of discovering clusters

for varying database sizes in the increasing scale and discover them in a reasonable amount of time.

2. Clustering is more a subjective than objective process in relation to other data mining techniques. Clustering algorithms require user inputs such as optimal number of clusters, similarity measure to be used, etc. However the amount of such information required by a clustering technique must be minimal to the extent possible. The complimentary situation only restricts the scope of the system and is also susceptible to inconsistent pattern discovery as a result of parameter variations.

3. The clustering technique should in no way be dependent on the order of the input database. Thus a database when subject to clustering and then later reordered (change the order of appearance of input records, say the trailing half of the original database now becomes the leading half) and clustered should result in the same knowledge or set of clusters. Clustering aims to group data based on characteristic similarity and not physical traits such as order of occurrence in the input database.

4. As with most data mining techniques, clustering should be capable of handling noisy data. Support for noise management is all the more essential in clustering for it aims to observe characteristic similarities and a method that eliminates noise only helps in optimal clustering. Otherwise the noisy data could as well be treated as an essential characteristic and constitute cluster(s) which should not be the case.

5. Extensive support for handling input database of high dimensions is a must. Remember data mining as a technique makes it presence felt only when the size of the data being processes is huge, which otherwise can always be challenged by human analysis. Clustering algorithms must also be capable of handling both numeric as well as categorical attributes. Support for condition or constraint based clustering is also essential. Finally the discovered cluster knowledge should be user readable and interpretable resulting in efficient applications of them that benefit the user, the sole purpose of the data mining system.

Further sections to follow in the chapter will primarily address clustering from the overall algorithm's operational point of view, types of data that have to be handled and the similarity measures to be used.

5.2 Data Types in Clustering

Clustering which principally aims at grouping related and characteristically similar data requires efficient data representation and formats. Two commonly used data representation formats are the data and dissimilarity matrix notation. The data matrix notation typically is matrix of m*n order, m being the number of records or objects and n being the number of attributes within a record, also

referred to as measurements or variables in clustering terminology. An example representation of data in matrix format could be as shown below.

$$\begin{bmatrix} a_{11} & \cdots & a_{1k} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{21} & \cdots & a_{2k} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & \cdots & a_{mk} & \cdots & a_{mn} \end{bmatrix}$$

Another alternate data representation format is the dissimilarity matrix notation where similarity/dissimilarity information among pair of objects or data is stored in a matrix format. It is represented as a n*n matrix with the first row representing the similarity between the first data and itself, the second row representing the similarity between the second data element and the first and itself and so on on a cumulative itself. The matrix shown below represents the dissimilarity between pairs of data element, a value of 0 denoting exact similarity and greater positive numbers denoting the scale of dissimilarity. This method of matrix notation is also referred to as the two mode matrix where row and columns relate to the two modes.

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ d(4,1) & \vdots & \vdots & 0 & \\ \vdots & \vdots & \vdots & \vdots & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

The above matrix notation d(i,j) represents the measured difference or similarity between data elements or objects i and j. The matrix is also referred to as the one mode matrix where both rows and columns represent the same mode (difference measure) as opposed to the two modes stored in the data matrix notation. Dissimilarity notation often tend be more condensed and widely used data formats in clustering techniques. The entire crux about clustering is thus measuring similarity between data elements and similarity computation is primarily dictated by the type of the data elements. The following section discusses the different types of variables (data elements) that can be encountered in databases and methods to compute similarly among them.

5.3 Variable Types & Similarity Computation

5.3.1 Binary Variables

Binary variables are those which can carry two possible values of 0 or 1 denoting the presence or absence of the attribute. Thus a variable such as smoke which indicates whether a person smokes or not can be represented by states of 1 or 0

Data Element j				
		1	0	Sum
Data Element i	1	a	b	a+b
	0	c	d	c+d
	Sum	a+c	b+d	T

Table 5.1: Binary Variables Representation

for smoker/non smoker. A generic representation of binary variables data of two data objects or data elements, assuming all have equal weightage is as shown in Table 5.1. In the table, a represents the presence of both data elements i and j is the number of variables that contain value 1 for both data element or objects i and j. On similar b represents the number of variables that equal 1 for object i and 0 for object j and so on. The total number of variables represented by the table, also referred to as contingency table is T=a+b+c+d.

Binary variables are infact further classified as being symmetric and asymmetric. A symmetric binary variable is one where both states of 0 and 1 are equally important and carry the same amount of information or emphasis. Thus with symmetric binary variables, there is no priority of a 0 value over 1.

Variables that reflect gender of human beings are possible candidates for symmetric binary variables, because both are equally possible and a fair society does not give preference of 1 (say male) over 0(female). Similarity measure based on symmetric binary variables is referred to as invariant, interpreting that the measure does not vary with the coding of the variable. Invariant similarity between two objects i and j is defined as $\frac{b+c}{T}$.

On the other hand binary variables where outcome of the states are not equally important as is the case with most medicine related application variables. A variable such as cancerous that indicates whether a person tests positive for cancer could require the more important outcome, namely cancerous by 1 (positive for cancer) and the normal one by 0(negative). Thus with asymmetric variables, the situation of a positive match (both variables are 1) is more significant than the case of a negative match (both are 0). Similarity measure based on such asymmetric variables is referred to as invariant similarity and is expressed as $\frac{b+c}{a+b+c}$. Note that the effect of state d is neglected in the similarity computation.

5.4 Interval Representation Variables

Interval representation variables are continuous measurements made on a nearly linear scale. Examples of such variables include height, weight, etc. Similarity between such variables generally uses distance measures such as euclidean, manhattan, etc. Such variables and their measurement units can directly impact the entire clustering technique. Height representation in metres, centimetres, etc. can result in varying cluster compositions. A smaller representation such as cm

could result in large range compared to a higher or greater representation such as metre and hence can impact the clustering structure.

Interval representation and scaled variables result in a standardized notation for such situations and alleviates any knowledge about the data. Standardization is generally performed by adopting a unitless notation and consists of the following steps namely (i) mean absolute deviation computation and (ii)standardized measurement. For a variable v, the mean absolute deviation is expressed $s_v = \frac{|a_{1v}-m_v|+|a_{2v}-m_v|+\cdots+|a_{nv}-m_v|}{n}$, where $a_{1v} \ldots a_{nv}$ are the n measurements for variable v and m_v is the mean value of v. The standardized measurement or z score of a variable v for value i is expressed as $z_{iv} = \frac{a_{iv}-m_v}{s_v}$. Note that deviations from the mean are not squared to minimize the effect of outliers.

Once data is standardized or for that matter even non standardized data, the next immediate requirement would be to compute the similarity between data objects and for interval scaled variables, similarity is generally computed based on distance measures such as euclidean measure defined as in Equation 5.1.

$$d(i,j) = \sqrt{|a_{i1} - a_{j1}|^2 + |a_{i2} - a_{j2}|^2 + \cdots + |a_{in} - a_{jn}|^2} \tag{5.1}$$

, where i and j are the two n dimensional objects or records or database elements.

The manhattan or city block distance similarity metric or measure is expressed as in Equation 5.2.

$$d(i,j) = |a_{i1} - a_{j1}| + |a_{i2} - a_{j2}| + \cdots + |a_{in} - a_{jn}| \tag{5.2}$$

Similarity measures defined by euclidean and manhattan distances satisfy the conditions that (i) distance between an object and itself is 0, distances are symmetric, in the fact that d(i,j)=d(j,i). Also the distance function results in a positive number and triangular inequality that $d(i,j) \leq d(i,h) + d(h,j)$ should be satisfied. The triangular inequality states that reaching an object j from i should not exceed the distance when reached via an intermediate object or point h. A generalization of euclidean and manhattan distances is the minkowski distance and is expressed as shown in Equation 5.3.

$$d(i,j) = \left(|a_{i1} - a_{j1}|^k + |a_{i2} - a_{j2}|^k + \cdots + |a_{in} - a_{jn}|^k\right) \tag{5.3}$$

Weighted euclidean distance measure is nothing but euclidean expressed in Equation 5.1 with each term multiplied by the appropriate weights. Weighted euclidean measure is expressed as shown in Equation 5.4.

$$d(i,j) = \sqrt{w_1|a_{i1} - a_{j1}|^2 + w_2|a_{i2} - a_{j2}|^2 + \cdots + w_n|a_{in} - a_{jn}|^2} \tag{5.4}$$

5.5 Other Variable Types

This section discusses the similarity measure computation for variables other than the ones discussed in the above sections. Nominal variables are those which

can be in more than 2 states. In a way its a generic form of a binary variable, however differing from binary variable that it can take on more than two states. Examples of nominal variables include color, shape, etc. Dissimilarity between any two nominally described variables i and j is expressed as $d(i,j) = \frac{p-m}{p}$, where m denotes the number of matches (similarity between i and j) and p is the total number of variables. Variations such as weighted nominal measure can be used with weight scalings associated with matches. Nominal variables can be expressed using asymmetric binary variables with a variable being created for each possible value of the nominal variable. Thus for a color state blue, a binary variable is created and value set to 1 while other binary variable's states are set to 0. Similarity computation is similar to the one discussed for binary variables.

Ordinal variables are similar to nominal except for the fact there is a certain degree of ordering or meaning incorporated in such variables. Variables that reflect the designation of an employee such as Foreman, Manager, General Manager etc. appear in a sequential order starting from the lowest one in the hierarchy to the top most. Such variables are infact referred to as discrete ordinal variables.

Continuous ordinal variables are the ones where it is only the ordering that is essential and not the magnitude. Examples of such variable are medal placings in various events where it is only the ordering such as gold, silver, etc. that is important as opposed to their values. Similarity measure computation is similar to interval scaled variables and can be computed using one of the distance based methods. However since an ordinal variable can possibly take v states, standardization requires that these be normalized within the [0,1] range. Initially the ordinal variable with s states is mapped onto a ranking scheme, where every possible state is assigned a rank. Then rank r_{iv} correlating to value i for variable v, the z score is computed as $z_{iv} = \frac{r_{iv}-1}{s-1}$.

Variables where values tend to rise or change in an exponential or on a non linear scale are generally referred to as ratio scaled variables. Such variables are best handled by treating them as interval scaled variables with an intermediate log scale transformation to ensure that the scales and ranges are not distorted.

Real life situations often contain a mixture of the above discussed type of variables and the clustering technique must be capable of handling them in an integrated fashion. In such cases the similarity measure is expressed as $d(i,j) = \frac{\sum_{v=1}^{n} \delta_{ij}^{v} d_{ij}^{v}}{\sum_{v=1}^{n} \delta_{ij}^{v}}$. The delta term indicates the variables and is set to 0 for missing values and asymmetric binary variables with both states carrying values 0, otherwise it is set to 1. If v is binary or nominal variable, then d_{ij}^{v} is set to 0 for equal values of i and j objects ($a_{iv} = a_{jv}$) and 1 otherwise. Interval, ratio and ordinal scaled variables are handled using measures discussed earlier.

5.6 Clustering Techniques

Clustering algorithms differ not only on the data type that they handle but also in the overall function and operation point of view. Clustering techniques are basically classified as partition based, hierarchical, density based, grid based and model based methods. This section discusses the different types of clustering techniques and sections to follow will detail on the various algorithm under each category.

1. **Partition based Methods**: These methods are based on the overall principle of dividing databases into partitions satisfying the criterion that objects within a partition are similar to the extent possible. Infact on a rough scale, partitions can be equated to the number of clusters. The input database is divided into k groups ensuring that a record placed in any one of the group does not duplicate and a group contains atleast one record or object. The partitions are further fine tuned (iteratively partitioned) to optimize the basic tenet of clustering, whereby object of least similarity are moved to other groups or partitions or clusters retaining the highly similar ones. Clusters are generally represented either by the mean value of the objects in them (k means methodology) or by the object or record that is closer to the cluster center (k medoids).

2. **Hierarchical Methods**: These methods are based on decomposition of data objects either in the forward or reverse direction. Two types are agglomerative and divisive. In agglomerative clustering, the decomposition is done in a bottom up fashion where every object or record to start with constitutes a cluster. Later clusters are merged to optimize the basic tenet of clustering. This merging can go on till the maximum cluster number being 1, the most optimal situation where all records are of the same cluster type. With divisive clustering, decomposition happens in the top down fashion, where to start with the entire database constitutes a cluster. Later clusters are split to optimize the principle of clustering.

3. **Density based Methods**: Such methods employ the number of data objects or density of a cluster as the measure opposed to distance metric adopted by partitioning methods. A given cluster is grown till the neighbourhood points or object's number exceeds a specified threshold.

4. **Grid Methods**: These methods require the data to be represented in spatial format where the object space is quantized to a finite number of cells forming a grid. Clustering is performed on the grid and such techniques support faster processing time as computation is based only on the number of cells in a dimension and not the number of objects or records.

5. **Model based Methods**: Model based methods suppose a model for the various clusters and then find the best fit for the data with the given

1. Initially a random set of k objects are chosen as the cluster centres.
2. do
3. Compute the mean value of the objects in a cluster. Initially each object represents the cluster mean or center.
4. Place each object in the cluster which exhibits the maximum similarity.
5. Update the new cluster means as a result of an object placement.
6. until there is no change in the cluster structure.

Figure 5.1: k-means Clustering Algorithm

model. Density functions relating to spatial distribution of objects are generally used to move clusters.

5.7 Partitioning Methods

5.7.1 k-means Clustering

Partitioning methods function based on the overall principle of grouping the input database objects into k partitions, each partition representing a cluster. Clusters are created optimizing a parameter such as cluster distance. The distance based similarity measure ensures that similar objects or records are placed within a cluster while dissimilar are organized in different clusters. One of the popular and widely used clustering technique based on distance measure computation is the k-means clustering algorithm. The algorithm is in Figure 5.1. The k means clustering technique is a centroid based approach that takes as input the number of partitions k and creates k clusters or partitions of the input database consisting of n objects or records, optimizing the principle of clustering that the resulting cluster's intra cluster similarity is high and inter cluster similarity is low.

To start with k of the objects or records from the input database are selected in a random order. Each object initially represents the cluster mean or center. The remaining objects are then assigned to clusters which offer the maximal similarity, in terms of distance from the computed cluster centres or mean of objects in the cluster. Once an object is clustered or placed in a cluster, the new mean of the cluster is updated. This process is repeated until the results of clustering converge or the optimal number of points within a cluster is reached.

Expressing mathematically, the error involved in the clustering task is expressed as $E = \sum_i^k \sum_{p \in C_i} |p - m_i|^2$, where m_i is the mean of cluster C_i, E is the sum of squared error for all objects in the database and p is the point that

represents an object. An optimizing criterion for clustering could be to keep the clusters compact and distinct.

The k means clustering approach scales well and handles large databases efficiently with the time complexity of the algorithm being O(nkt), where n is the number of database objects or records, k is the number of clusters or partitions and t is the number of iterations. It suits clusters that are compact in nature and well separated from other clusters. Limitations of the k means clustering technique is the fact that the number of partitions might not be realistic possibility.

Cluster means might not always computable and in such cases the algorithm might not be suitable. Also k means clustering is not suited for clusters of varying sizes and support for noise and outlier management is very minimal. Noise and outliers could themselves get treated as data and hence affect the mean value of a cluster or cluster center and hence directly impact the resulting cluster structures.

Variations of k means clustering such as similarity computation, initial k means selection and cluster mean computation strategies, etc. It is more often best to adopt an agglomeration approach that determines the overall number of clusters and hence progress to the actual phase of clustering for greater consistency and accuracy.

The k modes method replaces cluster means with cluster modes for categorical attributes . The technique employs new similarity measures to handle categorical attributes and a frequency oriented method to update the cluster modes. An integrated method that combines both k means and k modes can also be employed for real time clustering.

Another extension to the k means clustering approach would be to assign objects to clusters based on a weight that indicates the probability of the object belonging to the respective cluster. In such cases the cluster means are computed using weighted measures. This eliminates hard cluster boundaries wherein an object is not assigned to a dedicated cluster, but to a cluster associated with a probability measure. Scalability of k means clustering algorithm is ensured by identifying discardable, compressible and to be maintained in the main memory regions.

An object that is guaranteed to be a member of a cluster is discarded (its membership is sure). Objects whose membership is not guaranteed but belong to a tighter subcluster are referred to as compressible. Other objects are treated as to be retained ones. The clustering strategy concentrates only on compressible and to be retained in the main memory objects . This way the algorithm can be alleviated the bottleneck of having to interact with the secondary memory. Now data that are required is maintained in the main memory itself and hence can scale well with increasing database sizes.

