IMPLEMENTATION OF SEQUENTIAL PATTERN MINING ALGORITHM

Thesis submitted in partial fulfillment of the requirements for the award of the Degree of

MASTER OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING

OF BHARATHIAR UNIVERSITY

By

P. 652

T.SENTHIL KUMARAN
(Reg.No.0037K0010)

Under the Guidance of

Mr. K. R.BASKARAN B.E., M.S.
Asst. Professor
Dept. of CS&E, KCT

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

(Affiliated to Bharathiar University)

COIMBATORE – 641 006

2000 - 2001

CERTIFICATE

Department of Computer Science and Engineering

Certified that this is a bonafide report

of

the thesis work done by

T.SENTHIL KUMARAN
(Reg.No.0037K0010)

at

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641 006

During the year – 2000 – 2001

[illegible] 18/12

Guide
Mr.K.R.BASKARAN B.E., M.S.
Computer Science and Engineering Department
K.C.T., Coimbatore.

[illegible] 20/12

Head of the Department
Dr.S.THANGASAMY

Place : Coimbatore

Date : 20-12-2001

Submitted for Viva – Voce examination held at
Kumaraguru College of technology on 20.12.2001

[illegible]

Internal examiner

External Examiner

CERTIFICATE

This is to certify that this thesis work entitled **"IMPLEMENTATION OF SEQUENTIAL PATTERN MINING ALGORITHM "** being submitted by **T.SENTHIL KUMARAN**,(Reg.No.0037K0019 for the award of degree of **MASTER OF ENGINEERING IN COMPUTER SCIENCE** & ENGINEERING is a bonafide work carried under my guidance. The results embodied in this thesis have not been submitted to any other university or institute for the award of any degree or diploma.

[illegible]

18/12

Mr.K.R.BASKARAN B.E., M.S.

Assistant. Professor

Department of Computer Science and Engineering

Kumaraguru College of Technology

Coimbatore.

ACKNOWLEDGEMENT

It is the duty of the author to express his deep sense of gratitude to his **Parents,** and **Sisters** whose support and encouragement made to do this course **M.E., (COMPUTER SCIENCE ENGINEERING)** in this prestigious institution.

The author wishes to take this opportunity to offer a respectful note of thanks to **Dr. K.K.PADMANABAN, Ph.D.,** principal of the college, for the excellent facilities made available to accomplish this project work.

The author would like to deem it a privilege to record his sincere thanks to **Prof.S.THANGASAMY, Ph.D.,** Head of the Department of Computer Science & Engineering for his valuable suggestions and motivations during the entire period of this course.

The author would like to express his heartfelt gratitude to his guide **Mr. K.R.BASKARAN B.E., M.S.** for his valuable guidance, suggestions, and consistent encouragement, which lead to the successful completion of the project.

The author would like to express his heartfelt gratitude to his co guide **Mr. A. MUTHUKUMAR M.Sc.,M.C.A.,M.Phil.** for his valuable guidance, suggestions, and consistent encouragement, which lead to the successful completion of the project.

T.SENTHIL KUMARAN

SYNOPSIS

Data mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. Each data mining application class is supported by a set of algorithmic approaches used to extract the relevant relationships in the data; I have taken sequence-based analysis, it is one of the techniques in data mining.

Giving a large database of customer transactions, where each transaction consists of customer-id, transaction time, and the items bought in the transaction. First I find all sequential patterns with a user-specified minimum support, where the support of a sequential pattern is the percentage of data sequences that contain the pattern.

Introducing the problem of mining sequential patterns over such sequential pattern. Sequential pattern mining is the mining of frequently occurring patterns related to time or other sequences. The input data is a set of sequences, called data-sequences. Each data sequence is an ordered list of transactions (or Itemsets), where each transaction is a sets of items (literals). Typically there is a transaction-time associated with each transaction.

Implementing an algorithm to solve this problem, and empirically evaluate their performance using synthetic data. The proposed algorithm is AprioriAll. It has an excellent performance to solve this problem with respect to the number of transactions per customer and the number of items in a transaction.

CONTENTS

INTRODUCTION

1. INTRODUCTION

Generally, data mining is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

Most companies already collect and refine massive quantities of data. Data mining techniques can be implemented rapidly on existing software and hardware platforms to enhance the value of existing information resources, and can be integrated with new products and systems as they are brought on-line. When implemented on high performance client/server or parallel processing computers, data mining tools can analyze massive databases to deliver efficient results.

In my report provide an introduction to the basic technologies of data mining. Examples of profitable applications illustrate its relevance to today's business environment as well as a basic description of how data warehouse architectures can evolve to deliver the value of data mining to end users. First I explained Data, Information, Knowledge, Data where houses and Data mining.

1.1Data, Information, Knowledge

- **Data**

Data are any facts, numbers, or text that can be processed by a computer. Today, organizations are accumulating vast and growing amounts of data in different formats and different databases. This includes:

- Operational or transactional data such as, sales, cost, inventory, payroll, and accounting

- non-operational data, such as industry sales, forecast data, and macro economic data
- meta data - data about the data itself, such as logical database design or data dictionary definitions

- **Information**

The patterns, associations, or relationships among all this *data* can provide *information*. For example, analysis of retail point of sale transaction data can yield information on which products are selling and when.

- **Knowledge**

Information can be converted into *knowledge* about historical patterns and future rends. For example, summary information on retail supermarket sales can be analyzed in ight of promotional efforts to provide knowledge of consumer buying behavior. Thus, a nanufacturer or retailer could determine which items are most susceptible to promotional fforts.

.1.1 The evolution of data base technology

In the evolution from business data to business information, each new step has uilt upon the previous one. For example, dynamic data access is critical for drill-through data navigation applications, and the ability to store large databases is critical to data ining. From the user's point of view, the four steps listed in Table 1.1 were volutionary because they allowed new business questions to be answered accurately d quickly.

The core components of data mining technology have been under development r decades, in research areas such as statistics, artificial intelligence, and machine rning. Today, the maturity of these techniques, coupled with high-performance ational database engines and broad data integration efforts, make these technologies ctical for current data warehouse environments.

Evolutionary Step	**Enabling Technologies**	**Product Providers**	**Characteristics**
Data Collection (1960s)	Computers, tapes, disks	IBM, CDC	Retrospective, static data delivery
Data Access (1980s)	Relational databases (RDBMS), Structured Query Language (SQL), ODBC	Oracle, Sybase, IBM, Microsoft	Retrospective, dynamic data delivery at record level
Data Warehousing & Decision Support (1990s)	On-line analytic processing (OLAP), multidimensional databases, data warehouses	Pilot, Comshare, Arbor, Cognos, Microstrategy	Retrospective, dynamic data delivery at multiple levels
Data Mining (Emerging Today)	Advanced algorithms, multiprocessor computers, massive databases	Pilot, Lockheed, IBM, SGI, numerous startups (nascent industry)	Prospective, proactive information delivery

Table 1.1 the evolution of data base technology

1.1.2 Data Warehouses

Dramatic advances in data capture, processing power, data transmission, and storage capabilities are enabling organizations to integrate their various databases into *data warehouses*. Data warehousing is defined as a process of centralized data management and retrieval. Data warehousing, like data mining, is a relatively new term although the concept itself has been around for years. Data warehousing represents an ideal vision of maintaining a central repository of all organizational data. Centralization of data is needed to maximize user access and analysis. Dramatic technological advances are making this vision a reality for many companies. And, equally dramatic advances in data analysis software are allowing users to access this data freely. The data analysis software is what supports data mining.

1.1.3 Motivation for Data Mining

The major reason that data mining has attracted a great deal of attention in the information industry in recent years is due to the wide availability of huge amounts of data and the coming up need for turning such data in to useful information and knowledge. The information and knowledge gained can be used for applications ranging from business management, production control, and market analysis, to engineering design and science exploration.

For example scientific method turns to data only after choosing models is that in many fields, the expense and difficulty of collecting sufficient data has often been the limiting factor in research. Now, however, many researchers have literally more data than they know what to do with. Even most department stores are now capable of recording every purchase, by second, by zip code, and by any other variable they like. Some authors have even argued that computer technology is making meaning more difficult to find, by drowning it in too much data.