5.7.2 k-medoids Clustering

The k means clustering algorithm suffers from the limitation of not responding well to outliers and noisy data that could drastically alter the structuring of

clusters. The k medoids approach helps in eliminating this sensitivity by using medoid as a measure for similarity computation. Thus instead of choosing the mean value of objects in a cluster, the most centrally located object within a cluster. Thus the reference point is alone changed while retaining the distance based similarity measure computation.

Similar to k means clustering, initially k objects are chosen randomly as medoid representatives of the cluster. The remaining objects are assigned to the cluster to which it is most similar based on the medoid distances. Higher iterations have the cluster medoids replaced by non medoids and this process is iterated until the requisite clustering structure is achieved. The medoid replacement strategy is governed by the following conditions. A nonmedoid object (nm) is chosen as a replacement for a current medoid to improve clustering quality only if there is an improvement in the placing of objects within clusters in terms of distance between the object and the earlier and the new medoid. Every time a medoid changes, the squared error value between the original and replaced medoid placement is computed and this is summed up over all the nonmedoid replacements. If the overall cost is negative then the current medoid is swapped with the nonmedoid while otherwise the current medoid is acceptable in terms of minimizing the error.

PAM is one such medoid based clustering strategy and is referred to as the Partitioning Around Medoid and aims to create k partitions similar to k means clustering algorithm, with the difference that clustering is based on medoids rather than mean value of objects in the cluster. An initial set of k random objects are chosen as the medoids and the technique attempts to fine tune the cluster by choosing better medoids if present. A non medoid object replaces an existing medoid object only if it brings down the error cost significantly, otherwise the medoids remain unchanged. Similar in nature to k means technique, the medoids approach performs well in the presence of noise and outlier data which have less impact on medoids compared to means.

5.7.3 Other Partitioning Methods

Despite the improvements offered in terms of resistance to noise and outliers, k medoids approach much like k means clustering does not scale well to increasing data sizes. A sampling based method, CLARA or Clustering Large Applications responds well to the scalability issue. CLARA initially selects a sample of the original data and it is over this that the clustering based on medoids or PAM is performed. The selection process should be random to represent the original database. Sampling is done multiple times and PAM is applied over each sample, returning the best cluster structure as the output.

Performance of CLARA is dependent on the selection of the correct medoid in a sample as that from the original data by PAM. Thus to promote efficiency, the CLARA technique has been extended to support randomized search. The application called as Clustering Large Applications based on Randomized Search draws samples with a degree of randomness. A graph of the data points results in every node being treated as a potential solution. Cluster structure resulting

from medoid replacement is referred to as neighbour of the current clusters and the process iterates for the best neighbourhood search. If a better point is not identified, then the process is repeated with a newly selected set of nodes. Outlier detection is possible with CLARANS. Complexity of CLARANS is $O(n^2)$ as compared to $O(ks^2 + k(n-k))$, s being the sample size, k equating to the number of clusters and n being the total number of objects or records in the database.

5.8 Hierarchical Clustering Techniques

Hierarchical clustering methods operate based on the principle of decomposing databases either in a top down or bottom up fashion, resulting in what are called as divisive and agglomeration techniques. In agglomerative clustering, every individual object is treated as constituting a cluster initially. Later clusters are merged based on similarity of objects within them until a single cluster is reached, the optimal case of clustering where all objects in the database are of the same cluster type or a terminating condition is reached.

Divisive techniques operate in the reverse direction or bottom up and to start with treats the entire database as a single cluster. Cluster(s) are split based on similarity, to be more precise dissimilarity measure until every objects becomes a individual cluster or till a terminating condition such as number of clusters is reached. Clustering applications such as AGNES and DIANA are examples of agglomeration and divisive clustering techniques. Hierarchical measures generally employ distance measures such as minimum distance, maximum distance, mean distance and average distance in the process of similarity computation. Given $p - p'$ represents the distance between two points, m_i the mean for cluster C_i and n_i the number of objects in C_i, minimum distance is expressed as in Equation 5.5.

$$d_{min}(C_i, C_j) = min_{p \in C_i, p' \in C_j} |p - p'| \tag{5.5}$$

Maximum distance computes the maximum of the distance between the points. Mean distance is expressed as $d_{mean}(C_i, C_j) = |m_i - m_j|$ and average distance between two points is expressed as $d_{avg}(C_i, C_j) = \frac{\sum_{p \in C_i} \sum_{p' \in C_j} |p-p'|}{n_i n_j}$. Agglomeration methods suffer from the limitation that the entire procedure is heavily dependent on merge and split procedures and further clustering is based on the previously merged or split clusters. The issue is that once a merge or split is done, its effect cannot be undone and hence its effects cascade throughout the clustering process. Also in terms of scalability, as a result of evaluation of objects for the merge and split decision does not respond well for increasing database sizes.

An alternative suggested is to integrate partitioning and hierarchical methods resulting in multiple phase clustering techniques. A few of such hybrid clustering techniques are discussed in the sections to follow.

5.8.1 BIRCH Clustering Technique

BIRCH, Balanced Iterative Reducing and Clustering using Hierarchies is a hybrid technique that supports incremental and dynamic clustering of objects. It maintains a cluster feature tree to summarize cluster representations. For a N d dimensional database, cluster feature contains the values N, linear sum of the N points and square sum of the data points, N being the number of points in a sub cluster. A cluster feature tree is nothing but a height balanced that stores the cluster features at the various hierarchical levels of clustering. Internal nodes in the tree store cluster features of descendants. Branching factor controls the maximum number of children per node while a threshold parameter specifies the maximum diameter of the sub clusters stored at the nodes. Two major phases employed in BIRCH are scanning of the database to create an in memory cluster feature tree as a compressed data representation format retaining the cluster structures and a subsequent clustering technique at the leaf nodes.

New (incremental) objects are inserted into the tree at the leaf level to the closest sub-cluster. Cases where a sub-clusters diameter exceeds the threshold, nodes are split further after which the child information are passed back to the root node Cases of main memory constrained situations are handled by modifying the threshold parameter, building the tree in an incremental fashion at the leaf level. A clustering technique based on partitioning is then applied over the CF tree structure resulting in an $O(n)$ complexity algorithm for a database of n objects. It responds well to scalability issue but is dependent on the distribution of data points in a cluster and its shape.

5.8.2 CURE Clustering Technique

CURE or Clustering using Representatives alleviates the problem of BIRCH favouring spherical shaped clusters and is also better resistant to outliers. A fixed number of representative points as opposed to a single object are chosen in a scattered manner and later shrunk by a shrinking factor. Any two clusters with the closest pair of representative points are merged. The shrinking phase helps in eliminating the impact of outliers while the n representative points accommodate non spherical shaped clusters. CURE samples random partitions which are later partially clustered during the first scan. During the second scan, the partial clusters are further clustered to result in the final cluster structures. CURE consists of the following steps:

1. A random sample S is drawn from the original database.

2. S is partitioned further into a set of partitions.

3. Every partition is partially clustered.

4. Outliers are eliminated by random sampling, where clusters which do not grow are deleted.

5. The partial clusters represented by the n representative points are then clustered by moving towards the center of the final cluster structure. Representative points in newly formed cluster are shrunk and the resulting points denote the final cluster structure.

5.9 Density Based Clustering Techniques

Density based clustering algorithms creates regions with sufficiently high density and discovers clusters of arbitrary shapes. With respect to density based clustering approaches, the neighbourhood within a radius ϵ of an object is referred to its ϵ neighbourhood . An object whose ϵ neighbourhood contains a minimum number of points is referred to as a core object. Core objects help in determining density reachable objects.

A density based cluster is nothing but a set of density connected objects which are maximal with respect to density reachability. The density based clustering algorithm DBSCAN, looks for clusters by checking the ϵ neighbourhood of a point in the database. Whenever the ϵ neighbourhood of an object p contains more than the minimum number of points, then a new cluster with p as the core object is created. It then iterates to collect density reachable objects from the core objects and terminates when no new point can be added to the cluster.

The density based clustering approach discussed, DBSCAN requires a lot of parameters to be specified by the user such as ϵ and MinPts, etc. and such values specification for real time data could be extremely difficult and become a major bottleneck of the algorithm. This is overcome by another density based clustering approach called OPTICS, Ordering Points To Identify Cluster Structure which mandates every object to have two values namely core distance and reachability distance.

The core distance of an object is the smallest value that makes it a core object. For non core objects, core distance is not defined. Similarly reachability distance of an object a with respect to another object b is the greater value of the core distance of a and the euclidean distance between a and b. If a is not a core object, then its reachability distance is again undefined.

The OPTICS algorithm aims at creating augmented cluster ordering to facilitate interactive cluster analysis. The algorithm selects an object that is density reachable with respect to the lowest ϵ value so that clusters of high density are processed first. Clusters are extracted based on the above ordering information. OPTICS runs in o(nlgn) time, similar to DBSCAN.

Another density clustering technique is DENCLUE which is based on a collection of density functions. The effect of data points are modelled as functions, referred to as influence function which describes the effect of a data point within its neighbourhood. Density of data space is expressed as sum of the influence functions of all the data points in space and finally clusters are determined by establishing density attractors, which are nothing but local maxima of the overall density function.

The main benefit of denclue is that a strong mathematical model forms the basis of the clustering approach and can also accommodate other clustering methods such as partitions based, hierarchical, etc. This approach also performs well over data sets in the presence of noise and requires proper selection of density and noise threshold parameters.

5.10 Grid Based Clustering Methods

Grid based approaches offer benefits of fast processing as a result of clustering being performed on grid structure as opposed to the original data based clustering adopted by other techniques. Data are quantized into a finite number of cells forming a grid structure and it is this structure that is subject to further clustering operations. A few of the existing grid based approaches are STING that uses statistical properties of grid cells, WaveCluster that employs wavelet transform method and CLIQUE which represents a grid and density based approach for clustering in high dimensional data.

5.10.1 STING Clustering

STING or Statistical Information Grid uses statistical information stored in cells to generate clusters. It is a multiresolution technique which divides the spatial area into rectangular cells which are maintained at several levels resulting in a hierarchical structure. Every cell at a specific level is further partitioned into a collection of more cells at the next lower level. Properties such as mean, maximum are stored in the various cells and the clustering process is based on these statistical properties.

Parameters such as count which are attribute independent and mean , maximum, minimum, distribution type which are attribute dependent are maintained. Attributes of higher layer cells are computed from lower layer cell's attributes, for which computations are done from the original database. For higher layer cell's distribution, the frequently occurring distributions of the lower layer cells are selected. Distributions are either hand coded if known or otherwise determined by a hypothesis test.

A cell's relevance to the query on hand is determined using confidence interval and estimated range probability computation. Cells that are not related are eliminated and processing starts with the next layer of cells. This is iterated till the bottom most layer is reached. STING offers benefits of parallel processing and incremental updating and query independence in the fact that only summary information of data are stored in the grids.

The grid construction process requires a database scan resulting in a complexity of $O(n)$ and the query processing phase requires $O(m)$Q time, where m is the total number of cells at the lowest level of the hierarchical representation. Performance of the algorithm is very much dependent on the granularity of the lowest level cells, which when too fine results in increased processing time and too coarse results in cluster quality reduction. Another limitation of sting is

that spatial relationships between cells are not considered and hence resulting clusters quality and accuracy is reduced.

5.10.2 WaveCluster based Clustering Technique

It is yet another multi resolution clustering technique that summarizes data on a grid structure and later transforms the original feature space to find dense regions using a wavelet transform method. Every grid cell contains information of a collection of points that relate to the cell which is further used in subsequent transform and cluster analysis. A wavelet transform is a technique that decomposes signals into different frequency sub-bands. The transform procedure maintains the relative distance between objects at different resolution levels.

Wavelet transforms help clustering significant and dense regions and at the same time alleviate the effect of noise and information outside cluster boundaries. In the original data space, dense regions serve as attractors for the clustering process while as inhibitors for deviating points. The technique helps in identifying related and outlier regions in the data automatically. Multi resolution support helps in discovering clusters at varying levels of accuracy.

Wavelet Transform supports parallel processing and runs in $O(n)$ time where n is the number of objects in the database. It can handle large databases, discover arbitrary shaped clusters and support for outlier elimination. Also the performance is not dependent on the input order and user parameters such as number of clusters, radius, etc. The technique achieves better efficiency and cluster quality in relation to BIRCH, CLARANS and DBSCAN clustering methodologies.

5.10.3 CLIQUE Clustering Algorithm

CLIQUE is a clustering algorithm for high dimensional data integrating density and grid based clustering techniques. CLIQUE expands as Clustering in Quest and aims at identifying crowded areas in the data space. The original data is partitioned into non overlapping rectangular blocks and dense units in them are identified. A dense unit is one where the number of data points exceeds a specified threshold.

Subspaces relating to dense units are intersected to determine higher dimensional dense units. This is based on the anti monotone property of set theory which forms the basis of Apriori association mining algorithm, and for that matter most level wise techniques. Candidate dense units at level k are generated from dense units at level k-1 employing the anti monotone property of set theory discussed earlier. A minimal cover or description for each cluster is created. This identifies the maximal region that covers the cluster of connected dense units. It scales well to increasing database sizes and is insensitive to the order of data inputs.

5.11 Model Based Clustering Methods

Clustering based on models concentrates on matching or fitting the input data to the base mathematical model. Data are assumed to have been generated using several probability distributions. Two types of model driven clustering are statistical and neural networks based. The statistical approach employs conceptual clustering while neural networks model employs competitive learning mechanism.

5.11.1 Statistical Model Clustering

Conceptual clustering forms the base for statistical approaches where in addition to grouping similar or related objects, characteristics of the clusters are also generated. In effect conceptual clustering first employs clustering followed by characterization. Clustering is thus more features or concepts or classes oriented compared to being determined by the individual records. Probability measurements are generally employed in conceptual clustering systems and COBWEB is one such application which creates hierarchical clusters in the form of a classification tree.

A classification tree differs from a decision tree in the fact that it is composed of concepts as internal nodes whereas decision trees are generally composed of attributes and possible values. Here internal nodes represent concepts and associated probability of occurrence. Concept probabilities governed by attribute value conditions or conditional probability of the concept to occur are also included in the classification tree. Every child of a classification tree forms a partition and clustering involves a tree traversal identifying the best path in the classification tree.

Classification trees are evaluated on the basis of category utility which is the number of attribute values that can be correctly predicted given an partition. It enhances intra class similarity and minimizes inter class similarity and hence succeeds in identifying objects similar or contrasting to a concept. A given unknown object is momentarily placed in a node of the classification and the category utility of the partition as a result of the placement is computed.

Partitions that result in high utility values are likely to be good candidates for the object. For objects which cannot be accommodate by any of the tree nodes, category utility of partition as result of addition of new nodes to the tree. Depending on the partitions utility values, the object is placed in either the existing class or the new class. Merge and split operations allow to fine tune a partition and result in efficient classification of objects.

COBWEB suffers from the limitation of probabilistic independence assumption. It assumes that distribution of attributes are independent of one another and this might not be realistic assumption in real life databases, where attribute values are correlated. The tree height is not balanced and hence for attributes with large number of values, complexity of the algorithm degrades. Other conceptual clustering based applications are CLASSIT that supports incremental learning feature and AutoClass that uses bayesian concepts in the process of

clustering.

5.11.2 Neural Net Model based Clustering

Neural modes represent clusters by prototypes and new objects are assigned to clusters with maximum prototype similarity. Attributes of an object accommodated within a cluster can be predicted using the prototype. Two major types of neural net based cluster models are competitive learning and self organizing feature map based ones.

Competitive learning systems are based on the principle of winner take all situation for the object that is subject to the system. Winning units within a cluster are activated while others are inactivated. Units in a particular layer can receive inputs from all units in the lower layer and active units in a layer forms the input to the next layer. For an input from a layer below, units within a layer vie for a match. Inhibitors are in place to ensure that atmost only one unit is active within a layer. The winner then updates it weights in relation to other units so that future similar objects will have even stronger responses from it. Number of clusters and units per cluster are input parameters of the technique.

With self organizing maps, units with weight vector closer to the object are chosen as the winning units. Weights are adjusted as before to move closer to the object. The technique assumes that there is no ordering of inputs. The collection of units constitute a feature map and handles high dimensional data well.

Summary

Clustering is the data mining technique of grouping related records together and the chapter discusses the various cluster mining algorithms. Different types of clustering such as density driven, partition based, etc. were introduced and clustering applications based on each of these techniques has been covered in depth. The readers should now be able to appreciate the different types of clustering, applications and the extent of statistics involved. Data mining techniques in general and clustering to be more specific has its base in statistics. The very nature of the operation to look for related objects requires similarity measures. Clustering much like Classification is an important data mining technique that finds extensive application in real life data.

Review Questions

1. Discuss the important issues to be addressed by a data clustering system.
2. Differentiate Clustering from Data Classification mining technique.
3. Implement the k-means clustering algorithm in a programming language of your choice.

4. Explain the different types of clustering methodologies in detail.
5. Discuss the different types of data variables involved in clustering.
6. Compare and Contrast the various clustering methodologies. (refer external source)
7. Given data items {2, 4, 10, 12, 3, 20, 30, 11, 25}, trace the execution of k-means clustering algorithm for the same.
8. Discuss in detail the applications of clustering in other data mining techniques.

Chapter 6

Other Data Mining Techniques

The previous chapters addressed the key and important techniques in data mining such as classification, clustering, association rule mining. This chapter discusses data mining techniques such as prediction, outlier analysis, characterization and scope for application of other areas such as genetic algorithms in the domain of data mining. Prediction is the data mining technique employed to predict values of attributes that are missing or that are corrupt. Outlier analysis is the process of identifying deviant and errant behaviour in data. It differs from conventional data mining techniques in the fact that it looks out for uninteresting or unusual patterns as opposed to frequently occurring or usual patterns in the input database. It finds extensive application in fraudulent detection applications where the unusual fraud behaviour is more interesting compared to the other patterns from the specific requirement point of view. Characterization as a data mining technique aims to characterize the input database, generating feature descriptions for the various classes or types of data present in the database.

6.1 Data Prediction

Prediction is the data mining technique used to predict or appropriately guess the data values for attributes which are either missing or corrupt. Objective of prediction is to make an educated guess on the data values taking into consideration the data values available with other records and attributes in the database. It attempts to predict values considering the data distribution format in the input database. An application of data prediction would be to predict the salary of an employee in an organization given the experience details. The entire crux about prediction is to have the predicted value as close as possible to the intended or original value that might have been in place.

Prediction in a way might look analogous to the data classification technique

in data mining. Data classification aimed at generating (could be viewed as predicting) class labels for records based on training data information made available. The difference lies in the fact that prediction is more oriented towards data values while classification concentrates on class labels, which can be treated as being categorical in nature. A subtle way of differentiating could be to view classification as the process of generating overall characteristic of records in terms of class label information while prediction as the process of generating or predicting specific data values.

Data prediction as a technique incorporates the concept of regression from statistics in the process of predicting values. Linear regression is used to predict data values for models that are governed by straight line while multiple regression is used to predict values for data that do not exhibit linear properties or are governed by several (more than 1) predictor variables. Multiple regressions, also treated as polynomial regressions attempts to convert the non linear data model to a linear one and is hence an extension of its linear counterpart.

6.1.1 Linear Regression Based Prediction

Linear regression is the simplest amongst the different types of regression where data are modelled by straight lines. Data being modelled by straight lines could be interpreted as the growth of the data being in a linear form or follows a straight line. To be more precise, values for the respective attribute that is linearly modelled grow in linear fashion.

A regression function models a random variable B as a linear function of another random variable A. B could be treated as an output variable while A as an input variable. B is also referred to as the response variable modelled on the predictor variable (A). Thus the objective of linear regression given the setup of variables discussed above is to predict values of B given the values of A. Expressing this mathematically $B = \alpha + \beta A$, where α and β are regression coefficients that control the growth pattern, denoting the Y intercept and slope of the line. Variance of B is assumed to be a constant in the above equation.

Coefficients in the regression equation are solved using the method of least squares that works on the principle of minimizing error between the actual or original data and the estimated one. Given n data samples or records represented of the form $(x_1, y_1), (x_2, y_2) \ldots, (x_n, y_n)$, regression coefficients are estimated using the least squares method based on Equations 6.1 and 6.2.

$$\beta = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{x})}{\sum_{j=1}^{n}(x_i - \bar{x})^2} \tag{6.1}$$

$$\alpha = \bar{y} - \beta\bar{x} \tag{6.2}$$

In the above equations, $\bar{x}$ denotes the average of the values of x $(x_1, x_2, \ldots, x_n)$ and $\bar{y}$ is the average of the values of y $(y_1, y_2, \ldots, y_n)$. Consider the data shown in Table 6.1 composed of sales(S) and price details(P), the plot of which as shown in Figure 6.1. As the figure shows the two variables sales and price are linearly related and hence the data can be modelled based on linear regression of

Sales (S)	Price (P)
1	2
2	1.5
3	2.5
4	6
5	4.5
6	7
7	7.5
8	9
9	7.5
10	12

Table 6.1: Sample Data Set for Linear Regression

the form $B = \alpha + \beta A$, where B here equates to price and A relates to sales. Thus the objective of our prediction exercise is to build a model that can predict the value for sales of an item given its price. Using the equations discussed earlier, the regression coefficients work out as shown in Equations 6.4&6.3. The mean values for price and sales are 5.5 and 5.95 respectively. Thus the linear regression model function works out as Sales = 0.2355+1.039Price. Using this equation one can predict the value of sales for a price value of 12 as 12.7.

$$\begin{aligned}\beta &= \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{x})}{\sum_{j=1}^{n}(x_i - \bar{x})^2} \\ &= \frac{(1-5.5)(2-5.95) + (2-5.5)(1.5-5.95) + \cdots + (10-5.5)(12-5.95)}{(1-5.5)^2 + (2-5.5)^2 + (3-5.5)^2 + \cdots + (10-5.5)^2} \\ &= 1.039 \end{aligned} \tag{6.3}$$

$$\begin{aligned}\alpha &= \bar{y} - \beta\bar{x} \\ &= 5.95 - (1.039)5.5 \\ &= 0.2355 \end{aligned} \tag{6.4}$$

Multiple regression that involves more than one predictor or input variables is nothing but an extension of linear regression and can be modelled as a linear function of a multidimensional feature vector. Consider an example of a response variable Y modelled on two predictor variables X_1 and X_2 as shown in Equation 6.5. Regression coefficients involved in the equation can be solved for using the method of least squares.

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 \tag{6.5}$$

Non linear regression where data does not exhibit linear dependence such as a response and predictor variable(s) having a relationship that may be modelled by a polynomial function, also referred to as polynomial regression can be modelled by adding polynomial terms to the basic linear model. The non linear model

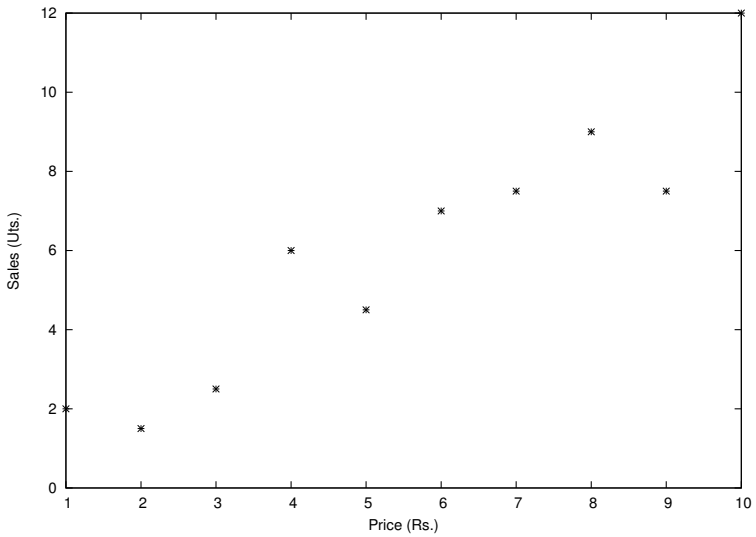

Figure 6.1: Plot for Data set shown in Table 6.1

can be converted to a linear one by by applying transformation to the variables and solving the coefficients using the method of least squares.

A function such as $Y = \alpha + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 X^4$ can be converted to a linear form by the following substitutions such as $X_1 = X, X_2 = X^2, X_3 = X^3$ *and* $X_4 = X^4$, resulting in the transformed linear equation $Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4$.

Other variants of regression models include logistic and poison regression which both fall under the category of generalized linear models. In such models the variance of the response variable is a function of the mean value unlike linear regression where the variance is a constant. Logistic regression models probability of event occurrence as a function of predictor variables. Also log linear models that approximate discrete multi dimensional probability distributions could be used to determine the probabilities for data cells. These models help in better and efficient representation of higher order data cubes and also provide good data compression.

6.2 Outlier Analysis

Data objects or elements that are entirely different from others or are inconsistent in comparison to other data elements is referred to as an outlier. Such data do not comply the general behaviour of the database or data model and exhibit deviant and aberrant beahaviour. An example could be the age attribute of a person containing a negative value. Though from the generic data mining point of view, most data mining systems and algorithms incorporate concepts that

eliminate if not reduce the effect of outliers, the deviant behaviour or pattern could by itself be an interesting knowledge. An example is the withdrawal attribute in a bank related database, where one huge amount could indicate a possible fraudulent usage of the bank account. The technique of data mining that identifies and mines such outliers or deviant and aberrant patterns that convey interesting valuable information is referred to as Outlier Analysis.

Outlier analysis or mining also finds application in business decision making to observe the spending trends of customers, in particular the class of customers who either make extremely huge or low purchases. It is also used in medical diagnosis to identify the deviating responses from a patient to a particular treatment and thereby help in detecting the actual cause. On a integrated scale, outlier mining could be defined as the process of grouping sets of records that behave in a different or deviant manner in comparison to the rest or majority of the data. Yes, it could be viewed as the process of clustering, but with the difference that here clusters look out for the objects or records that have the least similarity and different behaviour compared to the rest of the data. Thus a given a data set of n points and the expected number of outliers o, the objective of outlier mining is to find the top o objects in the database that exhibits highly dissimilar, inconsistent and exceptional behaviour.

Two key phases of outlier mining are identifying the inconsistent data in the input database and then the extraction of the expected number of outliers or deviant data points. The process of outlier analysis is complicated in cases of data involving multi dimensions and involving categorical attributes. Identifying exceptions with respect to numeric attributes are lot more easier in relation to their categorical counterparts. Regression, visualization models might not always a good alternative to outlier mining because of the issues involved with cyclic changing, seasonal data where detected outliers could infact turn out to be perfect real time value. Alternatively the clustering algorithms discussed in the earlier chapter could also be fine tuned to group deviant or outlier data, satisfying the condition that the data objects are highly dissimilar compared to the data elements of rest of the clusters. Three of the commonly used approaches to outlier analysis are statistical, distance and deviation based approaches.

The statistical model starts out with a distribution or probability model for the given data set and then looks out for deviation from the considered model. Distance based technique carries forward the concepts used in clustering, however with the modified objective of grouping or looking out for data points that lie at far away distances. A conceptual or characteristic analysis of objects to detect outliers is referred to as deviation based outlier mining, where characteristics for data objects are created and then objects that differ from these characteristics are identified as outliers. The following sections discuss in detail each of these approaches to outlier detection.

6.2.1 Statistics based Outlier Mining

The statistical approach to outlier analysis assumes a probability distribution or model for the given data set and then performs a satisfiability test, also referred

to as discordancy test to determine if data objects comply with the assumed model or not. Objects that buck the model are treated as outliers. The model requires that the data distribution, knowledge of distribution parameters, etc. be known to start with. It has its base from tests of goodness and fitness, where an hypothesis is accepted or rejected based no computed statistical parameters.

The method proceeds with two hypothesis, namely working and alternative hypothesis. To start with a distribution model F is formulated for the database and the objects of the database are assumed to follow the model. This part of the assumption is what is referred to as the working hypothesis or H, where the data objects comply with the base data distribution model or function. The alternative hypothesis or $\bar{H}$ includes cases where objects do not follow the distribution function modelled but in fact satisfy the trend of another model or function. Thus such objects are treated as outliers from the view point of the working hypothesis distribution model.

Inherent alternative distribution functions models databases where all objects in it come from the distribution postulated by the alternative hypothesis and none of the objects in the database satisfy the test imposed by the working hypothesis. This situation arises when the alternative hypothesis distributions are characterized by different mean or dispersion parameters. The intermediate situation where objects are not entirely failing the working hypothesis but are elements of the alternative hypothesis or contaminants from it. The slippage alternative model consists of object which arise independent of the initial working hypothesis distribution and a portion of it satisfy a modified distribution function imposed by the working hypothesis.

The actual process of outlier detection is either carried out in a block or a consecutive fashion. In case of block wise detection, all of the deviant objects are accepted as outliers or grouped as being consistent or normal data. With consecutive analysis, objects in the database that are least likely to be outliers are identified and eliminated first. This is the human intuition of reaching for a consensus by elimination by ruling out definite non contenders. If one such object turns out to be an outlier, then the rest that carry extreme or far away values in relation to the base are also treated as outliers. This method has its basis as identifying the key point where data dissimilarity starts and then generalizing the behaviour of outliers.

Statistical methods suffer from the limitation that they are not well responsive to multidimensional data situations and also knowledge about distributions and parameters are crucial to the entire process of outlier detection. Cases where not all possible tests have been carried out or it is not possible to model a standard distribution for the data cannot be handled by statistical methods. These limitations are overcome by the distance based method discussed in the following section.

6.2.2 Distance based Outlier Mining

Distance based outlier analysis is based on the principle of measuring distances of objects much like distance based clustering algorithms and then classify those

objects whose distances exceeds a specified threshold. Given a database D, an object o is termed an outlier when atleast a fraction of the objects in D (say f) are at a distance greater than d from o. All those objects that lie at a greater distance are interpreted as following a specific model which o which is deviant from the fraction of objects f in the database D is treated as an outlier. In terms of neighbourhood analysis, distance based methods concentrate on identifying those objects in the database which do not have a sufficient amount of adjacent or neighbourhood objects in the database. This method offers the advantage of reduced computations and avoids the complication of fitting distributions of models and then analyze for outliers based on whether objects in the database satisfy the working or alternate hypothesis.

Index based outlier detection algorithm aims at outliers based on neighbourhood and distance measures. It employs multidimensional indexing structures such as R trees in the process of the locating neighbours to a given object. Assume that there are a maximum of m objects that are present within the d neighbourhood of an object o and the value of m exceeds the limit set for an object to be viewed as an outlier, then o is ruled out as an outlier object. The index based algorithm runs in $O(sn^2)$ time where s is the number of dimensions and n is the total number of objects in the database. The index method scales well to increasing database sizes but however suffers from the limitation of a time consuming index creation operation. This limitation is avoided in the nested loop technique which attempts to minimize the input output operations and organizes the data efficiently by loading it on to an appropriate partition of the main memory buffer that reduces the disk operations.

Another approach to implementing distance based outliers is to group objects into cells and then identify outliers on a per cell basis as opposed to a per object basis approach adopted by other techniques. Distance based neighbourhood measure is employed in the process of identifying an entire cell or objects within the cell as outliers. Cases where a cell is composed of a mixture of outlier and non outlier objects, the object by object outlier technique described earlier is employed. Distance based methods require the values of distance and the set of objects that need to satisfy or fail the threshold to be user defined and more often than not arriving at optimal values remains an issue. This issue is alleviated in the conceptual or concept based outlier detection.

6.2.3 Deviation based Outlier Mining

Deviation based methods operate on the principle of identifying or establishing characteristics of objects that constitute a group and then establish outlier objects based on whether its characteristics deviate from the established one. Two approaches to deviation based outlier detection are sequential exception and data cube based technique. The sequential execution technique is similar to the human visualization approach of identifying deviant data and builds subsets of a given set of objects.

Objects are tested for dissimilarity based on a similarity function on a per subset basis. The technique maintains the following data structures in the

process of mining outliers: Exception set, Dissimilarity function, Cardinality function and Smoothing factor. The exception set consists of the deviant or outlier objects and forms the smallest set of deviant objects. The similarity function measures similarity between objects and returns low values for objects that are similar and high values for dissimilar objects. It is not based on distance metric and computes similarity in an incremental fashion on the subset in the prior level.