The idea behind data mining is to address this problem by applying computing ower to a greater portion of a scientific analysis. Just as SQL-queries and regression ave made it possible to perform collect relevant data and test models against the data of ie above method on relatively large databases, data mining hopes to carry out steps efine the problem, generate hypotheses, and use the results to generate new hypotheses itomatically on databases too large for humans to generate hypotheses effectively.

Data mining can therefore be viewed as a way to automate the scientific method. could also be seen as an alternative, since it puts data before models and experiments fore hypotheses. The differences between data mining and the traditional scientific ethod, however, may just reflect the most 'natural' way for a computer to look at merical data. Just as computer theorem provers proceed differently from human thematicians or chess computers emphasize look-ahead calculation over position, elligent data analyzers may work best when they infer everything they need from the a itself.

1.1.4 Data Mining

Data mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations. Some authors given definitions for data mining are listed below.

William J Frawley, Gregory Piatetsky-Shapiro and Christopher J Matheus

Data Mining, or Knowledge Discovery in Databases (KDD) as it is also known, is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. This encompasses a number of different technical approaches, such as clustering, data summarization, learning classification rules,finding dependency net works, analysing changes, and detecting anomalies.

Marcel Holshemier & Arno Siebes (1994)

Data mining is the search for relationships and global patterns that exist in larg databases but are `hidden' among the vast amount of data, such as a relationship betwee patient data and their medical diagnosis. These relationships represent valuabl knowledge about the database and the objects in the database and, if the database is faithful mirror, of the real world registered by the database.

1.1.5 Goal of KDD process

The unifying goal of the KDD process is to extract knowledge from data in t context of large databases. It does this by using **data mining methods** (algorithms) extract (identify) what is deemed knowledge, according to the specifications of measu and thresholds, using a database along with any required preprocessing, sub sampli and transformations of that database.

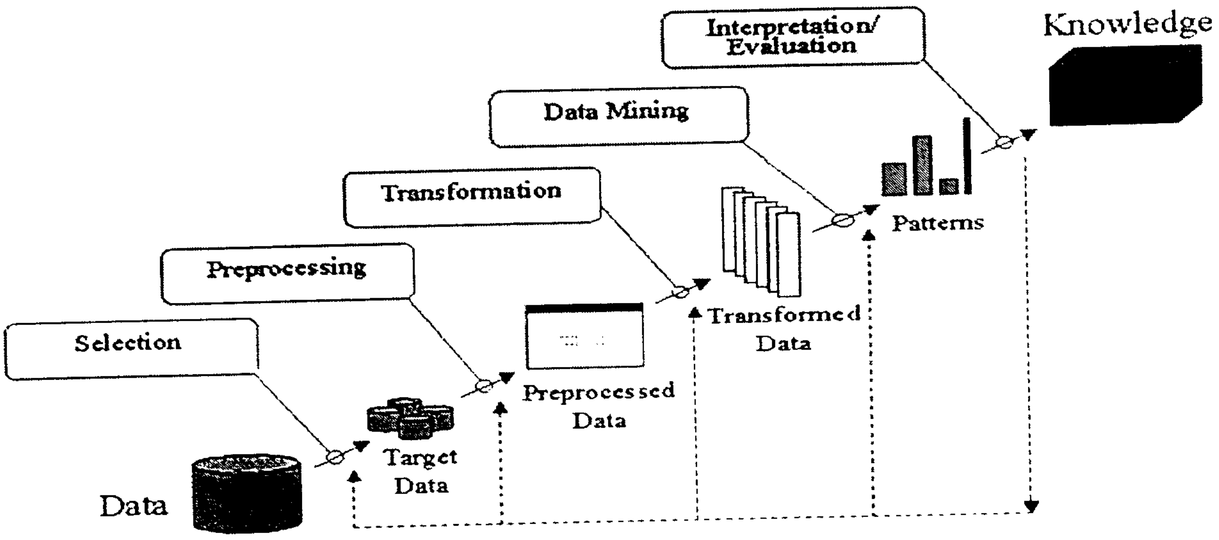

Fig 1.1 An Outline of the Steps of the KDD Process

The KDD consists of following steps

1. Developing an understanding of
    - the application domain
    - the relevant prior knowledge
    - the goals of the end-user
2. Creating a target data set: selecting a data set, or focusing on a subset of variables, or data samples, on which discovery is to be performed.
3. Data cleaning and preprocessing.
    - Removal of noise or outliers.
    - Collecting necessary information to model or account for noise.
    - Strategies for handling missing data fields.
    - Accounting for time sequence information and known changes.
4. Data reduction and projection.
5. Choosing the data-mining task.
6. Choosing the data mining algorithm(s).
7. Data mining.
8. Interpreting mined patterns.

Consolidating discovered knowledge.

1.2 The current status of the problem taken up

Data mining systems rely on data bases to supply the raw data for input and this raises problems in that databases are likely dynamic, incomplete, noisy, and large. Other problems arise as a result of the sufficiency and relevance of the information stored.

- **Limited information**

A database is often designed for purposes different from data mining and sometimes the properties or attributes that would simplify the learning task are not present nor can they be requested from the real world. Inconclusive data causes problems because if some attributes essential to knowledge about the application domain are not present in the data it may be impossible to discover significant knowledge about a given domain. For example cannot diagnose malaria from a patient database if that database does not contain the patient's red blood cell count.

- **Noise and missing values**

Databases are usually contaminated by errors so it cannot be assumed that the data they contain is entirely correct. Attributes which rely on subjective or measurement judgments can give rise to errors such that some examples may even be miss-classified. Errors in either the values of attributes or class information are known as noise. Obviously where possible it is desirable to eliminate noise from the classification information as this affects the overall accuracy of the generated rules.

Missing data can be treated by discovery systems in a number of ways such as;

1. simply ignore missing values
2. omit the corresponding records
3. gather missing values from known values
4. treat missing data as a special value to be included additionally in the attribute domain

Noisy data in the sense of being imprecise is characteristic of all data collection and typically fit a regular statistical distribution such as Gaussian while wrong values are data entry errors. Statistical methods can treat problems of noisy data, and separate different types of noise.

- **Uncertainty**

Uncertainty refers to the harshness of the error and the degree of noise in the data. Data accuracy is an important consideration in a discovery system.

- **Size, updates, and irrelevant fields**

Databases tend to be large and dynamic in that their contents are ever-changing as information is added, modified or removed. The problem with this from the data mining perspective is how to ensure that the rules are up-to-date and consistent with the most current information. Also the learning system has to be time-sensitive as some data values vary over time and the discovery system is affected by the `timeliness' of the data.

- **Cost, Time and Effort**

The data mining setup can be expensive running into hundreds of thousands of dollars. . Many man-hours of development are needed; involving complicated procedural steps and product choices. There is a need for data scrubbing or cleaning programs, and there is no single high-powered system that can handle this. Some of the data mining functions involve steep learning curves for the end-users, since higher computing power is directly related to the depth of knowledge on how the data mining system actually works. Writing SQL queries can be complex and difficult, even with a Windows-based front end tool. Extensive training and practice are still needed for most users.

- **Low-end software**

Some of the lower-end software available for data warehouse analysis tools is available for thousands of dollars, but these are piece-meal modules, not the enterprise-wide solutions necessary for data mining operations and businesses. These have limited query capabilities and its inability to perform multidimensional analyses - impossible to ask open-ended questions to find associations between data items. Many of the current data mining methods are not truly interactive and cannot incorporate prior knowledge about a problem except in simple ways.

1.3 Relevance and importance of the topic

Data mining derives its name from the similarities between searching for valuable business information in a large database — for example, finding linked products in gigabytes of store scanner data — and mining a mountain for a vein of valuable ore. Both processes require either sifting through an immense amount of material, or intelligently probing it to find exactly where the value resides. Given databases of sufficient size and quality, data mining technology can generate new business opportunities by providing these capabilities:

1.3.1 Data Mining Functionalities

Data mining methods may be classified by the function they perform or according to the class of application they can be used in. Some of the main techniques used in data mining are described in this section.