One example dissimilarity function is one that measures for deviation from the mean where dissimilarity increases (as a result of the squaring logic) when object or object sequences in subset at level i-1 does not match any member of elements of subset at level i. The cardinality function measures the count of objects in a set while the smoothing factor attempts to reduce the dissimilarity to the extent possible. The method works on the principle of comparing for similarity of objects in a set with its predecessor set, rather than comparing it with a complimentary one. The algorithm is fine tuned to eliminate the effect of input ordering of elements by repeating the process with a randomized sequence of subset of objects.

The data cube approach to outlier mining uses a data cube in the process of comparing normal and outlier objects. Data cube offers advantage of viewing the database at varied levels of abstraction with operations such as drill down and roll up to project data at specific or generic levels of the abstraction hierarchy. Visualization techniques are employed to compare the objects values in the data cube cells at different levels of abstraction and object whose value falls far away from its expected value is treated as an outlier if the same trend is observed at the varied levels of abstraction.

6.3 Conceptual Techniques

Data Mining techniques as discussed earlier are broadly classified as being descriptive or predictive in nature. Data mining techniques discussed so far employed inferencing or to be more precise resulted in predictions being made on the input database. Techniques such as association rule mining, prediction, classification, etc. all built models from the database or worked with standard models in the process of inferencing or predicting knowledge (to the extent possible accurate, complete and useful) from the input database. The other class of data mining techniques which involve more of characterization of database properties where no inferencing is employed but data are interpreted at varied abstractions and overall features or concepts are mined is referred to as descriptive data mining. This section discusses the data characterization and comparison conceptual data mining techniques. These are referred to as conceptual techniques due to the fact that the overall objective is to identify the characteristic or overall concept that best describes the database.

Characterization as a data mining techniques offers concise and summarized representations of the data (at concept level) while comparison aims to compare or contrast the properties of concepts of data objects or elements. The next

immediate issue that can crop up is how does a descriptive data mining system differ from its online analytical processing or OLAP counterparts. Essentially an OLAP system organizes data in databases as data cubes and support operations such as drill down and roll up. Aggregate operations such as sum, count, etc. are supported to present a collective view of the data. However such measures are defined only for numerical attributes or data and realistically databases may be composed of complex data types as spatial, text, etc. Aggregate or conceptual operations such as merging of texts, collection of non numerical data etc. are not supported in OLAP. In a way descriptive data mining systems could be viewed as an extended OLAP system with teh support for handling complex data types and operations over them.

Also another difference is the fact that OLAP systems are highly user driven with the roll up and drill down operations controlled by the users. Descriptive data mining systems on the other hand promote more automation in the process of concept description and thereby eliminates the need for user interaction and knowledge in the process of concept description. Requirements such as identification of analysis determining attributes, number of dimensions to be included etc. are alleviated in descriptive data mining systems.

6.3.1 Data Characterization & Generalization

Data generalization is the process of abstraction of large data sets from lower conceptual levels to higher levels. An example would be to project sales details of a company on a half yearly basis given that the database contains details on a fortnight basis. Data cubes and attribute oriented induction constitute two of the widely used techniques for generalization. Data cubes could be viewed as performing offline aggregation before user queries are processed by the OLAP system. However attribute oriented induction is an online technique is the process of generalization based on the number of distinct values of attributes. Attribute removal and generalization strategies are employed in the process of generalization. Identical and generic tuples are merged and their appropriate counts are maintained. Attribute oriented induction or AOI first aims at cornering the task relevant data and then performs attribute removal cum generalization.

Attribute removal removes attributes that have a large set of distinct values without any generalization operator or those where higher level concepts are defined in terms of other attributes. In such cases the attribute is deemed to be non contributing to the generic form and hence removed from the generalization process. When large distinct set attributes are present with generalization operators , an apt operator is chosen and applied to the attribute. This generalization will result in the generic form covering more records from the original database than by its detailed or specific equivalent. These strategies are also referred to as learning from examples.

Generalization is more of a subjective process, since the level to which an attribute has to be generalized could vary from user to user depending on the abstraction level sought by the query. Over generalization results in too much

genericty with practically no application while under generalization might result in too specific rules which are not satisfied on a large scale. The attribute generalization control maintains a balance between the two levels.

Thresholds are maintained to control the generalization level of attributes. These thresholds are either made uniform for all attributes in the database or separate thresholds for each attribute are maintained. Attribute removal is recommended in cases when the number of distinct values for attributes is large. Cases where generalization has to be reduced, the threshold values are decreased. Another alternative to generalization control is based on relation level thresholds, where relations that consists of a specific number of tuples that exceed a threshold are subject to further generalization.

Ideally attribute based generalization is applied first to generalize the attributes in the relation and then the second strategy is applied to achieve relation level generalization. Most database applications require support for quantitative analysis of data at the varied levels and hence in the process of generalization using induction counts of the tuples are also maintained. Attributes such as name, phone number, etc. are likely to contain large number of distinct values and are best removed during the process of generalization. Attributes such as gender are retained as it is as they are in a concise and generalized form. For attributes such as birth date, birth city, etc. for which concept hierarchies can be defined should be generalized. Again with such attributes, lower level details such as street number, etc. which are likely to be large in number again should be dropped.

The data cube alternative to generalization uses either a data cube that is precomputed before the query is submitted to the system or on a dynamic basis relevant to the task data. Generalizations are decided either based on the predefined conceptual levels or based on the attribute level of abstraction sought in the query. Operations such as drill down and roll up are applied to the cube to support the desired level of generalization. These methods suffer from the limitation of expensive cube computation and large storage requirements. The generalized data are present to the user in different formats such as cross tabulations (tabulations/tabular column representation), bar charts, pie charts, 3D representation, etc.

Attribute relevance analysis is a key phase of data characterization because otherwise statistically irrelevant and weak attributes might get included in the process and relevant attributes might not get the priority they deserve. As mentioned earlier, both situations of too few attributes or too many attributes might hinder the interpretation that users are able to derive. Attributes in databases that help in distinguishing a specific class of objects from another set or class are generally treated as being highly relevant. It is these attributes that characterize the different classes of records in the database and hence such attributes should be given preference over other not so distinguishing attributes. Techniques which incorporate rank relevance among attributes and then perform characterization or comparison operations are also referred to as analytical characterization and comparison. Measures such as information gain, entropy that reflect the information content of attributes in relation to others are generally

employed to identify task relevant attributes by most data mining techniques.

6.3.2 Data Comparison or Discrimination

Most database applications require data to be processed and visualized in relation to other types or classes of data present in the database. Data comparison is the process of mining descriptions that distinguish one class from other comparable classes. The class that is used for comparison is referred to as the target class and the other set of classes which are compared with the target class is referred to as the contrasting class(es).

The target and contrasting classes must be comparable and share similar dimensions and attributes. The attribute generalization approaches discussed earlier with respect to a single class can be extended to handle all the classes that are being compared. This generalizes all the classes to the same levels of abstraction and hence provides a uniform platform for the comparison operation. Thus attribute location with hierarchy such as city, state, country etc. could be generalized at the city level and later be used to compare the sales in a particular city with others over the last 3 years.

Data comparison as a data mining technique consists of data collection, relevance analysis, synchronous generalization and presentation of comparisons to the user. The data collection phase aims to gather relevant data from the query and clearly establish the target class and contrasting classes. Relevance analysis of attributes and dimensions helps in eliminating redundant and weak attributes from the comparison task. Synchronous generalization basically attempts to generalize the target and contrasting classes at the same levels of abstraction resulting in prime target class and prime contrasting classes. Finally the derived comparisons using generalization are presented to the user in the form of rules, tables and graphs. As before, a count measure that reflects the number of tuples that satisfy the comparison would contribute to the requirements of quantitative database applications.

Summary

The chapter introduced the readers to the concept of data prediction and outlier analysis. Prediction and outlier analysis find extensive applications in business decision making such as predicting or forecasting sales and observation of customer behaviours, etc. Outlier analysis is extensively used by fraudulent detection systems where end users behaviour are constantly monitored and the system is alerted in case of deviant or outlier behaviour. These data mining techniques are all the more significant and useful in todays open, competitive and to a certain extent greedy life styles. Characterization techniques that generate property or characteristic descriptions for the various data classes and compare one another has also been discussed. Data Mining as a branch of computer science started evolving since the 90's and till date has been extensively researched resulting in several efficient, useful data mining techniques and al-

gorithms for the same. The growth of the internet and information technology revolution has prompted a sudden change in the types of data that are being manipulated. With the advent of the internet, huge and enormous amount of unconventional data such as images, text, video, etc. are available and the need to discover knowledge from such data resources is increasing. The chapter to follow addresses this recent trend in data mining, referred to as Multimedia Data Mining.

Review Questions

1. Define data prediction and differentiate it from classification.
2. Explain the different types of regression models used for prediction.
3. Define an Outlier and list possible applications of outlier mining from your work domain point of view.
4. How does outlier mining differ from clustering?
5. Compare and Contrast the various outlier analysis techniques. (refer external source).
6. Define Data Characterization and Discrimination.
7. Define Attribute Oriented Induction and explain its significance from a specific data mining technique of your choice.
8. Consider a sample database of your choice and apply attribute oriented induction to technique to perform data generalization.

Chapter 7

Multimedia Data Mining - The Recent Trend

Multimedia, which perceives the integrated presentation of data that exists in multiple forms such as images, text, video, etc has come to stay as the dominant feature of modern age computing. The growth of internet and information technology revolution has resulted in huge amounts of multimedia data being available and the driving need to extract useful knowledge and meaningful information from them. Multimedia Data Mining (MDM) is the mining of high level information and knowledge from large multimedia databases. In other words, MDM is the correlation or adaptation of conventional data mining techniques such as Classification, Clustering, Association Analysis in the domain of multimedia data.

MDM differs from its conventional or relational counterpart in the presence of certain key data specific properties in multimedia data. Video and audio data possess temporal or time related properties while images are bound by spatial properties. The process of data mining must incorporate these properties in the knowledge extraction phase and the extracted patterns. Conventional data mining algorithms do not incorporate data specific properties and hence are not suited to MDM. The recent research trend of MDM concentrates on the evolution of existing data mining techniques that incorporate multimedia data specific properties. MDM is thus an adaptation of existing or evolution of new data mining techniques that address the data specific properties in the mining process and mined knowledge.

MDM is a recent research trend, with the coinage of the term around year 2000. Exhaustive research on conventional data mining has resulted in several algorithms and techniques for knowledge extraction. The presence of enormous amounts of non-conventional and complex types of data, the need to extract information from them and the prevalence of efficient data mining techniques has given way to Multimedia Data Mining. The chapter presents a survey of the existing techniques for MDM, primarily concentrating on image mining.

7.1 Mining Image Datasets or Image Mining

The process of knowledge extraction or mining from images, a sub-area of MDM is referred to as Image Mining. An immediate correlation of data classification in the domain of images is to generate class labels or information for input images that best describe the overall nature of the images (pictures) such as Scenery that relates to nature related pictures, Sports Stills that relates to sports related photographs and so on. The classification could be at varied levels of the hierarchical representation of the input data ranging from a generic class such as Scenery or Sports to specific labels such as Mountains to Athletics.

Another possible meaningful interpretation of a conventional data mining technique in the domain of images is Clustering that could be employed to group related pictures together. Images could also be subject to association rule mining technique to identify relationships among the various constituent objects and subsequently create the so called association rule based classification systems. An example would be to classify Scenery images based on the occurrence of objects such as birds, mountains, etc. Sections to follow in this chapter detail on the various kinds of knowledge that can be mined from images and the existing literature and works on image mining.

7.2 Association Mining on Images (or) Image Association Mining

Image Association Mining is the process of discovering associations and relationships among the constituent objects of the image. It finds application in association rule based image classification systems where the identified associations are related to various classes and based on the associations mined the images are classified. An example as mentioned earlier is classification of nature related pictures as Sceneries based on the occurrence of objects such as birds, mountains frequently in the input image. Frequent objects (pattern) mining has been an important research issue over the years and several algorithms have been proposed to mine frequent patterns efficiently. Frequent pattern mining in temporal domain is the thrust research area addressed by the thesis. Frequent pattern mining and algorithms for the same are discussed in greater detail and depth in Chapter 5.

7.2.1 MultiMediaMiner System Prototype

Image association mining is the process of discovering interesting relationships and associations among the various constituent objects of the image. Han et.al proposed a system prototype for MDM capable of performing data summarization, comparison, classification, association mining and clustering. The prototype includes four major components namely Image Excavator that extracts images or videos from multimedia repositories, a preprocessor that extracts image features, a user interface and a search engine for matching queries with

image or video features in the database.

The processed database stores description information such as image file name, image type, keywords list, etc and feature descriptors that are vectors denoting visual characteristic. Some of the possible vectors are colour, most frequent colour vector, etc. Layout descriptors contain colour and edge layout vectors. All images are assigned an 8*8 grid and the most frequent colours for each of the 64 cells are stored in the colour layout vector and the number of edges for each orientation in each of the cells is stored in the edge layout descriptor.

The multimedia associator module or MM-Associator finds association rules from the relevant sets of data in image databases. Types of associations mined include examples such as "An Image that is big and related to sky is blue with a possibility of 68% or An image that is small and related to sky is dark blue with a possibility of 55% ".

7.2.2 Perceptual Association Rules

Tesic et.al proposed an image association mining algorithm that generates perceptual association rules. Perceptual associations are an extension of traditional associations and are used to distill frequent perceptual events in large image datasets in the process of discovering interesting patterns based on spatial properties. The scheme follows a three phased approach comprising of perceptual labeling of image regions based on a visual thesaurus, tabulation of first and second order associations using spatial event cubes and mining of higher order associations and rules. Images are labeled in a perceptual manner using a visual thesaurus. The spatial event cubes incorporate the image specific spatial adjacency property. The conventional Apriori algorithm is adapted to mine perceptual associations from the transformed image data made available by the end of the second phase.

7.2.3 Recurrent Items Mining in Multimedia Data

Zaiane et.al proposed a recurrent items multimedia data mining technique based on progressive resolution refinement. The technique employs a progressive resolution refinement approach in which frequent item sets at rough resolution levels are mined and progressively finer resolutions are mined on the candidate frequent item sets obtained from mining rough resolution levels. The work improves the earlier segmentation approach based on properties such as colours and textures in images. This content based multimedia data mining system prototype is extended to include visual features localization, spatial relationships, etc in the process of MDM.

Algorithms that mine multimedia data in general and image in specific require recurrent items and spatial relationships to be incorporated. Feature localization is used to identify features based on locality and proximity as opposed to the image segmentation approach that might not always give a good

representation of the image content. The approach addresses the issue of recurrent or repetitive features mining that might present more information than just information about the existence of the feature. An image is modeled as a transaction whose items are treated as the visual features that are extracted. Items on the left hand side of an association rule repeating on the right hand side is an interesting factor in image analysis.

Recurrent objects in images are a commonality and present interesting perspective of images. Transactions are basically composed of objects of the image that are extracted based on visual features such as colour, textures or spatial relationships such as horizontal next-to, vertical next-to, overlap, etc. From the transformed transactional data, an Apriori principle based algorithm that generates recurrent associations between the constituent objects is proposed.

Zaiane et.al proposed a technique for discovering spatial associations in images. It employs a three phased approach consisting of feature localization, spatial relationship abstraction and association discovery steps. Feature localization extracts distinctive areas in images based on colours and textures. The second step identifies spatial relationships between the extracted areas in images such as contains, overlaps, etc. resulting in transactions representing images. The last step discovers associations in the images from its transactional equivalent.

Feature localization improves the results obtained via conventional image segmentation approach. The work employs properties namely locality and proximity incorporating features such as local enclosure, geometric parameters such as mass, centroid and variance for the local enclosure. The locales associated with images are compared and spatial relationships are evaluated based on the concept of minimum bounding circle, which is the smallest circle that could contain the whole locale. The technique adopts an Apriori principle based recurrent item mining algorithm to discover spatial associations from images or transactional equivalent of images.

7.3 Image Classification - A Data Mining Approach

Image Classification is the process of classifying images to predefined labels employing classical data classification algorithms. As mentioned earlier classification systems are generally preceded by an association mining phase to discover associations and then use the extracted associations to classify images. Association rule based classification systems tend to have greater classification accuracy compared to pure classification systems.