- **Classification**

By simple definition, in classification/clustering we analyze a set of data and generate a set of grouping rules which can be used to classify future data. For example, one may classify diseases and provide the symptoms which describe each class or subclass. This has much in common with traditional work in figures and machine learning. However, there are important new issues which arise because of the total size of the data. One of the important problems in data mining is the Classification-rule learning which involves finding rules that partition given data into predefined classes. In the data mining domain where millions of records and a large number of attributes are involved, the execution time of existing algorithms can become expensive, particularly in interactive applications.

- **Clustering**

Clustering approaches address segmentation problems. These approaches assign records with a large number of attributes into a relatively small set of groups or "segments." This assignment process is performed automatically by clustering algorithms that identify the distinguishing characteristics of the dataset and then partition the ndimensional space defined by the dataset attributes along natural cleaving boundaries. There is no need to identify the groupings desired or the attributes that should be used to segment the dataset.

Clustering is often one of the first steps in data mining analysis. It identifies groups of related records that can be used as a starting point for exploring further relationships. This technique supports the development of population segmentation models, such as demographic-based customer segmentation. Additional analyses using standard analytical and other data mining techniques can determine the characteristics of these segments with respect to some desired outcome. For example, the buying habits of multiple population segments might be compared to determine which segments to target for a new sales campaign.

- **Association Rule:**

An association rule is a rule which implies certain association relationships among a set of objects in a database. In this process we discover a set of association rules at multiple levels of abstraction from the relevant set(s) of data in a database. For example, one may discover a set of symptoms often occurring together with certain kind of diseases and further study the reasons behind them. Since finding interestin association rules in databases may disclose some useful patterns for decision suppor selective marketing, financial forecast, medical diagnosis, and many other applications, has attracted a lot of attention in recent data mining research. Mining association rul may require iterative scanning of large transaction or relational databases which is qui costly in processing. Therefore, efficient mining of association rules in transaction and/ relational databases has been studied substantially.

- **Sequential Analysis:**

In sequential Analysis, we seek to discover patterns that occur in sequence. This with data that appear in separate transactions (as opposed to data that appear in the s transaction in the case of association). In Traditional market-basket analysis deals w collection of items as part of a point-in-time transaction. A variant of this problem oc when there is additional information to tie together a sequence of purchases (for exar an account number, a credit card, or a frequent buyer/flyer number) in a time series. In this situation, not only may the coexistence of items within a transaction be important, but also the order in which those items appear across ordered transactions and the amount of time between transactions.

There are many algorithms proposed that try to address the above aspects of data mining. Compiling a list of all algorithms suggested/used for these problems is a difficult task. I have thus limited the focus of this report to list only some of the algorithms that have had better success than the others. These topics are discussed in detail in chapter 3, 4, 5.

- **Algorithms:**

The promise of data mining is attractive for executives and IT professionals looking to make sense out of large volumes of complex business data. The promise that programs can analyze an entire data warehouse and identify the key relationships relevant to the business is being pushed as a panacea for all data analysis woes. Today's data mining tools have typically evolved out of the pattern recognition and artificial intelligence research efforts of both small and large software companies. These tools have a heavy algorithmic component and are often rather "bare" with respect to user interfaces, execution control, and model parameterization. They typically ingest and generate UNIX flatfiles (both control and data files) and are implemented using a single-threaded computational model.

1.3.2 Potential Applications

Data mining has many and varied fields of application some of which are listed below.

- **Retail/Marketing**

  1. Identify buying patterns from customers
  2. Find associations among customer demographic characteristics

3. Predict response to mailing campaigns
4. Market basket analysis

- **Banking**
    1. Detect patterns of fraudulent credit card use
    2. Identify `loyal' customers
    3. Predict customers likely to change their credit card affiliation
    4. Determine credit card spending by customer groups
    5. Find hidden correlations between different financial indicators
    6. Identify stock trading rules from historical market data
- **Insurance and Health Care**
    1. Claims analysis - i.e. which medical procedures are claimed together
    2. Predict which customers will buy new policies
    3. Identify behavior patterns of risky customers
    4. Identify fraudulent behavior
- **Transportation**
    1. Determine the distribution schedules among outlets
    2. Analyze loading patterns
- **Medicine**
    1. Characterize patient behavior to predict office visits

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1Knowledge Discovery in Data base [From 7.3.1]

Knowledge Discovery and Data Mining (KDD) is an interdisciplinary area focusing upon Methodologies for extracting useful knowledge from data. The ongoing rapid growth of online data due to the Internet and the widespread use of databases have created an immense need for KDD methodologies. The challenge of extracting knowledge from data draws upon research in statistics, databases, pattern recognition, machine learning, data visualization, optimization, and high-performance computing, to deliver advanced business intelligence and web discovery solutions.

IBM Research has been at the forefront of this exciting new area from the very beginning. Key advances in robust and scalable data mining, methods for fast pattern detection from very large databases, text and web mining, and innovative Business intelligence applications have come from our research laboratories.

2.2Data mining Vs Query tool [From 7.1.1]

Firstly it is important to realize that query tools and data mining tools ar
complementary. A data-mining tool does not replace a query tool, but it does give th
user a lot of additional possibilities. Suppose that you have a large file containir
millions of record that describe your customer's purchase over the last ten year. There
a wealth of potentially useful knowledge in such a file, most of which can be found
firing normal queries at the database, such as who bought which product on what data'
'hat is the average turnover in a certain sales region in July?' and so on. There
however, knowledge hidden in your database that is much harder to find using SC
Example would be the answers to questions such as' what is an optional segmentatior
my clients?'(That is how do I find the most important different customer profiles?')
'What are the most important trends in customer behavior?'. Of course, these questi
could be answered using SQL .You could try to guess for yourself some defining crit
for customer profiles and query the database to see whether they work or not. In a pro
of trial and error, one could gradually develop intuitions about what the impo

distinguishing attributes are. Processing in such a way, it could take days or months to find an optimal segmentation for a large database, while a machine –learning algorithm like a neural network or a genetic algorithm could find the answer automatically in a much shorter times sometimes even in minutes or a couple of hours. Once the data mining tool has found a segmentation, you can be use your query environment again to query and analyze the profiles found

One could say that if you know exactly what you are looking for, use SQL; but if you know only vaguely what you are looking for, turn to data mining. Generally there are far more occasion when your initial approach is vague than times when you know precisely what you are looking for, it is this that has motivated the recent surge of interest in data mining.

2.3Learning and Discovery in Knowledge-Based Databases [From 7.2.1]

We present our perspective of database mining as the confluence of machine learning techniques and the performance emphasis of database technology. We describe three classes of database mining problems involving classification. associations, and sequences, and argue that these problems can be uniformly viewed as requiring discovery of rules embedded in massive data. We describe a model and some basic operations for the process of rule discovery. We show how the database mining problems we consider map to this model and how they can be solved by using the basic operations we propose. We give an example of an algorithm for classification obtained by combining the basic rule discovery

2.4Association rules [From 7.2.2]

Database mining is motivated by the decision support problem faced by most large retail organizations. Progress in bar-code technology has made it possible for retail organizations to collect and store massive amounts of sales data, referred to as the basket data. A record in such data typically consists of the transaction date and the items bought in the transaction. Success-full organizations view such databases as important pieces of the marketing infrastructure. They are interested in instituting information-driven marketing processes, managed by database technology, that enable marketers to develop and implement customized marketing programs and strategies.

The problem of mining association rules over basket data was introduced in an Example of such a rule might be that 98% of customers that purchase tires and auto accessories also get automotive services done. Finding all such rules is valuable for cross-marketing and attached mailing applications. Other applications include catalog design, add-on sales, store layout, and customer segmentation based on buying patterns. The databases involved in these applications are very large. It is imperative, therefore, to have fast algorithms for this task.

In this paper is very useful to know about the association rule because the Apriori property employed in association rule mining can be applied to mining sequential patterns algorithm because if a sequential pattern of length k is infrequent, its superset(of length k+1) cannot be frequent. Therefore, most of the methods for mining sequential pattern adopt variations of Apriori – like algorithms although they may consider different parameter strings and constrains

2.5Sequential Patterns [From 7.2.3]

The input data is a set of sequences, called data-sequences. Each data sequence i a ordered list of transactions (or itemsets), where each transaction is a sets of item (literals). Typically there is a transaction-time associated with each transaction. sequential pattern also consists of a list of sets of items. The problem is to find a sequential patterns with a user-specified minimum support, where the support of sequential pattern is the percentage of data sequences that contain the pattern.

Elements of a sequential pattern need not be simple items. ``Fitted Sheet and f sheet and pillow cases", followed by ``comforter", followed by ``drapes and ruffles" is example of a sequential pattern in which the elements are sets of items. This problem v initially motivated by applications in the retailing industry, including attached maili add-on sales, and customer satisfaction. But the results apply to many scientific in the medical domain, a data-sequence may corresp to the symptoms or diseases of a patient, with a transaction corresponding to the symptoms exhibited or diseases diagnosed during a visit to the doctor. The patterns discovered using this data could be used in disease research to help identify symptoms/diseases that precede certain diseases.

I have taken the problem from this paper. In this paper is very useful to know about the concept of sequential pattern and how it is mined. The author proposed an algorithm for sequential pattern mining algorithm named as Apriori all. I implemented the algorithm.

2.6Data Mining Examples [From 7.3.3]

- **.Northern Bank**

A subsidiary of the National Australia Group, the Northern Bank has a major new application based upon Holos from Holistic Systems now being used in each of the 107 branches in the Province. The new system is designed to deliver financial and sales nformation such as volumes, margins, revenues, overheads and profits as well as uantities of product held, sold, closed etc.

'he application consists of two loosely coupled systems;

- a system to integrate the multiple data sources into a consolidated database,
- Another system to deliver that information to the users in a meaningful way.

he Northern is addressing the need to convert data into information as their products eed to be measured outlet by outlet, and over a period of time.

ie new system delivers management information in electronic form to the branch twork. The information is now more accessible, paperless and timely. For the first ie, all the various income streams are attributed to the branches which generate the siness.

alcolm Longridge, MIS development team leader, Northern Bank

- **Delphic Universities**

The Delphic universities are a group of 24 universities within the MAC initiative who have adopted Holos for their management information system, MIS, needs. Holos provides complex modelling for IT literate users in the planning departments while also giving the senior management a user-friendly EIS.

Real value is added to data by multidimensional manipulation (being able to easily compare many different views of the available information in one report) and by modelling. In both these areas spreadsheets and query-based tools are not able to compete with fully-fledged management information systems such as Holos. These two features turn raw data into useable information.

Michael O'Hara, chairman of the MIS Application Group at Delphic

- **Harvard – Holden**

Harvard university has developed a centrally operated fund-raising system that allows university institutions to share fund-raising information for the first time.

The new Sybase system, called HOLDEN (Harvard Online Development Network), is expected to maximize the funds generated by the Harvard Development Office from the current donor pool by more precisely targeting existing resources and eliminating wasted efforts and redundancies across the university. Through this streamlining, HOLDEN will allow Harvard to pursue one of the most ambitious fund-raising goals ever set by an American institution to raise $2 billion in five years.

Harvard University has enjoyed the nation's premier university endowment since 1636. Sybase technology has allowed us to develop an information system that will preserve this legacy into the twenty-first century

Jim Conway, director of development computing services, Harvard University

- **J.P. Morgan**

This leading financial company was one of the first to employ data mining/forecasting applications using Information Harvester software on the Convex Examplar and C series.

The promise of data mining tools like Information Harvester is that they are able to quickly wade through massive amounts of data to identify relationships or trending information that would not have been available without the tool

Charles Bonomo, vice president of advanced technology for J.P. Morgan

The flexibility of the Information Harvesting induction algorithm enables it to adapt to any system. The data can be in the form of numbers, dates, codes, categories, text or any combination thereof. Information Harvester is designed to handle faulty, missing and noisy data. Large variations in the values of an individual field do not hamper the analysis. Information Harvester has unique abilities to recognize and ignore irrelevant data fields when searching for patterns. In full-scale parallel-processing versions, Information Harvester can handle millions of rows and thousands of variables.

PROPOSED LINE OF ATTACK

3. PROPOSED LINE OF ATTACK

Database mining is motivated by the decision support problem faced by most large organizations. Progress in bar-code technology has made it possible for retail organizations to collect and store massive amounts of sales data, referred to as the basket data. A record in such data typically consists of the transaction date and the items bought in the transaction. Very often, data records also contain customer-id, particularly when the purchase has been made using a credit card or a frequent-buyer card. Catalog companies also collect such data using the orders they receive. We introduce the problem of mining sequential patterns over this data. Sequential pattern mining functions are quite powerful and can be used to detect the set of customers associated with some frequent buying patterns. Elements of a sequential pattern need not be simple items. ``Fitted Sheet and flat sheet and pillow cases", followed by ``cover", followed by ``long curtains and towel" is an example of a sequential pattern in which the elements are sets of items. This problem was initially motivated by applications in the retailing industry, including attached mailing, and customer satisfaction. But the results apply to many scientific and business domains.

Mining sequential pattern

Task: Based on the problems to find the maximal sequences among all sequences that ave a certain specified minimum support. Each such maximal sequence represents a equential pattern.

DETAILS OF PROPOSED METHODOLOGY

4. DETAILS OF THE PROPOSED METHODOLOGY

Sequential pattern functions analyze a collection of records over a period of time for example to identify trends. Where the identity of a customer who made a purchase is known an analysis can be made of the collection of related records of the same structure (i.e. consisting of a number of items drawn from a given collection of items). An example of such a pattern is that customers typically rent "Star Wars", then "Empire Strikes Back", and then "return of the Jedi". Note that these rentals need not be consecutive. Customers who rent some other videos in between also support this sequential pattern. The records are related by the identity of the customer who did the repeated purchases. Such a situation is typical of a direct mail application where for example a catalogue merchant has the information, for each customer, of the sets of products that the customer buys in every purchase order. A sequential pattern function will analyze such collections of related records and will detect frequently occurring patterns of products bought over ime. A sequential pattern operator could also be used to discover for example the set of urchases that frequently precede the purchase of a microwave oven.

Sequential pattern mining functions are quite powerful and can be used to detect e set of customers associated with some frequent buying patterns. Use of these nctions on for example a set of insurance claims can lead to the identification of equently occurring sequences of medical procedures applied to patients, which can help entify good medical practices as well as to potentially detect some medical insurance ud.

Problem declaration

I am given a database D of customer transactions. Each transaction consists of the owing fields: customer-id, transaction-time, and the items purchased in the saction. No customer has more than one transaction with the same transaction-time. I ot consider quantities of items bought in a transaction: each item is a binary variable esenting whether an item was bought or not.

An Itemset is a non-empty set of items. A sequence is an ordered list of Itemsets. Without loss of generality, we assume that the set of items is mapped to a set of contiguous integers.

We denote an Itemset i by (i1,i2 : : : im), where ij is an item. I denote a sequence s by (s1,s2: : : sn) where sj is an Itemset. Itemset: a non-empty set of items, $\langle i1\ i2\ i3 \ldots \rangle$.

Sequence: an ordered list of Itemsets, $\langle s1\ s2\ s3 \ldots \rangle$.

A sequence $\langle a1\ a2 \ldots an \rangle$ is contained in $\langle b1\ b2 \ldots bn \rangle$ if there exist $i1 < i2 < \ldots < in$ such that $a1 \subseteq bi1$, $a2 \subseteq bi2$, … $an \subseteq bin$ E.g., $\langle (3)(4\ 5)(8) \rangle \subseteq \langle (7)(3\ 8)(9)(4\ 5\ 6)(8) \rangle$, since $(3) \subseteq (3\ 8)$, $(4\ 5) \subseteq (4\ 5\ 6)$ and $(8) \subseteq (8)$.However, note that sequence $\langle (3)(5) \rangle \not\subset \langle (3\ 5) \rangle$ (and vice versa).