7.3.1 Association Rule based Image Classification Systems

Zaiane et.al propose an image classification technique based on image association mining. It is used to classify medical images to aid cancer analysis, alternatively referred to as mammography. The preprocessing phase employs

data cleaning and transformation to isolate or remove incomplete, noisy and inconsistent data from images. Cropping drops unwanted image portions which eliminates most of the background information and noise. This handles the situation of large images where background constitutes more than half the image content and images scanned at differing illuminations resulting in too light or too dark images. Differing image sizes are addressed by normalizing image coordinate values. Image enhancement techniques in spatial or frequency domain improve the image quality. Histogram equalization is employed to avoid over bright or dark situation in images, preceded by a noise removal step so as to avoid enhancement of noisy information.

The next phase of feature extraction is employed to create ready to mine transactional databases. Statistical parameters namely mean, variance, skewness and kurtosis are the features that are extracted. Original image is split into four parts namely NE, NW, SW and SE for better localization of regions of interests in images. Normal images stored the resulting or extracted features associated with a transaction identifier while abnormal (cancerous) images stored only features extracted from the abnormal or cancerous part associated with a transaction identifier. Finally from the transformed transactional image data, classes are mined by employing association rule base classification.

Objects of the image are modeled by the categories and features extracted. A class is treated as a separate training collection and association rule mining is applied to it. Transactions that model the training documents represented by a specific category are grouped together. Then an Apriori based association rule mining algorithm is employed to generate rules involving objects and class identifiers. Generic rules are favoured over specific rules and in cases of large rule sets, pruning based on confidence factor is employed to keep out uninteresting and more specific rules, retaining generic and interesting rules.

Zaiane et.al proposed an image classification system incorporating neural networks in the classification phase. It employed preprocessing and feature extraction phases as in their earlier work. However instead of adopting an association rule based classification approach, a neural network based classification model is proposed. The neural network architecture is composed of three layers, input, hidden and output layer. The input layer composed of nodes equal to the number of elements in one transaction. The output layer node gives the class of the image namely normal or abnormal. Internal weights of the neural network are adjusted according the transactions used in the learning process in the training phase. Medical images are classified making use of the trained neural net.

7.4 MDM using P trees

Reference [3] proposes a spatial data structure, the peano count tree that provides an efficient, lossless, compressed and a data mining ready representation of the various multimedia data. Three different data organization formats supported are band sequential, band interleaved by line and band interleaved by

pixel formats. Reference [3] proposes a bit sequential format for data organization that supports compression and hierarchical representation features in variance to the band sequential format.

Peano count trees are nothing but specific data structures that allow for storing and mining multimedia features efficiently and accurately. The image is divided into quadrants and the tree maintains a record of the count of 1 bits of the image data for every quadrant. P trees are similar to the quadtrees counterpart. The tree supports logical complement, and and or operations that permit efficient manipulation of the bit wise representation of the image to represent the regions of interest. The system supports data mining techniques to perform classification and association mining.

7.5 Video Mining Techniques

Video Mining or Video Data Mining is the process of discovering knowledge or interesting patterns from video data and databases. In other words, it is the correlation of conventional data mining techniques such as classification, association mining, clustering, prediction, etc. in the domain of video data. Video mining finds application in video database indexing and management. Video data mining adopts a data mining approach to knowledge extraction from video data as opposed to a conventional signal processing approach.

Most video mining systems incorporate signal processing to transform the original video data to an alternate format that is suitable for further mining operations. Video shot segmentation, key frame selection strategies, textual and audio analysis are employed in the process of video data transformation. Collectively these techniques are referred to as video preprocessing and is essentially the data transformation phase of a data mining system.

Classification of videos as news, commercials, sports, etc. and movies as romantic, comic, tragic, etc. and grouping(clustering) of related video data for efficient indexing are a few applications of video mining. Futuristic event prediction, special pattern and event detection, unusual event identification or outlier detection are a few of the types of information that can be extracted from video data. This chapter presents a survey of video mining, throwing light on the existing literature for the same.

7.6 Types of Knowledge that can be mined from Videos

Reference [4] discusses the possible knowledge that can be mined from videos. Movie data often involves more than one stream such as audio/video and hence makes the process of video and multimedia data mining all the more challenging. The prototype miner proposes to demarcate textual, audio and video information and perform further data separation analysis that extracts individual components such as scene, objects of video, sounds and voice from audio data.

Textual analysis such as conceptual hierarchies, keywords, etc. extract metadata and structural information from video data. It is infact a transformation of the original video data to an alternate format that is suitable for further mining operations.

Possible knowledge types that can be mined are the compositional structure of the movie such as scene sequencing, shot segments, etc. Interesting events such as emotions, mood, etc. are also mined from the video data. An act of crying might indicate sadness while clapping or laughter might indicate happiness. Acts of bomb blasts, gunshots could indicate street violence. Associations or event patterns such as acts of violence are generally followed by sad scenes can be mined.

Associations could be mined at various levels of the hierarchical representation. For instance associations involving certain kind of objects in a frame representative of a scene are associated with other kinds of objects in the frames that represent the next scenes to follow. Also higher level associations such as directors, money invested, movie type, etc. could be mined from video data. Association mining when employed in the domain of domain multimedia data could be used to predict future objects and events based on the occurrence of certain sequence of events or kinds of objects frequently and in an order.

Video data could also be subject to clustering to group related movies together resulting in the formation of clusters such as Romantic Ending Movies, Tragedy Movies, etc. An individual video could also be subject to classification that generates the above mentioned classes describing the overall nature of the movie based on the story plot. Also trends in movies such as those that begin with happy scenes(happy beginnings), sad beginnings have happy or sad endings or combinations of them can be mined. Video data could also be subject to association rule based classification, where frequent patterns or events or episodes mined from videos could be used to classify them. Thus events such as gun shots, blasts followed by sadness could be a possible indicator for movie type being violent.

Association mining could also be employed in video summarization, wherein the frequent patterns generated that occur frequently in the video data are included in the summary. Event prediction is another scope of application for frequent patterns and events generated. Futuristic events could be predicted based on the occurrence of a certain sequence of events frequently. An example application would be to predict the occurrence of crowd trouble in the presence of events such as communal rallies, inflammatory political speeches and religious processions. Video association mining and its applications are discussed in greater detail in the chapter to follow.

7.7 MDM framework for raw video sequences

Reference [5] proposes a general framework for real time video data mining. First it groups the input frames to a set of basic units that are relevant to the structure of the video. The groups are referred to as segments and forms

the building blocks for video databases and video data mining. Later segments are characterized to cluster into similar groups by extracting features such as motion, object, colours, etc. from each segment. The decomposed segments are finally clustered into similar groups. Three types of videos as identified in [5] are the produced, raw and medical videos. Movies, news, dramas, etc. constitute produced videos. Traffic, surveillance videos, etc. are examples of produced videos while ECG's and other health related videos constitute medical videos. A general framework that can handle raw videos in real time and perform video data mining is proposed.

The framework is composed of five major phases namely grouping frames to segments, feature extraction, indexing/clustering, video data mining and video data compression. The first phase groups the incoming frames into meaningful pieces or segments as opposed to the traditional shot segmentation approach. The second phase characterizes segments by extracting features such as motion, objects, etc.Then the decomposed segments are clustered into similar groups based on k means clustering algorithm. The next two phases actually perform the mining of the processed video and data storage in compressed form. Also a meta-data and knowledge base that includes output of the various phases is created and hence aids in object identification, object movement pattern, interesting pattern recognition (normal or abnormal). An extension of this framework is applied in [6] for real time video data mining for surveillance.

7.8 Video Classification Prototype

Reference [7, 8] proposed a video mining prototype for efficient database indexing, management and access in the domain of medical video data. Video shot segmentation and representative frame selection strategies are used to transform the original continuous stream of video to physical data units. Later video shot grouping, merging and clustering techniques are employed to organize the video shots in a hierarchical fashion.

The proposed classminer prototype supports video classification, summarization and retrieval operations. It specifies a video content structure that is composed of video shots, video group, video scene and clustered scenes. A video shot is nothing but the collection of frames that result from a single continuous run of the camera from the moment it is turned on till it is turned off. A video group is an intermediate between physical shots and semantic scenes, wherein a group could be a collection of temporally or spatially related video shots. A video scene is a collection of semantically related and temporally adjacent groups that project higher level concepts and stories.

Thus video content structure is created in four steps namely (i) video shot detection, (ii) group detection, (iii) scene detection and (iv) scene clustering. The original continuous video sequence is first segmented into physical shots and the resulting video shots are then grouped into semantically richer groups later. Similar neighbouring groups are then merged into scenes. Finally clustering is employed to group similar scenes together and thereby eliminate redundant

information in the video data. Video shot and scene detection employs classification and merging operations during which representative shots of a group and representative groups of a scene are identified. Clustering of video scenes is performed using the k-means clustering algorithm.

Once the video structure is mined, event mining strategies are applied to detect event information within the detected scenes. Five types of special frames and regions are detected namely slides, black frame, frame with face, frame with large skew area and frame with blood red regions. The event mining strategy employed identifies various event categories such a presentation scene, dialogue scene, clinical operation, etc. A presentation scene is identified as one that is composed of group of shots that contain slides or clip art frames. There should be no speaker change within shots and atleast one shot should contain a face closeup.

A dialogue scene is nothing but a group of shots containing both face and speaker changes. Speaker changes are deemed to occur at adjacent shots and atleast one speaker should be repeated more than once. A clinical operation relates to medical events such as surgery, diagnosis, etc. Clinical operation is identified as a group of shots without speaker change and atleast one shot is composed of blood red or closeup of skin regions. To support efficient video overview, visualization and easier video content access, a scalable video skimming tool is also supported by the classminer prototype.

Four different levels of skimming are supported ranging from representative shots of clustered scenes to all scenes to all groups, and to all shots respectively. Efficient interfaces that control the various skimming level are provided. Efficient video content access based on mined events is presented to the user in a coloured interface format where varied regions in the video are represented with colours that relates to the various event categories that have been mined or classified. Video summarization and hierarchy based video browsing are other applications of the classminer prototype.

7.9 VideoCube: A tool for Video Mining & Classification

A novel tool for video mining and classification was proposed by Pan et.al. It classifies video clips into one of possible classes such as news, commercials, etc. It supports an automatic feature extraction phase utilizing independent component analysis. It works with video and audio information and generates vocabulary that describes both still images and motion. Classes are identified with a list of features (basis functions) that are used to compress its members.

A new video clip that is to be classified is assigned to the corresponding class (vocabulary) that can compress the video clip the best. Independent component analysis much like principal component analysis is an useful tool for identifying structures in input data. Visual features are derived from pixel information incorporating spatial and temporal adjacency features which are grouped into

cubes as units of processing. Visual and audio bases comprising visual and audio features are extracted using independent component analysis. The set of audio and video bases that give the smallest reconstruction error is identified as the one that compresses the clip the best.

7.10 VideoGraph-A tool for Video Classification

Another tool for video classification that has been proposed by Pan et.al is VideoGraph. The tool visualizes the structure of the plot of a video sequence by stitching together similar scenes that are apart in time. The tool automatically determines and visualizes the story plot of a video clip and finds application in classifying videos as being news, commercials, etc. Shot groups namely persistent and recurrent are identified. A shot group is a set of shots, which is nothing but a set of consecutive similar video frames. A shot group composed of multiple, consecutive shots is referred to as persistent while one that contains multiple, non-consecutive shots is referred as recurrent. Basic shot groups are those that are both recurrent and persistent and are used in the construction of videograph, which is a directed graph in which nodes correspond to shot groups and edges denote their temporal succession. Videograph finds application in classification, story plot visualization and efficient video browsing.

7.11 Video Editing Rules Mining Prototype

A video editing system support system that extracts video editing rules employing data mining techniques was proposed by . The process of video editing imparts meaning to video data by connecting the various fragments. Video editing rules extracted constitute the video grammar which dictates the shot connection procedure. Metadata information such as shot size, duration, camera movement, etc. are incorporated in the grammar. An Apriori principle based strategy of generating candidate and frequent periodic patterns is employed in the process of cinematic rules generation. Rhythms and semantic patterns in movies denoting the appearance or disappearance of target character's and their durations are extracted using data mining techniques. The extracted rhythms are employed in topic classification such thrilling, romantic, uneventful, battle etc.A binary tree based classification model that identifies possible classes is proposed and is also applied for browsing large videos at a glance (summarization).

To expedite the process of semantic patterns extraction in video data and hence video editing, a parallel video data mining method. Raw level metadata from the various shots of the video are extracted to generate the spatial and temporal semantic information. Metadata such as colour hue, colour saturation for the spatial aspect and shot size, duration, camera movement, etc. for the temporal aspect are extracted resulting in a multistream sequence, which is to be mined for sequential patterns. The sequential pattern mining algorithm

employed is based on the level-wise principle of Apriori, incorporating parallel computation strategies for improved performance.

7.12 Other Video Mining Approaches

An audio visual event detection based on semantic labels mining is used to demarcate commercials and program(s) segments from television program videos [9]. The scheme extracts audio/visual features from video chunks and then employs k-means clustering to identify commercial/program clusters. Another commercials detection cum sports highlights generation prototype employing supervised audio followed by unsupervised unusual event discovery[10]. The proposed technique is computationally simple and robust compared to a purely unsupervised approach which is computationally complex and unmanageable. References [11–13] propose video classification and summarization prototypes employing hidden markovian models.

Reference [14] proposes a news video segments classification scheme using low level audio visual processing and domain knowledge. Segments such as anchor/reporter, voice over and sound bite are identified using audio and video processing techniques. A genre classification prototype to classify movies/video data into action and non action and sub classify them into comedy, horror, etc., commercials, news, music and sports videos is proposed in [15, 16]. Event analysis of videos to identify the various events based on distance measure or clustering is proposed in [17]. A video data base management system and a framework for video data mining supporting content representation, indexing, storage, query processing is proposed in [18]. References [19] perform movie review or opinion mining to identify classes such as positive and negative using supervised and unsupervised techniques. GeoPlot performs spatial data mining on video libraries correlating the conventional association rule data mining technique to discover relationships between natural calamities such as earthquakes, hurricanes, floods, etc. [20]

7.13 Video Association Mining

Video Association Mining is the process of identifying interesting relationships and associations in a video data. It is the correlation of conventional association rule mining technique in the domain of video data, discovering video sequences or patterns subject to the constraint of being frequent with respect to the support count. Extracted video associations find immediate application in event prediction cum detection, classification, summarization, indexing and retrieval operations[21–23]. Like most video mining systems, video association mining involves two key phases of transformation and association mining. The association mining phase concerns generating frequent patterns and hence video associations from the transformed video data made available by the end of the transformation phase. This chapter discusses video association mining and the

existing prototype for the same.

7.14 Video Associations

Video Associations establish interesting relationships among the constituent entities of video data. Videos are generally classified as being those with content structure such as news, movies, etc. and those without content structure[21–23]. These videos are subjected to editing process to package shots into scenes that convey video scenarios. Two types of scenes that are possible are those that are composed of visually similar shots and those that consist visually distinct shots. The former results from shots of same object taken from different viewpoints while the later consists of shots of different objects.

Associations discovered from scenes consisting of visually similar shots are referred to as intra associations. An example is a shot cluster sequence such as AAA, where all items involved in the association are the same. It is also referred to as self coherence of A that indicates inherent sequential association of A and itself. Associations discovered from scenes composed of visually distinct shots such as ABC (A, B, & C are visually distinct shots) are referred as inter associations, involving different items in the associations.

Formally a video association is defined as a sequential pattern $X_1 \cdots X_L$, where X_i is a video item and L denotes the length of the association. With inter associations the intersection of the various video items or shot clusters is empty. The process of video association mining incorporates temporal ordering of items or shot clusters such that $X_i^t < X_j^t$, denoting the fact that X_i happens before X_j. Two measures of statistical significance that govern the process of video association mining are temporal support and confidence, analogous to support and confidence parameters of conventional association rule mining.

Temporal aspect that is crucial to video association mining is incorporated in the form of temporal distance parameter which denotes the distance between neighbouring items. Temporal distance between any two items is the number of shots that appear between them. An item in video association mining is basically a video shot cluster or group consisting of visually similar shots. As before a video shot records frames resulting from a single continuous run of the camera, from the moment it is turned on to the moment it is turned off.

The temporal support of an association (sequence or pattern) is defined as the number of times the association is shown sequentially in the entire shot cluster sequence subject to the temporal distance threshold. A temporal distance threshold of ∞ covers patterns of all temporal distances. Confidence of an association or a pattern is the ratio between the temporal support of the association and number of maximal possible occurrences of the association. For example, the temporal distance of pattern AB in the sequence ACBE is 1 and ADEB is 2. Temporal Support (TS) is the number of times the pattern appears in the input sequence subject to the temporal distance threshold. Thus TS of pattern ABC in the input sequence ABABACABC for TD=0 is 1 and TD=∞ is 2.