Customer sequence a sequence of transactions ("shopping baskets") of a customer, ordered by transaction times Ti:$\langle Itemset(T1)\ Itemset(T2) \ldots Itemset(Tn) \rangle$.

A customer supports a sequence s if s is contained in the customer sequence for this customer. The support for a sequence is defined as the fraction of total customers who support this sequence.

A customer supports a sequence s if s is contained in the customer-sequence for this customer. The support for a sequence is defined as the fraction of total customers who support this sequence. Below given database D is a sample database I am going to explain my project based on the example. We call a sequence satisfying the minimum support constraint a large sequence. Example. Consider the database shown in Fig. 4.1

Given a database D of customer transactions, the problem of mining sequential patterns is to find the maximal sequences among all sequences that have a certain user-specified minimum support. Each such maximal sequence represents a sequential pattern. We call a sequence satisfying the minimum support constraint a large sequence.

ple Database

Date	TID	Items
01/08/2001	1	30
02/08/2001	2	10,20
03/08/2001	3	30,50,70
04/08/2001	4	30
05/08/2001	5	90
06/08/2001	1	90
07/08/2001	2	30
09/08/2001	4	40,70
10/08/2001	2	40,60,70
11/08/2001	4	90

Fig 4.1 Sample Database

4.2Problem based associated work

In the problem of discovering what items are bought together in a transaction over basket data was introduced. While related, the problem of finding what items are bought together is concerned with finding intra-transaction patterns, whereas the problem of finding sequential patterns is concerned with inter-transaction patterns. A pattern in the first problem consists of an unordered set of items whereas a pattern in the latter case is an ordered list of sets of items.

Discovering patterns in sequences of events has been an area of active research in AI. However, the focus in this body of work is on discovering the rule underlying the generation of a given sequence in order to be able to predict a reasonable sequence continuation (e.g. the rule to predict what number will come next, given a sequence of numbers). We on the hand are interested in finding all common patterns embedded in a database of sequences of sets of events (items).

My problem is related to the problem of finding text subsequences that match a given UNIX grep utility). There also has been work on finding text subsequences that approximately match a given string. These techniques are oriented toward finding matches for one pattern. In my problem, the difficulty is in figuring out what patterns to try and then efficiently finding out which ones are contained in a customer sequence.

Techniques based on multiple alignments have been proposed to find entire text sequences that are similar. There also has been work to find locally similar subsequences. However, as pointed out in these techniques apply when the discovered patterns consist of consecutive characters or multiple lists of consecutive characters separated by a fixed length of noise characters.

Closest to my problem is the problem formulation in the context of discovering similarities in a database of genetic sequences. The patterns they wish to discover are subsequences made up of consecutive characters separated by a variable number of noise characters. A sequence in my problem consists of list of sets of characters (items), rather than being simply a list of characters.

Thus, an element of the sequential pattern we discover can be a set of characters (items), rather than being simply a character. My solution approach is entirely different. The solution is not guaranteed to be complete, whereas we guarantee that we have discovered all sequential patterns of interest that are present in a specified minimum number of sequences. My solution is targeted at millions of customer sequences.

4.3Organization of the Project

We solve the problem of finding all sequential patterns in five phases:

i) SORT PHASE.

ii) LITEMSET PHASE.

iii) TRANSFORMATION PHASE.

iv) SEQUENCE PHASE.

v) MAXIMAL PHASE.

gives this problem decomposition

inding Sequential Patterns

The length of a sequence is the number of Itemsets in the sequence. A sequence of th k is called a k-sequence. The sequence formed by the concatenation of two uences x and y is denoted as x.y.

The support for an Itemset i is defined as the fraction of customers who bought e items in i in a single transaction. Thus the Itemset i and the 1-sequence (i) have the me support. An Itemset with minimum support is called a large Itemset or LItemset. ote that each Itemset in a large sequence must have minimum support. Hence, any large equence must be a list of lItemsets.

4.4.1Mining Sequential Algorithm

I split the problem of mining sequential patterns into the following phases:

- **Sort Phase**

In the sort phase we have two major things in the database, 1. Customer-id and 2. Transaction- time. In the sample database customer -id as the major key and transaction-time as the minor key. Based on the customer- id we are going to sort the database and

Apply to next phase. For example Fig 4.1 is a sample database; this step implicitly converts the original transaction database into a database of customer sequences. Fig. 4.2 shows the sorted database for the example database in Fig. 4.1

TID	DATE	ITEMS
1	01/08/2001	30
1	06/08/2001	90
2	03/08/2001	10,20
2	04/08/2001	30
2	05/08/2001	40,60,70
3	06/08/2001	30,50,70
4	07/08/2001	30
4	09/08/2001	40,70
4	10/08/2001	90
5	11/08/2001	90

Fig 4.2 Sorted Database

Pseudo code for the Sort Phase

```
cust_file = new FileReader("cust.dat");
cust_data = new BufferedReader(cust_file);
```

```
while((s = cust_data.readLine()) != null)

if(!(s.equals("")))
v.addElement(s);

cust_data.close();
cust_file.close();

data = v;
Collections.sort(data);

String  m;
for(int i=0; i<data.size(); i++)
{
m = (String)data.get(i);
m += "\r\n";
System.out.print(m);
sort_file.write(m.getBytes());
}
sort_file.close();
System.out.println("Sorted.\n\n");
v.clear();
}
```

- **Merge Phase**

In the phase is very important because in this phase the data base size to be minimized. For example in Fig 4.2 shows the sorted data base in the database customer-id is same but the purchased items are different. So we remove the repeated customer –

id and merge the customer's purchased items. Fig. 4.3 shows the merged database for the example database in Fig. 4.2.

TID	ITEMS
1	<(30)(90)>
2	<(10,20) (30) (40,60,70)>
3	<(30,50,70)>
4	<(30)(40,60)(90))
5	<(90)>

Fig 4.3 Merged Database.

Pseudo code for the Merge Phase

```
cust_file = new FileReader("sort.dat");
cust_data = new BufferedReader(cust_file);
while((s = cust_data.readLine()) != null)
   v.addElement(s);
cust_data.close();
cust_file.close();
```

```
data = v;
 String merge = new String();
  String m = new String();
  for(int i=0; i<data.size(); i++)
     m = (String)data.get(i);


      m += ' ';
     sb.setString(m);
      ok = sb.getStrings();
    if (i==0)
     merge_file.write(m.getBytes());
       merge = ok[0].trim();
       else
       if (merge.equals(ok[0].trim()))
          merge="";
        for(int k=1;k<sb.getCount();k++)
        merge += ok[k];
         merge += " ";
         merge_file.write(merge.getBytes());
            merge=ok[0].trim();


          else
          merge_file.write(m.getBytes());
           merge = ok[0].trim();
           merge_file.close();
```

• LItemset Phase

In the phase we are going to find largest purchased items in the database. To fir

item user specified the minimum support based on this value we check the items.

than and equal the minimum support to insert the item in the large database. For exam we set the minimum support value is 25%. We apply this value in Fig 4.4

An above example of a sequence that does not have minimum support is sequence {(10 20) (30)}, which is only supported by customer 2. The sequences { (},{ (40) }, { (70) },{ (90) }, { (30) (40) }, { (30) (70) } and { (40 70) }, though hav minimum support, are not in the answer because they are not maximal.

Then we given the sequence number of these large items because we can process in the existing purchased items number, it is very difficult. We apply the sam database in Fig 4.4.