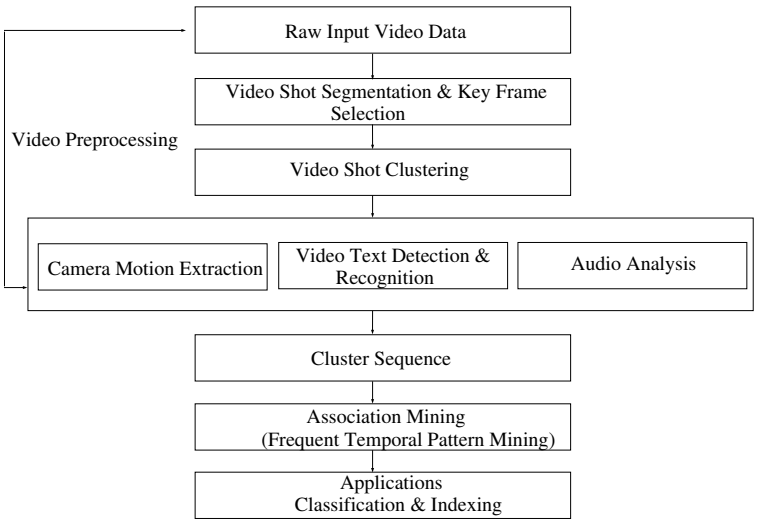

Figure 7.1: Video Association Mining System Architecture

7.15 Video Association Mining Prototype

X.Wu et.al [21–23] proposed video association mining algorithms for video summarization and efficient database management. They explore the area of video association mining in which association patterns are characterized by sequentially associated video shots and cluster information. Two key phases namely Transformation and Mining are employed in the process of video association mining. The transformation phase converts the original video input into an alternate transactional or relational data format. It is achieved by video shot clustering where the original video is subject to clustering, resulting in a shot cluster sequence, a cluster being a collection of visually similar shots. Clustering is based on the principle of maximizing intra cluster and minimizing inter cluster similarities.

7.15.1 System Architecture for Video Association Mining

The proposed system architecture shown in Figure 7.1 composes video preprocessing, video association mining engine and applications (video indexing/ summarization) components. The original input video data is subject to preprocessing with the overall objective of transforming the original video data (non relational) to transactional or relational data format. The preprocessing phase results in shot cluster sequences, after which the problem of video association mining gets reduced to that of mining frequent patterns or sequences (cluster sequences). Generated associations and frequent patterns find application in association based classification, indexing and summarization operations.

7.15.2 Video Preprocessing

The video preprocessing component of the architecture employs shot detection techniques discussed in [7, 8] to detect video shots. Given a video data V consisting of N shots $S_1 \cdots S_N$, representative frame selection strategies [7, 8] are employed to determine key frames (representative frames) among the detected shots. For every key frame visual properties such as colour, texture are extracted. Subsequent to shot segmentation, video shot clustering is employed to group visually similar shots into clusters. A distance based clustering [] is used to group shots that are similar into a cluster. Clustering is optimized by incorporating splitting and merging procedures, with the objective of creating different clusters with sufficiently different characteristics or features.

Video Shot Clustering

Clustering based on distance similarity aims at maximizing intra cluster and minimizing inter cluster similarities. The merge and split procedures employed in [21–23] aim to achieve the optimal number of clusters meeting the basic principle. The merge procedure to start with treats every shot as a cluster and iteratively merges or combines most similar clusters until the distance between the most similar clusters exceeds a specified threshold. This effectively reduces the number of clusters subject to the condition that the mered clusters are the least dissimilar pair among all clusters. Two clusters are merged if their intra cluster similarity is high.

Merging could result in clusters with large intra cluster distances and hence is followed up by a splitting operation. Splitting is employed to split or divide clusters with large visual variances. The split procedure fine tunes the clusters obtained at the end of the merging phase, dividing a cluster into two subclusters (new clusters) and hence effectively increasing the number of clusters. For a specific cluster with maximal intra cluster distance, two member shots with the largest distance are identified and assigned to two different clusters. Remaining shots of the original cluster are assigned to the two new clusters based on their cluster distances. Intracluster distance for a specific cluster is iteratively computed and the one that exceeds a specified threshold is split. This procedure is repeated until all clusters's intra cluster distance does not exceed the specified threshold.

By the end of the clustering phase, shots are assigned labels and a shot cluster sequence is constructed by sequentially aggregating the class information (labels) of each shot by its original temporal order. Reference [21–23] extends preprocessing phase by incorporating video text detection (textual analysis of video), camera motion characterization and audio event detection strategies, resulting in a hybrid stream or sequence. Our thrust area of research is the association mining phase of video association mining, concentrating on efficient frequent pattern mining algorithms for the same. The association mining and hence frequent pattern mining scheme of the existing video association mining prototype is based on Apriori's level wise principle and hence suffers from the

inherent repeated scans limitation.

7.16 Applications of Video Associations

Video Association Mining finds varied applications ranging from video summarization to event detection and retrieval operations. A summary of the input video can be generated by including the most frequent patterns or sequences. Clusters that appear in longer associations with strong support and confidence convey important scenarios in the video and hence are included in the summary. Association based summarization could present the viewer with visually concise and semantically related frames from the clusters that address the important video scenarios.

Video associations are also employed to detect events and special patterns. Association mining helps in establishing associations among the video units and hence detect special patterns and events. The detected associations are as well employed for retrieval tasks. Conventional retrieval techniques present the user with units that have highest similarity with the specified query, resulting in the situation that the presented units are isolated from their context and scenarios. Thus retrieval techniques based only on visual similarity might not be able to convey content and scenario information. Video associations can be applied in retrieval tasks wherein in addition to displaying the units with highest visual similarity, other units that have the highest association with each retrieved unit are presented to the user. This results in a more content and context specific retrieval scheme.

In the context of video data, one can establish associations between the various objects in a frame that are representative of a scene. Associations can be established at higher levels of abstraction to identify relationship between the director, movie type, movie plot, etc. A real time application of the generated associations would be to predict futuristic events based on the occurrence of a certain sequence of events frequently. Crowd trouble could be predicted in cases of presence of events such as communal rallies, inflammatory political speeches and religious processions.

Reference [4] identifies the different types of knowledge that can be mined from cinematic data. Interesting events (event detection) such as emotions, mood can be mined from the video data. Events such as acts of crying indicating sadness, laughter indicating happiness and bomb blasts indicatory of street violence can be used in association patterns to predict future course of events. Associations or event patterns such as acts of violence are followed by sad scenes can be mined. Classification is another area where generated associations can be employed. The mined frequent patterns or events could be classify videos. Events such as gunshots, violence, bomb blasts followed by sadness could be used to classify videos as being violent while laughter, happiness could be indicators for the movie type being romantic or comic, etc.

7.17 Frequent Temporal Pattern (FTP) Mining

Frequent Temporal Pattern (FTP) mining, much like conventional frequent item-set or pattern mining is an important phase of video association mining. It is the process of generating all possible frequent patterns subject to the temporal distance and support thresholds discussed in the earlier chapter. In a way, FTP mining is nothing but the conventional process of frequent pattern mining incorporating the temporal factors. This section discusses FTP mining and the existing Apriori based algorithm for mining frequent temporal patterns. Apriori has been established to suffer from the repeated scans limitation and the thesis proposes novel algorithms that overcome this setback of Apriori in the domain of FTP mining.

Frequent item-set or pattern mining (FPM) is essentially the predecessor to sequence pattern and frequent temporal pattern mining. FPM is an essential phase of conventional association rule mining and research over the years primarily concentrated on this aspect, resulting in the evolution of several efficient algorithms[24–35]. The Frequent Pattern (FP) tree based algorithm [24] requires only two overall scans of the original input and adopts a tree or pattern growth approach. It is not suited to FTP mining since it loses track of the ordering of symbols within a pattern. Patterns AB and BA are identical in frequent pattern mining but in FTP mining they are treated as different patterns. The Dynamic Item-Set Counting approach [36] reduces the number of scans by constructing frequent sets in a simultaneous fashion. Frequent set mining algorithms discussed in [25–28] either lack the temporal or ordering property or require several repeated scans of the input sequence when used for FTP mining and hence there is an impending research requirement for efficient algorithms for the same.

Temporal Data Mining which concentrates on data mining incorporating application specific temporal properties has been an active field for some time now [37]. A progressive partition miner that mines association rules for a publication database incorporating temporal aspect in the form of exhibition period is proposed by Lee et.al [38]. It employs a Apriori level wise principle logic in the process of frequent patterns construction. References [39–42] propose Apriori based algorithms for the frequent pattern mining phase in temporal domain. These algorithms inherently suffer from the repeated scans limitation of Apriori and do not suit our requirement for efficient algorithms for FTP mining phase of Video Association Mining.

FTP mining, can also be treated as Sequence Pattern Mining(SPM). GSP [43], an apriori principle based SPM algorithm does not meet our requirement of avoiding the huge repeated scans setback. A few of the other SPM algorithms avoiding the repeated scans setback of Apriori are FreeSpan [44], PrefixSpan [45] and SPADE[46]. FreeSpan and PrefixSpan adopt a pattern growth approach similar to FP growth but incorporate ordering aspect in the mining process. The database projection logic used in these algorithms do require repeated scans, but it is significantly lesser than Apriori. Also the scans are not over the entire database but over a trimmed version. These algorithms are not suited

Table 7.1: Level Wise Execution of Apriori Algorithm

Input Sequence	L_1	C_2	L_2	C_3	L_3
				ABC(4)	
		AA(2)		ACB(4)	
		AB(5)	AB(5)	ACC(3)	
A B C B C		AC(4)	AC(4)	BAC(3)	
C A C B C	A(5)	BA(4)	BA(4)	BCA(3)	ABC(4)
C A B C A	B(6)	BB(3)	BC(6)	BCC(3)	ACB(4)
B A C B C	C(9)	BC(6)	CB(5)	CBA(3)	CBC(4)
		CA(3)	CC(4)	CBC(4)	
		CB(5)		CCB(3)	
		CC(4)		CCC(3)	

to FTP mining, since they require the input to be in the form of transactional records. Transactionalizing the input sequence to a record format results in loss of temporal continuity across records. Hence there is an impending research requirement of efficient algorithms for mining frequent temporal patterns.

7.17.1 Apriori based FTP Mining Algorithm

FTP mining though similar to the frequent set mining phase of conventional association rule mining differs from its transactional counterparts in the temporal support and distance factors. References [25–28] provide a detailed survey of algorithms for frequent set mining in conventional data domain. The existing video association mining technique employs Apriori algorithm [47] in the process of FTP mining. Apriori, the first algorithm for frequent set mining is based on a level wise principle and the anti-monotone property of set theory that "Every subset of a frequent set is also frequent".

Apriori based FTP mining algorithm [21–23] first constructs frequent patterns (item-sets) of length 1 or L_1 from candidate patterns of length 1 or C_1 (all possible unique symbols in the input sequence). It then generates higher level candidate patterns (C_i) from immediate previous level frequent patterns or L_{i-1}. Thus C_2 is generated by self joining L_1 with itself (i.e $C_2 = L_1 * L_1$). Subsequent frequent patterns (L_i *or* L_2) are generated by subjecting the corresponding level candidate patterns to the support factor. Since a complete scan of the database is required for a candidate set to be identified as frequent or infrequent, the number of repeated scans of the original database during the entire frequent pattern mining process is huge and is a serious limitation. A stepwise execution of the Apriori based algorithm for the sample input sequence A B C B C C A C B C C A B C A B A C B C with TD=∞ and TS=4 is shown in Table 7.1. References [48–50] address this aspect of eliminating the repeated input scans limtiation of Apriori in the process of generating frequent patterns subject to temporal constraints.

7.18 Text Data Mining or Text Mining

Previous sections in the chapter discussed multimedia data mining primarily concentrating on mining knowledge from image and video data. Textual data also constitutes a major form of multimedia input and the amount of textual data available outsizes its image and video counterpart primarily due to the fact that textual data forms a part and parcel of most inputs (multimedia). Also amongst the various multimedia input forms, text is the most simplest and conventional databases require only slight extensions, if not none to represent text data. This and sections to follow discuss text mining detailing the issues involved and types of knowledge that can be mined from textual data.

Textual data has and is always on the rise coming from various sources such as documents, articles, electronic mail messages, electronic books, etc. In this modern age of electronic everything, text data has become essential to input representation and the amount of such data available is extremely large and the need to extract information from them is driving and imminent. Textual data is semi structured in the sense that it is neither completely structured nor is it completely unstructured. A document could be described by attributes such as author, title, date of publication, etc. Information retrieval techniques that retrieve documents based on user queries do more of a syntactic comparison and the result of such queries is often too small a number. The potential to extract hidden patterns and knowledge in textual data is vast and forms the crux of text data mining.

7.19 Information Retrieval from Text Databases

Information retrieval concerns the organization and retrieval of information from huge and voluminous textual data or documents. It aims at retrieving or returning relevant documents based on user keywords. Applications include library management system, which supports operation s such as searching for books by title, author, year of publication, etc. The efficiency and performance of text data analysis systems is generally evaluated using two basic measures namely precision and recall.

Precision typically indicates the correctness of the responses and represents the percentage of retrieved documents that are infact relevant or matching to the input query. Assuming documents relevant to a query is represented by RELEVANT and the documents returned as results for a user query is RETURNED, precision of a text data analysis system is expressed as shown in Equation 7.1.

$$precision = \frac{|\{RELEVANT\} \cap \{RETURNED\}|}{|\{RETURNED\}|} \tag{7.1}$$

Recall measure of a text analysis system denotes the percentage of documents that are relevant to a query and were returned as results to the user query. It

is expressed as shown in Equation 7.2.

$$recall = \frac{|\{RELEVANT\} \cap \{RETURNED\}|}{|\{RELEVANT\}|} \tag{7.2}$$

Thus precision denotes the capability of the system to return documents matching a query while recall denotes the ability of the system to return relevant documents for a given input query. Keyword or syntactic matching forms an essential phase of most text analysis systems. Documents are generally represented by string or keywords such as cricket, world-cup, etc. These keywords are infact explicitly specified by the user or extracted from the user query and then the database of textual data is compared for the occurrence of the specified keywords.

Techniques that perform such a keyword based information or document retrieval system are referred to as syntactic systems, in the sense that they look out only for the occurrence of the keywords with no emphasis on the context or the semantics of the process. For example a keyword search such as airways might retrieve all documents that contain the keyword airways. But realistically there is the possibility of related and relevant documents that contain a different word (but still mean the same) such as air-travel. Hence syntactic systems cannot be a final alternative to information retrieval and it is in this aspect that text mining plays a vital role. In fact information retrieval could be viewed as a database retrieval task while semantic oriented text systems could be treated as text mining prototypes.

Two key issues involved with text data processing are synonymy and polysemy that could alter the results returned in a major way. Synonymy refers to the situation of a keyword occurring in various alternate forms such as the one discussed above. Polysemy refers to the situation of a single keyword referring to completely and entirely different perspectives such as a word bank that could be correlated to the conventional money management organization or relate to word trust or a river bank. It is entirely dependent on and dominated by the context in which it is being used. Similarity base retrievals operate on the principle of assigning relevance scores to documents and finally returns those documents for whom relevance score for a specific user query is large or exceeds a specified threshold.

Most text analysis and information retrieval systems based on keyword similarity employ the process of identifying stop and stem words in documents. This phase actually helps in eliminating the frequently occurring and insignificant words in the document and thereby reduce the string representation of the same. Stop words are commonly used english words such as a , an, the, of, for, etc. which serve as articles and prepositions in the process of creating meaningful sentences but as it is do not carry significant meanings with them. Stem words are words that forms the root for the varied forms of the word such as running, ran whose stem word is run. The stop and stem words list basically helps in reducing the document size and representation.

Having identified stop and stem words, the text analysis system then represents the various documents in the form of a term document matrix, terms

Term/Doc	d_1	d_2	d_3	d_4	d_5	d_6
t_1	320	50	15	58	80	13
t_2	340	65	20	62	74	8
t_3	12	28	220	320	64	22
t_4	21	145	390	420	55	43

Table 7.2: A sample term document frequency matrix

constituting the columns and documents the rows. A value of 0 indicates the non occurrence of the term in the document and non zero values indicate the frequency of occurrence. Also other representations such as a value 1 indicating the presence and 0 indicating the absence of the term in the document or relative frequency of the term in comparison to all the terms in the document is also followed. An example term document matrix representation is as shown in Table 7.2.

Similarity of documents is measured by measuring the similarity of term document frequency matrix values and the cosine measure where the cosine similarity between two document vectors dv_1 & dv_2 is measured using the formula $sim(v_1, v_2) = \frac{v_1 . v_2}{|v_1||v_2|}$. The term document frequency matrix notation for a real time text database could be extremely large as a result of large number of terms and documents and thereby increases the complexity of systems that process text to satisfy various user queries. It is common practice to employ standard data reduction techniques such as Latent Semantic Indexing which aims to capture the key relationships in documents and reduce the term document matrix size for further efficient processing.

7.20 Latent Semantic Indexing

Latent Semantic Indexing as a text processing technique aims to reduce the dimensions involved in the term document frequency matrix representation and uses the singular value decomposition . The technique aims at bringing down the size of the term document matrix. Given a matrix t*d term document frequency matrix, the singular value decomposition method eliminates rows and columns to bring down the matrix size to a relatively small s*s. The elimination is done in a fashion that only insignificant parts that convey the least information in the term document frequency matrix are removed and significant parts that convey main information in the document are retained. The singular value decomposition as a data reduction technique has been discussed in Chapter 2 on preprocessing techniques. Key phases of the latent semantic indexing method are as follows:

1. To start with an initial frequency matrix is created.

2. Singular value decompositions of the above matrix is created by splitting the original matrix into three smaller matrices, U, S & V where U and V

are orthogonal matrices (i.e $U^TU = I$, I being the identity matrix) and S is a diagonal matrix composed of singular values. The reduced matrix size is K*K.

3. Every document's document vector is replaced by the new one which eliminates the terms left out by the singular value decomposition.

4. The replaced vectors are organized and indexed using indexing techniques.

Other indexing technique such as inverted index, signature file are also used for organizing and manipulating the text databases in an efficient manner. The inverted index structure maintains two hash indexed tables namely document table and term table. The system supports efficient querying of applications such as finding all documents associated with a set of terms or finding terms given a set of documents. The document table consists of a set of document records each of which are composed of docid and postings list which contains the list of terms in the document, sorted based on some relevance measure. The indexing structures though easy to implement cannot handle issues of synonymy and polysemy and also the postings list could be extremely space consuming. Another alternate representation is the signature file that stores signature or key records for documents in the database. Signature terms are of b bits long and initialized to 0. Signature matches are treated as similarity at the bit level. Indexing structures are generally applied after the stop and stem word analysis.

7.21 Text Mining

Text Mining or Text Data Mining is the process of extraction of hidden knowledge and useful information from large text databases. It is the correlation of conventional data mining techniques such as association mining, classification, clustering, etc. over text databases. Association mining finds application in establishing associations between the terms and keywords which are used in automatic tagging of documents and minimizing the effect of insignificant parts of the document on the performance of the text processing system.

In the context of textual data, association mining treats documents as transactions and the the various terms involved in the document as items of the transaction. Thus the problem of association mining concentrates on identifying relationships between terms, documents and keywords. Association mining that finds in ultimate classification systems is also a possibility in textual databases. The discovered associations are employed in the process of classifying documents. Also traditional classification techniques that construct a model for the input database and use it to test new documents is employed. Text classification systems are generally based upon the capability of the terms and keywords in distinguishing documents. Clustering is also employed in the process of grouping related and similar documents. Clustering optimizes the principle of maximizing the similarity of documents within a cluster or group.

7.22 Web Data Mining

The World Wide Web hosts a huge collection of documents that are used by varied users across the internet and world wide. The amount of data available over the internet has seen a tremendous increase in the recent years. The web contains collection of documents that are appropriately linked and access cum usage information. All these data throw the imminent challenge of extracting useful knowledge and patterns from them. The technique that deals with knowledge extraction from web data is referred to as Web Data Mining.

Web data mining or web mining as it is popularly referred to encounters far more complicated and complex issues in relation traditional data mining or data mining over conventional database operations. The web is often too large and huge for a data warehouse construction and hence the support of mining operations via it is not realistic. The very nature of the web makes it a dynamic data source, with not only the internet growing in terms of user community but also web pages or documents undergoing constant updation in due course of time. Also web mining faces the challenge of presenting the actual web pages of interest and use to the user and avoid returning a huge collection of documents which are in no way related to the user's interest area.

One of the fundamental tasks over the internet and the world wide web is searching for specific pages based on keywords provided by the user. One such famous site is google that supports keyword based document location over the web. As they put it today, anything you need or any data you need can always be located over the internet and the gateway to the data if you are unsure of its location is via google! Searching over the web is an art and is mastered over a period of time. Experienced users might be able to retrieve documents much faster compared to novice users.

The quickness with which documents are located is directly dependent on how accurate the keywords or search terms have been specified. The polysemy concept where the same term related to different interpretations depending on the context in which it is used also affects web searching. Search for documents containing keyword "data mining" could in addition to relevant pages or documents also return documents related to other industry (such as coal mining) related data. The other issue of synonmy in which case a term is referred to by several other alternate terms such as knowledge discovery for data mining. In such situations documents containing the word knowledge discovery for data mining might not get returned in the resultant set of documents when searching for data mining. Thus the process of searching must be intelligent and semantic oriented rather than being entirely based on syntactic keyword matching strategies.

Web mining supports additional features in relation to web searching. The nature in which the webpage are accessed, also referred to as web access pattern mining, the way in which the web documents are used or web usage mining and web content or structure mining are some of the emerging data mining trends in the domain of web data.

7.23 Authoritative Web Pages Identification

Authoritative web pages are defined as those web paegs or documents that are relevant as well as of the highest quality in terms of maximal relevance. The web structuring is such that a page or document often links to another pages(s). This inherent linking structures available in web pages consolidate the notion of authority for web pages. A link to another page B from page A is treated as an explicit endorsement or certification on B's relevance to contents represented by A. Thus the link structures in web pages help in resolving the issue of authoritative web pages identification. Two types of paegs based on relevance and authority are

1. authorities which provide the best source of information on a given topic (search topic)

2. hubs, which provide collections of links to authorities

A key issue related to efficient identification of relevant pages is that the authorities are hardly self descriptive, in the sense that pages do not contain descriptive information about themself. Thus a company such as Maruti Udyog Limited is hardly going to contain the term Indian Automobile Manufacturers. Thus an entirely text based or syntactic strategy might not be a wise approach for identifying relevant pages. Also from the hubs point of view, not all pages linked to by a page are entirely relevant to the base page. The classical case of advertiser's (sponsor's) pages such as Air Travel agencies web pages linked to from an email page is actually in no way related to the main page.

One of the key characteristics exhibited by authority and hubs is that they are interrelated. A good authority is a page that is pointed to by many good hubs while a good hub is a page that points to many good authorities. Hyper Link Inducted Topic Search or HITS as it is popularly called is a search technique that bases its search based on concepts of authority and hubs discussed earlier. It computes lists of hubs and authorities for web search topics or keywords. Two key phases of the HITS technique are sampling and weight propagation. The sampling component constructs a focussed selection of several web pages that are likely to be rich in relevant authorities. The weight propagation phase estimates numerical values for hub and authority weights on an interactive basis. Pages which have the highest weights are returned as hubs and authorities for the given search topic.

The web is treated as a directed graph composed of nodes and directed edges between them. Given a subset of nodes, the nodes induce a subgraph SG containing all edges that connect any two nodes in SG. Objective of HITS is to construct a subgraph that is composed of rich and relevant set of web pages. To start with the query terms are used to collect an approximate or rough number of root set of pages using an index based search engine.

Since most of the pages are assumed to be topic relevant, the root set of pages is then expanded into a base set of pages which includes all pages linked by the root pages. It is from this base set of pages that the authoritative pages

are located. One fine tuning strategy adopted is that links between two pages that share the same web domain for navigational purpose are not treated as authorities. Every page p is assigned non negative authority weight x_p and hub weight y_p. The actual magnitude of these weights is insignificant to the process of locating authoritative pages. The sum of these weight values are bound by normalization. A page p with a large weight x_p is treated as a better authority while a page with large weight y_p is treated as a better hub.

The various weights are initially set to constant values and are updated as follows. In cases of a page being pointed by many good hubs, its authority weight is increased. Thus the weight value for x_p for page p is updated as the sum of y_q over all pages q that link or refer to p. Thus $x_p = \sum_{q|q\to p} y_q$, where $q \to p$ denotes the fact that page q links to page p. Similarly if a page points to many good authorities, the hub weight is incremented as $y_p = \sum_{q|p\to q} x_q$. An alternate efficient and compact representation of the weight updates is the matrix notation. Pages are numbered $1 \ldots n$ and A defines an n*n adjacent matrix whose $(i,j)^{th}$ entry is set to 1 if page i links to page j and 0 otherwise.

Possible values for x and y are represented as vectors $x = (x_1, x_2, \ldots x_n)$ and y=$(y_1, y_2, \ldots y_n)$. With such a representation in place the weight updates are now represented as $x \leftarrow A^T y$ and $y \leftarrow AX$. Thus $x \leftarrow A^t y \leftarrow A^T AX = (A^T A)x$ and $y \leftarrow Ax \leftarrow (AA^T)y$. Final value for vector x is the result of power iteration on $A^T A$. It is inferred from linear algebra that $A^T A$ and AA^T when normalized converge to their respective principal eigen vectors. This forms the mathematical basis for HITS establishing the significance of pages and links.

The system returns list consisting of pages with largest authority weights for a given search topic. The query terms (text based search topic) is required only for root set construction, after which searching is entirely link based computation oriented and is more semantical compared to traditional text only based search engines. Despite its semantic and intelligent search advantages offered, HITS does suffer from certain bottlenecks. Since all links out of a hub page propagate the same weight, certain pages deemed as authoritative by HITS might not actually be. As an example pages on data mining experts (home pages of resource persons) might have links to personal data related resource pages. In such situations the entire set of links returned as authoritative might fall of from the user's requirements.

7.24 Correlation of Data Mining Techniques in Web domain

Web data mining or web mining is formally defined as the correlation or adaptation of conventional data mining techniques such as association mining, classification, clustering, etc. in the domain of web data. Classification could be used to classify web pages or documents into one of the predefined class labels. Examples could be classifying web pages as academics, industry, scientific, research oriented, etc. Association mining could also be employed to estab-

lishing associations and relationships between web pages and documents. It helps in identifying related or associated and frequently visited web pages. This knowledge could be employed in classification based on association rule based classification systems. A certain set of pages visited by a user could be used in predicting or classifying the web pages and documents.

Web usage mining helps in revealing the knowledge hidden in the log (user login details) files of a web server. Interesting patterns concerning the users navigational behaviour can be identified by applying data mining methods over the web data. User and page clusters and possible correlations between web pages and user groups can be established. The web usage mining is generally composed of three phases namely data preparation, pattern discovery and pattern analysis phases. Web log data are preprocessed inorder to identify users, sessions, page views and so on. Next statistical data mining methods such as association rules, sequential pattern discovery, classification and clustering are applied over the preprocessed data to discover knowledge or patterns.

Association rule mining is used to reveal correlations between pages accessed during a server session. These relations indicate the possible relationship between pages that are often viewed together even if they are not directly connected. The extracted knowledge helps in efficient design and restructuring of the website. Clustering is used to group related web pages and users together. Page clustering identifies groups of pages that are conceptually related while user clustering groups together similarly behaving (in terms of web navigation) users. Individual user categories can also be mined using the data classification technique which assigns class labels (categories) to the user data captured on the web.

7.25 Sequential Pattern Mining

Sequence Pattern Mining (SPM) is the process of discovering relationships between occurrences of sequential events. SPM differs from FPM in the fact that ordering of patterns is crucial in SPM. Patterns BA and AB are treated as being identical in FPM while they are considered as different patterns in SPM. It is the process of extracting all possible sequential patterns that exceed the specified minimum support threshold.

One of the basic algorithms for SPM is based on the Apriori logic proposed for FPM. Candidate and subsequent frequent sequences are generated in a level wise manner. GSP[43], AprioriAll[21] are a few of the apriori logic based sequential pattern mining approaches. These algorithms suffered from the repeated scans limitation of Apriori, an issue that led to the evolution of pattern growth algorithms for SPM much like FP growth for FPM. FreeSpan & PrefixSpan overcome the repeated scans limitation of Apriori based algorithms. They are an extension of the FP growth algorithm to mine frequent sequences, incorporating the ordering of symbols within a sequence or a pattern.

FreeSpan[44] avoids the repeated scans of the entire database by recursively projecting a sequence database into sets of smaller databases associated with

the patterns that have been generated so far. PrefixSpan[45] extends FreeSpan by reducing the size of the projected databases and avoiding the situation of having to check for every possible candidate sequence. It first scans the sequential database once to generate frequent sequence of length 1 or L_1. Then the database is split into different partitions, each partition being a projection of the sequence database that has the corresponding length 1 sequence as prefix. The projection process is iterated until the projected database becomes empty or no new frequent patterns can be generated. SPADE, SPIRIT are a few of the other algorithms proposed to overcome Apriori's limitation in the domain of sequence pattern mining.

Time series data mining, another branch of data mining is similar to sequence pattern mining, however it differs in the aspect of temporal or time related characteristics dominating the process of mining patterns or knowledge. Patterns discovered by time series methods require the temporal or time aspect to be compulsorily incorporated while sequence pattern mining could also be correlated in non temporal domains where explicit temporal properties are not required. Subsequence and whole sequence matching form a few of the key phases of similarity matching in time series data.

Time series data are generally subject to transformation to convert the data from time to frequency domain and then employ matching techniques to retrieve patterns. Distance preserving and data independent transformations such as discrete Fourier and wavelet transforms are employed to perform the conversion from time to frequency domain. Indexing strategies such as R trees are used in the process of similarity matching . Also window based techniques where the data is split into smaller pieces or windows of a specific length and then matching is performed by window sliding techniques. We have only given a short introduction to the advanced issues in data mining and interested readers must pursue further publications and materials on the emerging techniques.

Summary

Multimedia Data Mining, the recent research trend in data mining is an emerging field and techniques for the same are in the evolution phase with potential for quality and dedicated research on the same. The chapter discussed few of the existing literatures on multimedia data mining addressing text, images and video data and knowledge extraction from them. MDM differs from its conventional counterpart in the presence of data specific properties and the entire objective of MDM is to make the process of data mining address these properties. This chapter could in no way be treated as a definitive material on MDM and should be followed up with extensive referral to recent publications. The aim has been to introduce the readers to the concept of data mining over multimedia data. MDM is the future of knowledge extraction for the simple reason that from now on data cannot exist in a simple and unified format. Real time data is bound to become more complex as a result of more and more sophisticated devices, methods of recording data and hence the issues in MDM is bound

to be on the rise and crucial

Review Questions

1. Define Multimedia Data Mining and issues involved with the same.
2. Define Recurrent features mining and issues involved in mining recurrent objects from images.
3. Discuss the different types of knowledge that can be mined from Image data.
4. Define Image Association Mining and discuss techniques and applications of the same.
5. Define Video Mining and identify the different types of knowledge that can be mined from video data.
6. Discuss the HITS strategy to searching for documents over the web.
7. Compare and Contrast the various sequential pattern mining algorithms. (refer external source)
8. What is Web Personalization and correlate the techniques discussed in web mining for the same.
9. Survey the various search engines supported such as Google, Lycos, Yahoo, etc. based on features such as user friendliness, efficiency of retrieval, relevance of pages returned, etc. (assignment, refer external sources)

Bibliography

[1] M. Mehta, R. Agrawal, and J. Rissanen. SLIQ:A Fast Scalable Classifer for Data Mining. In *5^{th} International Conference on Extended Database Technology: Advances in Database Technology*, pages 18–32, 1996.

[2] R. Agrawal, S. Ghosh, T. Imielinski, B. Iyer, and A. Swami. An Interval Classifier for Data Mining Applications. In *18^{th} International Conference on Very Lareg Data Bases (VLDB)*, pages 23–27, 1992.

[3] W.Perrizo, W.Jockheck, A.Perera, D.Ren, W.Wu, and Y.Zhangi. Multimedia Data Mining using P trees. In *International Workshop on Multimedia Data Mining (MDM/KDD 2002)*, pages 19–29, 2002.