Min. support 25% (i.e. 25*5/100=1.25)

TID	ITEMS	
1	<(30)(90)>	>= **Minimum suppo purchased Items a** <(30)(90)> (1&4) and <(30)(40 70)> (2&4) are maximal
2	<(10,20) (30) (40,60,70)>	
3	<(30,50,70)>	
4	<(30)(40,60)(90))	
5	<(90)>	

Fig 4.4 Display above Min.Supported Database

LARGE ITEMS	MAPPED TO
(30)	1
(40)	2
(70)	3
(40 70)	4
(90)	5

Fig 4.5 Display Large items Mapped Sequence number Database

Pseudo code for the LItem Phase

```
for(int h=0;h<tmp.size();h++)
    {
 cx=1;
for(int f=0;f<v.size();f++)
    {
if(tmp.elementAt(h).equals(v.elementAt(f)))
 cx++;
    }
 cx--;
 if (cx>=min)
 {
 m=Integer.toString(cx);
m+=(String)tmp.elementAt(h);
vx.add(m);
}

}

Comparator r = Collections.reverseOrder();
Collections.sort(vx,r);
for( i=0; i<vx.size(); i++)
System.out.println("\t"+vx.get(i));
String temp=new String();
for(i=0;i<vx.size();i++)
{
m = Integer.toString(i+1);
temp = (String)vx.get(i);
m+=temp.substring(2);
large_file.write(m.getBytes());
large_file.close();
```

- **Transformation Phase**

There is a need to repeatedly check which large Itemsets are contained in customer sequences. To make this fast, each customer sequence is transformed to a list of large Itemsets. Then the large Itemsets are mapped to integers

We need to repeatedly determine which of a given set of large sequences are contained in a customer sequence. To make this test fast, we transform each customer sequence into an alternative representation. In a transformed customer sequence, each transaction is replaced by the set of all LItemsets contained in that transaction. If a transaction does not contain any LItemset, it is not retained in the transformed sequence. If a customer sequence does not contain any LItemset, this sequence is dropped from the transformed database. However, it still contributes to the count of total num customers.

A customer sequence is now represented by a list of sets of LItemsets. Each LItemsets is represented by (l1; l2; . . .; ln), where li is a LItemset. This transf database is called DT. Depending on the disk availability, we can physically crea transformed database, or this transformation can be done on the fly, as we rea customer sequence during a pass. For Example based on the Fig 4.3 &Fig 4.5 val transform original sequences to transform sequences

CID	Original Sequence	Transformed Sequence	After Mapp
1	<(30)(90)>	{(30),(90)}	{1,5}
2	<(10,20) (30) (40,60,70)>	{(30),(40),(70),(40 70)}	{1,2,3,4
3	<(30,50,70)>	{(30),(70)}	{1,3}
4	<(30)(40,60)(90))	{(30 40),(70),(90),(40 70),(90)}	{1 ,2,3,4,5
5	<(90)>	(90)	{5}

Fig 4.6. Transformed original sequence to transformed sequence

For above example, during the trans- formation of the customer sequence with Id 2, the transaction (10 20) is dropped because it does not contain any LItemset and the transaction (40 60 70) is replaced by the set of LItemsets ((40), (70), (40 70)).

Pseudo code for the Transformation Phase

```
Vector v_sub = new Vector();
Vector v_mer = new Vector();
Vector v_lar = new Vector();
cust_file = new FileReader("subsets.dat");
cust_data = new BufferedReader(cust_file);
 while((s = cust_data.readLine()) != null)
   {
  if(!(s.equals("")))
     v_sub.addElement(s);
    }
   cust_data.close();
   cust_file.close();
  cust_file = new FileReader("merge.dat");
  cust_data = new BufferedReader(cust_file);
    while((s = cust_data.readLine()) != null)
      if(!(s.equals("")))
      v_mer.addElement(s);
      cust_data.close();
      cust_file.close();


      }
      catch(FileNotFoundException e)
    cust_file = new FileReader("large.dat");

     cust_data = new BufferedReader(cust_file);
     while((s = cust_data.readLine()) != null)
         if(!(s.equals("")))
```

```
cust_data.close();
cust_file.close();
 String m1=new String();
 String m2=new String();
 String m3=new String();
 String m4=new String();
 String m5=new String();
 String  m;
 int l=0;
 Vector tr = new Vector();
 Vector trf = new Vector();
 int mi,ni;
 mi=0;ni=0;

 trans_file = new FileOutputStream("trans.dat");
 for(int j=1;j<v_sub.size();j++)
 m1 =(String)v_sub.get(j);
 m5=m1.substring(mi+1,ni);
 if(s.compareTo(m5)==0)
 m = (String)v_mer.get(l)+" # ";
  l++;
 for(int i=0; i<tr.size(); i++)
  m += (String)tr.get(i);
  for(int i=0; i<trf.size(); i++)
   m += (String)trf.get(i);
   trans_file.write(m.getBytes());
    else
    mi=0;ni=0;
    for( int k=0;k<v_lar.size();k++)
    m2 =(String)v_lar.get(k);
    m3=m2.substring(mi+1,ni);
    if(m3.compareTo(m5)==0)
if(!tr.contains(m5))
tr.addElement(m5.trim());

m4=m2.substring(0,mi);
trf.addElement(m4.trim());
if (v_mer.size()>0)
m = (String)v_mer.get(1)+" # ";
for(int i=0; i<tr.size(); i++)
m+= (String)tr.get(i);
for(int i=0; i<trf.size(); i++)
m+= (String)trf.get(i);
trans_file.write(m.getBytes());
```

Sequence Phase

The large Itemsets are used to find the desired sequences.

Implementing algorithms for the sequence phase. The general structure of th algorithms is that they make multiple passes over the data. In each pass, we start wit seed set of large sequences. We use the seed set for generating new potentially la sequences, called candidate sequences. We find the support for these candidate sequen during the pass over the data. At the end of the pass, we determine which of the candic sequences are actually large. These large candidates become the seed for the next pass the first pass, all 1-sequences with minimum support, obtained in the LItemset ph form the seed set.

Implementing familiar of algorithms, which we call count-all. The coun algorithms count all the large sequences, including non-maximal sequences. The r maximal sequences must then be pruned out (in the maximal phase). We present count-all algorithm, called AprioriAll, based on the Apriori algorithm for finding l Itemsets .

- **Algorithm AprioriAll**
  - Based on the normal Apriori algorithm
  - Counts all the large sequences
  - Prunes non-maximal in the "Maximal phase"

Algorithm Apriori

An association rule-mining algorithm, Apriori has been developed for rule mining in large transaction databases by IBM's Quest project team. A itemset is a non-empty set of items. They have decomposed the problem of mining association rules into two parts

Find all combinations of items that have transaction support above minimum support. Call those combinations frequent itemsets.

Use the frequent itemsets to generate the desired rules. The general idea is that if, say, ABCD and AB are frequent itemsets, then we can determine if the rule AB CD holds by computing the ratio r = support(ABCD)/support(AB). The rule holds only if r >= minimum confidence. Note that the rule will have minimum support because ABCD is frequent.

Apriori All algorithm was proposed. Apriori algorithm generates the candidate Itemsets to be counted in the pass by using only the Itemsets found large in the previous pass-without the transaction in the database.

The key idea of Apriori All algorithm lies in the "*downward-closed*" property of support which means if an Itemset has minimum support, then all its subsets also have minimum support is called frequent Itemset having k item can be generated by joining frequent Itemsets having k-1 items, and deleting those that contain any subset that is not frequent.

Apriori All is an influential algorithm for mining frequent Itemset for Boolean association rules. The name of the algorithm is based on the fact that the algorithm uses *rior knowledge* of frequent Itemset properties, as we shall see below. Apriori All employs an interactive approach known as a *level–wise* search, here k-Itemset are used to explore (k+1)-Itemset. Starting by finding all frequent 1-Itemsets (items with 1 item) denoted as L_1 , we then consider 2-Itemsets say L_2, and so forth. The finding of each L_k requires one full scan of the database. So during each iteration only candidates found to be frequent in the previous iteration are used to generate a new candidate set during the next iteration. The algorithm terminates when there are no frequent k-Itemsets.

To improve the efficiency of the level-wise generation of frequent in Itemsets, an important property called the ***Apriori property***, presented below, is used to reduce the search space. In order to use the *Apriori property*, all nonempty subsets of a frequent Itemset must also be frequent. This property is based on the following observation. By definition, if an Itemset I does not satisfy the minimum support threshold, min_*sup*, then I is not frequent, that is, $P(I)<min\_sup$. If an item A is added to the Itemset I, then resulting Itemset I (i.e., $I \cup A$) cannot occur more frequently than I. Therefore, $I \cup A$ is not frequent either, that is, $P(I \cup A)<min\_sup$.

This property belongs to a special category of properties called ***anti-monotone*** in the sense that if a set cannot pass a test, all of its subsets will fail the same test as well. It is called anti-monotone because the property is monotonic in the context of failing a test.

- **Algorithm Apriori All**

Apriori All. Find frequent Itemsets using an iteractive level-wise approach base on candidate generation

Notation is given below

K-Itemset	An Itemset having k Itemsets
L_k	Set of frequent k-Itemset(those with minimum support)
C_k	Set of candidate k-Itemset(potentially frequent Itemset

Pseudo code for the Apriori All algorithm

```
L1 = large 1-sequences; // Result of LItemset phase
for ( k = 2; Lk-1 0; k++) do
begin
        Ck = New Candidates generated from Lk-1 (see below)
        foreach customer-sequence c in the database do
         Increment the count of all candidates in Ck that are contained in c.
        Lk = Candidates in Ck with minimum support.
 end

                Answer = Maximal Sequences in k Lk ;
```

The algorithm is given above. In each pass, we use the large sequences from the previous pass to generate the candidate sequences and then measure their support by making a pass over the database. At the end of the pass, the support of the candidates is used to determine the large sequences. In the first pass, the output of the lItemset phase is used to initialize the set of large 1-sequences. The candidates are stored in hash-tree to quickly find all candidates contained in a customer sequence.

Apriori Candidate Generation

The Apriori-generate function takes as argument L_{k-1}, the set of all large (k-1)-sequences. It works as follows. First join L_{k-1} with L_{k-1}

Insert into C_k

Select p.lItemset$_1$, ..., p.lItemset$_{k-1}$, q.lItemset$_{k-1}$

From L_{k-1} p, L_{k-1} q

Where p.LItemset$_1$ = q.lItemset$_1$. . .

p.LItemsetk-2 = q.lItemset$_{k-2}$;

Next delete all sequences c C_k such that some (k-1)-subsequence of c is not in L_{k-1}

Based on this Apriori All algorithm we find the maximum sequence of the pattern. I will give a maximal sequence of this algorithm based of the above example.

For example, consider the set of 3-sequences L3 shown in Fig. A. If this is given as input to the Apriori-generate function, we will get the sequences shown in Fig. B after the join. After pruning out sequences whose subsequences are not in L3, the sequences shown in Fig. C will be left. For example, (1 2 4 3) is pruned out because the subsequence (2 4 3) is not in L3.

Figure A: Candidate 3-Sequences

Sequence	Support
{1 2 3}	2
{1 2 4}	2
{1 3 4}	3
{1 3 5}	2
{2 3 4}	2

Figure B: Candidate 4-Sequences (after join)

{1 2 3 4}
{1 2 4 3}
{1 3 4 5}
{1 3 5 4}

Figure C: Candidate 4-Sequences (after pruning)

{1 2 3 4}

We need to show that $C_k \supseteq L_k$. Clearly, any subsequence of a large sequence must also have minimum support. Hence, if we extended each sequence in L_{k-1} with all possible large Itemsets and then deleted all those whose (k-1)-subsequences were not in L_{k-1}, we would be left with a superset of the sequences in L_k. The join is equivalent to extending L_{k-1} with each large Itemet and then deleting those sequences for which the (k-1)-sequence obtained by deleting the $(k-1)^{th}$ Itemset is not in L_{k-1}. Thus, after the join step, $C_k \supseteq L_k$. By similar reasoning, the prune step, where we delete from C_k all sequences whose (k-1)-subsequences are not in L_{k-1} .

- **Maximal Phase**

Find the maximal sequences among the large sequences in practice, starting from the largest sequences, delete all their subsequences. For above example find the maximal sequences among the set of large sequences. Sequence phase to reduce the time wasted in counting non-maximal sequences. Having found the set of all large sequences S in the sequence phase, the following algorithm can be used for finding maximal sequences. Let the length of the longest sequence be n.

RESULTS OBTAINED

5. RESULTS OBTAINED

I implemented in this algorithm using java1.3

This following section contains 7 options; the user can choose any one option and get the results.

```
Main menu

1. Add Customer Data
2. Sort Customer Data
3. Merge Customer Data
4. Large Items
5. Transformation
6. Sequence
7. Exit

Enter any choice: 1
______________________

Do You Like Process Existing Data Or Add Your Own Data

If Press Y means Existing Or If Press N Means Own Data
N
Enter Customer Number Except 0 :
1
Enter Purchased Items :
asd
Enter Customer Number Except 0 :
2
Enter Purchased Items :
cdfr
Enter Customer Number Except 0 :
3
Enter Purchased Items :
aqwe
Enter Customer Number Except 0 :
1
Enter Purchased Items :
cxsae
Enter Customer Number Except 0 :
2
Enter Purchased Items :
fdew
Enter Customer Number Except 0 :
```

```
s-one item)
1 asd
2 cdfr
3 aqwe
1 cxsae
2 fdew
Data Loaded.


 1. Add Customer Data
 2. Sort Customer Data
 3. Merge Customer Data
 4. Large Items
 5. Transformation
 6. Sequence
 7. Exit

 Enter any choice : 2
 ______________________

 1 asd
 1 cxsae
 2 cdfr
 2 fdew
 3 aqwe

 Data Sorted.


 1. Add Customer Data
 2. Sort Customer Data
 3. Merge Customer Data
 4. Large Items
 5. Transformation
 6. Sequence
 7. Exit

 Enter any choice :3
 ______________________

 1 asd cxsae
 2 cdfr fdew
 3 aqwe
 Data Merged.


 1. Add Customer Data
 2. Sort Customer Data
```

```
3. Merge Customer Data
4. Large Items
5. Transformation
6. Sequence
7. Exit

Enter any choice : 4
___________________________
```