[4] D.Wijesekara and D.Barbara. Mining Cinematic Knowledge–Work in Progress. In *International Workshop on MDM/KDD*, pages 98–103, Aug 2000.

[5] J. Oh and B. Bandi. Multimedia Data Mining framework for raw video sequences. In *International Workshop on Multimedia Data & Management (MDM-KDD)*, pages 1–10, 2002.

[6] JungHwan Oh, JeongKyu Lee, and Sanjaykumar Kote. Real Time Video Data Mining for Surveillance Video Streams. In *7th Pacific-Asia Conference, PAKDD 2003*, pages 222–233, 2003.

[7] X. Zhu, W G. Aref, J. Fan, A C Catlin, and A K Elmagarmid. Medical Video Mining for Efficient Database Indexing, Management and Access. In *International Conference on Data Engineering (ICDE)*, pages 569–580, 2003.

[8] X. Zhu, W G. Aref, J. Fan, and A K Elmagarmid. ClassMiner:Mining medical video content structure and events towards efficient access and scalable skimming. In *ACM SIGMOD Workshop*, pages 9–16, 2002.

[9] King-Shy Goh, Koji Miyahara, Regunathan Radhakrishan, Ziyou Xiong, and Ajay Divakaran. Audio-Visual Event Detection based on Mining of Semantic Audio-Visual Labels. In *Conference on Storage & Retrieval for Multimedia Databases*, volume 5304, pages 292–299, 2003.

[10] Ajay Divakaran, Koji Miyahara, Kadir A. Peker, Regunathan Radhakrishnan, and Ziyou Xiong. Video Mining Using Combinations of Unsupervised and Supervised Learning Techniques. In *SPIE Electronic Imaging Conference on Storage and Retrieval for Media Databases*, volume 5307, pages 235–243, 2004.

[11] Cheng Lu, Mark S. Drew, and James Au. Classification of Summarized Videos Using Hidden Markov Models on Compressed Chromaticity Signatures. In *ACM Multimedia*, pages 479–482, 2001.

[12] Li-Qun Xu and Yongmin Li. Video Classification Using Spatial-Temporal Features and PCA. In *ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo - Volume 3 (ICME '03)*, pages 485–488, 2003.

[13] Xavier Gibert, Huiping Li, and David Doermann. Sports Video Classification Using HMMS. In *ICME*, 2003.

[14] Kim Shearer, Chitra Dorai, and Svetha Venkatesh. Incorporating Domain Knowledge with Video and voice data analysis in news broadcasts. In *International Workshop on Multimedia Data Mining(MDM/KDD 2000)+ACM SIGKDD conference*, pages 46–53, 2000.

[15] Zeeshan Rasheed and Mubarak Shah. Movie Genre Classification By Exploiting Audio-Visual Features Of Previews. In *16^{th} International Conference on Pattern Recognition*, pages 1086–1089, 2002.

[16] Stephan Fischer, Rainer Lienhart, and Wolfgang Effelsberg. Automatic Recognition of Film Genres. In *3^{rd} International ACM Conference on Multimedia*, pages 295–304, 1995.

[17] Lihi Zelnik-Manor and Michal Irani. Event-Based Analysis of Video. In *Computer Vision and Pattern Recognition*, 2001.

[18] W. Aref, A. Catlin, A. Elmagarmid, J. Fan, M. Hammad, I.Ilyas, M. Marzouk, and X. Zhu. A Video Database Management System for Advancing Video Database Research. In *International Workshop on Management Information Systems*, 2002.

[19] Pimwadee Chaovalit and Lina Zhou. Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches. In *38^{th} Annual Hawaii International Conference on System Sciences - Track 4*, 2005.

[20] Jia-Yu Pan and Christos Faloustos. GeoPlot:Spatial Data Mining on Video Libraries. In *11th International conference on Information and knowledge management*, pages 405–412, 2002.

[21] X.Zhu and X.Wu. Mining Video Associations for Efficient Database Management. In *18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1422–1432, 2003.

[22] X.Zhu and X.Wu. Sequential Assocaition Mining for Video Summarization. In *IEEE International Conference on Multimedia & Expo.*, volume 3, pages 333–336, 2003.

[23] X.Zhu, X.Wu, and A.K.Elmagramid et.al. Video Data Mining : Semantic Indexing and Event Detection from the Association Perspective. *IEEE Transactions of Knowledge and Data Engineering*, 17(5):665–677, 2005.

[24] J.Han, J.Pei, and Y.Yin. Mining Frequent Patterns without Candidate Generation. In *ACM SIGMOD International Conference on Management of Data*, pages 1–12, 2000.

[25] Bart Goethals. Survey on Frequent Pattern Mining. In *Helsiniki 2004:43*, 2003.

[26] Bart Goethals. *Efficeint Frequent Pattern Mining.* PhD thesis, Trasnational University of Limburg, Belgium, 2002.

[27] Jian Pei. *Pattern Growth Methods for Frequent Pattern Mining.* PhD thesis, Simon Fraser Univeristy, 2002.

[28] O.R. Zaiane and Mohammad El-Hajj. Advances and Issues in Frquent Pattern Mining. In *8th Pacific Asia Conference on Knowledge Discovery & Data Mining (PAKDD'04)- Tutorial Notes*, 2004.

[29] Rajanish Dass and Ambuj Mahanti. An Efficient Technique for Frequent Pattern Mining in Real-Time Business Applications. In *38th IEEE Hawaii International Conference on System Sciences.(HICSS 38)*, 2005.

[30] Mohammad El-Hajj and Osmar R. Zaiane. COFI-tree Mining: A New Approach to Pattern Growth with Reduced Candidacy Generation. In *Frequent Itemset Mining Implementation (FIMI) Workshop*, 2003.

[31] William Cheung and Osmar R. Zaiane. Incremental Mining of Frequent Patterns Without Candidate Generation or Support Constraint. In *7th International Database Engineering and Applications Symposium (IDEAS 2003)*, pages 111–116, 2003.

[32] Georges Gardarin, Philippe Pucheral, and Fei Wu. Bitmap Based Algorithms For Mining Association. In *International BDA Conference*, pages 157–175, 1998.

[33] J. Mata, J.L. Alvarez, and J.C. Riquelme. An Evolutionary Algorithm to Discover Numeric Association Rules. In *2002 ACM symposium on Applied computing*, pages 590–594, 2002.

[34] Wai-Ho Au and Keith C.C Chan. FARM: A Data Mining System for Discovering Fuzzy Association Rules. In *Advances in evolutionary computing: theory and applications*, pages 819–845, 2003.

[35] Osmar R. Zaiane, Mohammad El-Hajj, and Paul Lu. Fast Parallel Association Rule Mining Without Candidacy Generation. In *IEEE 2001 International Conference on Data Mining (ICDM'2001)*, pages 665–668, 2001.

[36] S.Brin, J.Ullmann, R.Motwani, and S.Tsur. Dynamic Itemset Counting. In *ACM SIGMOD International Conference on Management of Data*, pages 255–264, 1997.

[37] Claudia M. Antunes and Arlindo L Oliveira. Temporal Data Mining: An Overview. In *International Workshop on Temporal Data Mining*, pages 1–13, 2001.

[38] Chang-Hung Lee, Cheng-Ru Lin, and Ming-Syan Chen. On Mining General Temporal Association Rules in a Publication Database. In *International Conference on Data Mining (ICDM)*, pages 337–344, 2001.

[39] Francisco Guil, Alfonso Bosch, Antonio Bailn, and Roque Marn. A Fuzzy Approach for Mining Generalized Frequent Temporal Patterns. In *IEEE ICDM 2004 Workshop on Alternative Techniques for Data Mining and Knowledge Discovery*, 2004.

[40] Xingzhi Sun, Maria E. Orlowska, and Xiaofang Zhou. Finding Event-Oriented Patterns in Long Temporal Sequences. In *International Conference on PAKDD 2003*, pages 15–26, 2003.

[41] Francisco Guil, Alfonso Bosch, and Roque Marin. TSET: An Algorithm for Mining Frequent Temporal Patterns. In 1^{st}*International Workshop on Knowledge Discovery in Data Streams*, 2004.

[42] Balaji Padmanabhan and Alexander Tuzhilin. Pattern Discovery in Temporal Databases: A Temporal Logic Approach. In 2^{nd}*International Conference on Knowledge Discovery and Data Mining*, 1996.

[43] R.Srikant and R.Agrawal. Mining Sequential Patterns – Generalizations and Performance Improvements. In 5^{th}*International Conference on Extending Database Technology (EDBT)*, pages 3–17, Mar 1996.

[44] J.Han and J.Pei et.al. Freespan: Frequent Pattern Projected Sequential Pattern Mining. In *ACM SIGKDD International Conference on Knowledge Discovery in Databases*, pages 355–359, 2000.

[45] J.Han and J.Pei et.al. Mining Sequential Patterns by Pattern–Growth:The PrefixSpan Approach. *IEEE Transactions of Knowledge and Data Engineering*, 16(11):1424–1440, 2004.

[46] M.Zaki. SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning*, 40:31–60, 2001.

[47] R.Agrawal and R.Srikant. Fast Algorithms for Mining Association Rules. In *International Conference on Very Large DataBases (VLDB)*, pages 487–499, 1994.

[48] B.SivaSelvan and Ilango Krishnamurthi. An Efficient Video Association Mining Algorithm. *Journal of Computer Society of India*, 35(3):56–67, 2005.

[49] B.SivaSelvan and N.P. Gopalan. An Efficient Frequent Temporal Pattern (EFTP) Mining Algorithm. *Information Technology Journal*, 5(6):1043–1047, 2006.

[50] N.P. Gopalan and B. SivaSelvan. An m-ary tree based Frequent Temporal Pattern (FTP) mining Algorithm. In *6^{th} IEEE INDICON International Conference on Emerging Trends in Information & Communication Technology*, Sep 2006.

[51] Jiawei Han and Micheline Kamber. *Data Mining:Concepts and Techniques*, chapter 1–9. Morgan Kauffman, 2001.

[52] O.R.Zaiane, J.Han, Z.Li, and J.Chiang. Multimedia Miner–A System Prototype for Multimedia Data Mining. In *ACM SIGMOD International Conference on Management of Data*, pages 581–583, 2002.

[53] O.R.Zaiane, M.L Antonie, and A.Coman. Mammography Classification by an Association Rule based Classifier. In *International Workshop on MDM with ACM SIGKDD*, pages 62–69, 2002.

[54] J.Tesic, S.Newsam, and B.S Manjunath. Mining Image Datasets using Perceptual Association Rules. In *SIAM 16^{th} International Workshop on Mining Scientific and Engineering Datasets+3^{rd} International Conference on SDM*, pages 71–77, 2003.

[55] O.R.Zaiane, J.Han, and H.Zhu. Mining Recurrent items in Multimedia with Progressive Resolution Refinement. In *International Conference on Data Engineering*, pages 461–470, 2000.

[56] O.R.Zaiane, Z.Li J.Han, and J.Hou. Mining Multimedia Data. In *CASCON'98: Meeting of Minds*, pages 83–96, Aug 1998.

[57] O.R.Zaiane. *Resource and Knowledge Discovery from the Internet Multimedia Repositories*. PhD thesis, Simon Fraser Univeristy, 1999.

[58] Robert Sedgewick. Permutation Generation Methods. *ACM Computing Surveys*, 9(2):137–164, Jun 1977.

[59] Osmar R. Zaiane and Jiawei Han. Discovering Spatial Associations in Images. In *Data Mining & Knowldge Discovery in Databases: Theory, Tools & Techniques II, SPIE, 14th International Symposium on Aerospace/Defense, Simulation and Control*, pages 138–147, 2000.

[60] J. Pan and C. Faloustos. VideoGraph: A New Tool for Video Mining and Classification. In *1st ACM + IEEE Joint Conference on Digital Libraries (JCDL)*, pages 487–499, 1994.

[61] Maria-Luiza Antonie, Osmar R Zaiane, and Alexandu Coman. Application of Data Mining Techniques for Medical Image Classification. In *2nd International Workshop on Multimedia Data Mining (MDM/KDD 2001)+7th ACM SIGKDD*, pages 94–101, 2001.

[62] George F Luger and William A Stubblefield. *Artificial Intelligence(2nd ed): Structures & Strategies for Complex Problem Solving*, chapter 13, pages 624–632. Pearson Education, 1993.

[63] J. Ross Quinlan. Induction of Decision Trees. *Machine Learning*, 1:81–106, 1986.

[64] H. Haddad and P. Mulhem. Association Rules for Symbolic Indexing of Still Images. In *2001 International Conference on Artificial Intelligence*, 2001.

[65] Yuya Matsuo, Miki Amano, and Kuniaki Uehara. Mining Video Editing rules in video streams. In *ACM International Multimedia Conference*, pages 255–258, 2002.

[66] Kimiaki Shirahama, Yuya Matsuo, and Kuniaki Uehara. Extracting Alfred Hitchcock's Know-How by Applying Data Mining Technique. In *1st International Workshop on Objects Models and Multimedia Technologies (OMMT 2003)*, pages 43–54, 2003.

[67] Jia yu Pan and Christos Faloustos. VideoCube: A Novel tool for Video Mining and Classification. In *International Conference on Asian Digital Libraries (ICADL)*, pages 194–205, 2002.

[68] Azriel Rosenfeld, David Doermann, and Daniel DeMenthon, editors. *Video Mining*. Kluwer Academic Publishers, 2003.

[69] Kimiaki Shirahama, Koichi Ideno, and Kuniaki Uehara. Video Data Mining: Mining Semantic Patterns with Temporal Constraints from Movies. In *1st IEEE International Workshop on Multimedia Information Processing and Retrieval (MIPR 2005)*, pages 589–604, 2005.

[70] Kazuhisa Iwamoto Kimiaki Shirahama and Kuniaki Uehara. Video Data Mining: Rhythms in a Movie. In *2004 IEEE International Conference on Multimedia and Expo (ICME 2004)*, pages 1463–1466, 2004.

[71] Kimiaki Shirahama, Yuya Matsuo, and Kuniaki Uehara. Mining Semantic Structures in Movies. In *15th International Conference on Applications of Declarative Programming and Knowledge Management*, pages 229–240, 2004.

[72] Yuya Matsuo, Kimiaki Shirahama, and Kuniaki Uehara. Video Data Mining: Extracting Cinematic Rules from Movie. In *4th International Workshop on Multimedia Data Mining (MDM/KDD 2003)*, pages 18–27, 2003.

[73] Wenmin Li, Jiawei Han, and Jian Pei. CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules. In *International Conference on Data Mining*, pages 369–376, 2001.

[74] Margaret H Dunham and S Sridhar. *Data Mining: Introductory & Advanced Concepts*, chapter 1–6, 8 & 9. Pearson Education, 2006.

[75] Alex A. Freitas. A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery. In *Advances in evolutionary computing: theory and applications*, pages 819–845, 2003.

[76] J. Alon, S. Sclaro, G. Kollios, and V. Pavlovic. Discovering Clusters in Motion Time-Series Data. In *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, 2003.

[77] Wei-Hao Lin, Rong Jin, and Hauptmann. Meta-classification of Multimedia Classifiers. In *International Workshop on Knowledge Discovery in Multimedia and Complex Data*, 2002.

[78] Tong Lin and Hong-Jiang Zhang. Automatic Video Scene Extraction by Shot Grouping. In *International Conference on Pattern Recognition (ICPR)*, page 4039, 2000.

[79] George Kollios, Stan Sclaroff, and Margrit Betke. Motion Mining: Discovering Spatio-Temporal Patterns in Databases of Human Motion. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2001.

[80] Neal Lesh, Mohammed J. Zaki, and Mitsunori Ogihara. Mining Features for Sequence Classification. In *5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 342–346, 1999.

[81] H. Toivonen. Sampling Large databases for Association Rules. In *22nd International Conference on VLDB*, pages 134–145, 1996.

[82] M J Zaki, S. Parthasarathy, M. Ogiharu, and W-Li. New Algorithms for Fast Discovery of Association Rules. In *3rd International Conference on Knowledge Discovery and Data Mining*, page 283, 1997.

[83] M.J Zaki. Scalable Algorithms for Association Mining. *IEEE Transactions of Knowledge and Data Engineering*, 12(3):372–390, 2000.

[84] A. Savasare, E. Omielinski, and S. Navathe. An Efficient Algorithm for mining Association Rules in Large Databases. In *21st International Conference on VLDB*, pages 432–444, 1995.