In this option is used to find all the large items in the database, to count the items and its subsets and check if the items are >= Minimum.

```
----------------------
  Minval =  1.2
----------------------
Count |Items
----------------------
      3|e
      3|d
      3|a
      2|w
      2|s
      2|f
      2|c
      2|ae

  1. Add Customer Data
  2. Sort Customer Data
  3. Merge Customer Data
  4. Large Items
  5. Transformation
  6. Sequence
  7. Exit

  Enter any choice : 5
  ___________________________


  1 asd cxsae   # d s a ae e c  # 2 5 3 8 1 7

  2 cdfr fdew   # f d c w e  # 6 2 7 4 1

  3 aqwe  # ae e w a  # 8 1 4 3



   1. Add Customer Data
   2. Sort Customer Data
```

```
4. Large Items
5. Transformation
6. Sequence
7. Exit

Enter any choice : 6
________________________


Enter Minimum Support Values

30

Algorithm Apriori all starting now.....

Input configuration: 8 items, 3 transactions ,minsup = 30%


Frequent 1-itemsets:

[1, 2, 3, 4, 5, 6, 7, 8]


Frequent 2-itemsets:

[1 2, 1 3, 1 4, 1 5, 1 6, 1 7, 1 8, 2 3, 2 4, 2 5, 2 6, 2 7, 2 8, 3 4, 3 5, 3 6,
 3 7, 3 8, 4 5, 4 6, 4 7, 4 8, 5 6, 5 7, 5 8, 6 7, 6 8, 7 8]



Frequent 3-itemsets:

[1 2 3, 1 2 4, 1 2 5, 1 2 6, 1 2 7, 1 2 8, 1 3 4, 1 3 5, 1 3 6, 1 3 7, 1 3 8, 1
4 5, 1 4 6, 1 4 7, 1 4 8, 1 5 6, 1 5 7, 1 5 8, 1 6 7, 1 6 8, 1 7 8, 2 3 4, 2 3 5
, 2 3 6, 2 3 7, 2 3 8, 2 4 5, 2 4 6, 2 4 7, 2 4 8, 2 5 6, 2 5 7, 2 5 8, 2 6 7, 2
 6 8, 2 7 8, 3 4 5, 3 4 6, 3 4 7, 3 4 8, 3 5 6, 3 5 7, 3 5 8, 3 6 7, 3 6 8, 3 7
8, 4 5 6, 4 5 7, 4 5 8, 4 6 7, 4 6 8, 4 7 8, 5 6 7, 5 6 8, 5 7 8, 6 7 8]

Frequent 4-itemsets:

[1 2 3 4, 1 2 3 5, 1 2 3 6, 1 2 3 7, 1 2 3 8, 1 2 4 5, 1 2 4 6, 1 2 4 7, 1 2 4 8
, 1 2 5 6, 1 2 5 7, 1 2 5 8, 1 2 6 7, 1 2 6 8, 1 2 7 8, 1 3 4 5, 1 3 4 6, 1 3 4
7, 1 3 4 8, 1 3 5 6, 1 3 5 7, 1 3 5 8, 1 3 6 7, 1 3 6 8, 1 3 7 8, 1 4 5 6, 1 4 5
 7, 1 4 5 8, 1 4 6 7, 1 4 6 8, 1 4 7 8, 1 5 6 7, 1 5 6 8, 1 5 7 8, 1 6 7 8, 2 3
4 5, 2 3 4 6, 2 3 4 7, 2 3 4 8, 2 3 5 6, 2 3 5 7, 2 3 5 8, 2 3 6 7, 2 3 6 8, 2 3
7 8, 2 4 5 6, 2 4 5 7, 2 4 5 8, 2 4 6 7, 2 4 6 8, 2 4 7 8, 2 5 6 7, 2 5 6 8, 2
```

```
Frequent 5-itemsets:

[1 2 3 4 5, 1 2 3 4 6, 1 2 3 4 7, 1 2 3 4 8, 1 2 3 5 6, 1 2 3 5 7, 1 2 3 5 8, 1
2 3 6 7, 1 2 3 6 8, 1 2 3 7 8, 1 2 4 5 6, 1 2 4 5 7, 1 2 4 5 8, 1 2 4 6 7, 1 2 4
6 8, 1 2 4 7 8, 1 2 5 6 7, 1 2 5 6 8, 1 2 5 7 8, 1 2 6 7 8, 1 3 4 5 6, 1 3 4 5
7, 1 3 4 5 8, 1 3 4 6 7, 1 3 4 6 8, 1 3 4 7 8, 1 3 5 6 7, 1 3 5 6 8, 1 3 5 7 8,
1 3 6 7 8, 1 4 5 6 7, 1 4 5 6 8, 1 4 5 7 8, 1 4 6 7 8, 1 5 6 7 8, 2 3 4 5 6, 2 3
4 5 7, 2 3 4 5 8, 2 3 4 6 7, 2 3 4 6 8, 2 3 4 7 8, 2 3 5 6 7, 2 3 5 6 8, 2 3 5
7 8, 2 3 6 7 8, 2 4 5 6 7, 2 4 5 6 8, 2 4 5 7 8, 2 4 6 7 8, 2 5 6 7 8, 3 4 5 6 7
, 3 4 5 6 8, 3 4 5 7 8, 3 4 6 7 8, 3 5 6 7 8, 4 5 6 7 8]

Frequent 6-itemsets:

[1 2 3 4 5 6, 1 2 3 4 5 7, 1 2 3 4 5 8, 1 2 3 4 6 7, 1 2 3 4 6 8, 1 2 3 4 7 8, 1
2 3 5 6 7, 1 2 3 5 6 8, 1 2 3 5 7 8, 1 2 3 6 7 8, 1 2 4 5 6 7, 1 2 4 5 6 8, 1 2
4 5 7 8, 1 2 4 6 7 8, 1 2 5 6 7 8, 1 3 4 5 6 7, 1 3 4 5 6 8, 1 3 4 5 7 8, 1 3 4
6 7 8, 1 3 5 6 7 8, 1 4 5 6 7 8, 2 3 4 5 6 7, 2 3 4 5 6 8, 2 3 4 5 7 8, 2 3 4 6
7 8, 2 3 5 6 7 8, 2 4 5 6 7 8, 3 4 5 6 7 8]

Frequent 7-itemsets:

[1 2 3 4 5 6 7, 1 2 3 4 5 6 8, 1 2 3 4 5 7 8, 1 2 3 4 6 7 8, 1 2 3 5 6 7 8, 1 2
4 5 6 7 8, 1 3 4 5 6 7 8, 2 3 4 5 6 7 8]

Frequent 8-itemsets:

[1 2 3 4 5 6 7 8]

Execution time is: 0 seconds.

The Maximum Sequence is: 1 2 3 4 5 6 7 8
```

CONCLUSION & FUTURE OUTLOOK

6. CONCLUSIONS AND FUTURE WORK

We introduced a new problem of mining sequential patterns from a database of customer sales transactions and presented an algorithm for solving this problem. Experiments show that AprioriAll scale linearly with the number of customer transactions. They also have excellent performance properties with respect to the number of transactions in a customer sequence and the number of items in a transaction. In some applications, the user may want to know the ratio of the number of people who bought the first k + 1 items in a sequence to the number of people who bought the first k items, for 0 < k <20 AprioriAll will become the preferred algorithm.

- **I plan to extend this work along the following lines:**

1. **Extension of the algorithms to discover sequential patterns across item categories.**

   An example of such a category is that a dishwasher is a kitchen appliance is a heavy electric appliance, etc.

2. **Improve the efficiency of the mining sequential pattern algorithm:**

Efficient algorithms to mine frequent patterns are crucial to many tasks in data mining. Since the Apriori algorithm was proposed in 1994, there have been several methods proposed to improve its performance. However, most still adopt its candidate set generation-and-test approach. In addition, many methods do not generate all frequent patterns, making them inadequate to derive association rules. I will implement pattern decomposition (PD) algorithm that quickly reduces the size of the dataset on each pass making it more efficient to mine all frequent patterns in a large dataset. The proposed algorithm avoids the costly process of candidate set generation and saves a great amount of counting time with reduced datasets. Our empirical evaluation shows that the algorithm outperforms Apriori by one order of magnitude and is more scalable.

REFERENCES

7.REFERENCES

7.1 Reference book

7.1.1 Jiawei Han, Micheline Kamber. *Data Mining: "Concepts and Techniques"*, HARCOURT INDIA PRIVATE LIMITED, 2001.

7.2 Journals

7.2.1 R. Agrawal, T. Imielinski, A. Swami: ``Database Mining: A Performance Perspective". IEEE Transactions on Knowledge and Data Engineering, Special issue on Learning and Discovery in Knowledge-Based Databases, Vol. 5, No. 6, December 1993, 914-925.

7.2.2 Rakesh Agrawal and Ramakrishnan Srikant. "*Fast algorithms for mining association rules in large databases*". In Proc. of the VLDB Conference, Santiago, Chile, September 1994.

7.2.3. R. Agrawal, R. Srikant: ``*Mining Sequential Patterns", Proc. of the Int'l Conference on Data Engineering (ICDE)*, Taipei, Taiwan, March 1995.

7.3 Websites

7.3.1 IBM Corp.. (1995) "*Data mining - an IBM overview*", http://ibm.com. IBM's view of data mining - explains the data mining techniques in some detail, July 06, 2001

7.3.2 *Analysis of Data Mining Algorithms* by Karuna Pande Joshi, www.more.net/~karuna, July 29, 2001

7.3.3 The Parallel Computer Centre, nor of The Queen's University of Belfast. "*what is data mining* ", www.qub.ac.uk , July 30, 2001.

APPENDICES

8. APPENDICES

List of Abbreviations in Data mining

KDD	Knowledge Discover in Database
AOI	Attribute Oriented Induction
ARCS	Association Rule Clustering System
TID	Transaction ID
OLAP	On- line analytical Processing
OLTP	On Line Transaction Processing
OLAM	On- line analytical Mining
DBMS	Data Base Management System.
MDAC	Microsoft Data Access Components
DMM	Data Mining Model
ODBC	Open Data Base Connectivity
PD	Pattern Decomposition
FP	Frequent Pattern
RDBMS	Relational Data Base Management System.