FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Machine Learning for Supermarket Data Analysis

Salik Ram Khanal

WORKING VERSION

Master Thesis

Supervisor: Prof. Jaime dos Santos Cardoso

July 26, 2016

Abstract

The number of supermarkets is increasing day by day but comparatively the number of customers are not increasing in the same ratio. In the early days, the price was the only differentiator factor among retailers. Nowadays, many factors need to be taken into consideration to provide better services to the customers. The application of new technology in the supermarkets, and in retail in general, is an important factor to enhance the business and customer satisfaction. Reducing the shopping time, queuing time in the checkout, payment time, etc., are most important factors for customer satisfaction. Automatic checkout systems are one of the technologies used in the supermarket for customer satisfaction but they rely in the supermarkets' scanning system. The use of the smartphone is an innovative technique in many fields, and it also has higher potential in recent supermarket applications. The customer can use his own mobile application to scan the barcode at the time of shopping. If a customer scans items' barcode himself, and a timestamp is associated with each item, then, the list of items in each basket is recorded as an ordered set. This extra information enables the mining of additional knowledge about the clients' behavior and preferences. For instance, the shopping behavior of the customers who spend more time inside the supermarket may be different from those customers who spend less time.

In this thesis, we try to utilise information about the shopping duration to provide retailers with additional business knowledge. Using a data mining approach, we explore diverse methodologies like clustering, association rule mining, and sequential pattern mining. Each transaction is clustered using two features, one is shopping duration and another is total price. Finally, the association rules and sequential pattern mining are applied in each cluster. It is found that the association of item and frequent sequence items are different for each cluster and depend on the shopping duration and total price. For the purpose of the association rule mining and sequential pattern mining, the selected features include a list of items in each transaction, date, time, etc. Sequential pattern mining for ordered itemset is applied to find the most frequent patterns of the items. The information of the most frequent patterns can be applied to best path analysis, the arrangement of items inside the supermarket, promotion of items, the suggestion of items while preparing the shopping list, etc. Many sequential

pattern mining algorithms are analysed and evaluated, leading to the selection of the most relevant ones. To better explore the characteristics of the data in our problem, we propose modifications to one of the state of the art methods, achieving an improved performance.

Table of Contents

List of Figures

List of Tables

Chapter One: Introduction

1.1 Overview

Data mining is a process of analysing data from different perspective to extract hidden knowledge which can be used to increase revenue or cost, whereas machine learning techniques are the application of a defined model on a defined data set. Knowledge is the information that is extracted by using data analysis techniques, and the overall steps of knowledge extraction from raw data is called knowledge discovery in database (KDD). It covers a wide area, including business analysis. KDD is a step by step procedure for data analysis from an understanding to visualization.

The steps of KDD are described next:

Understanding: In this step, the objective of the problem is defined. In supermarket data analysis, understanding represents the type of information that is going to be extracted from the supermarket data.

Selection of dataset: Here, to achieve the predefined information or knowledge, what data or features are required; should be defined. The data may be stored in the database in many tables. Therefore, we select tables and then features (fields) from selected tables.

Pre-processing: The data in the database may have huge in storage size which could not be useful for processing instead increasing more burden, therefore, data need to refine, so that noise and irrelevant data are removed.

Data transformation: From a large set of features, we select only appropriate tables and its features. For example, in supermarket sales data, a table may have many fields which are not appropriate for data mining algorithms.

Selection of data mining task and algorithms: Proper selection of data mining and machine learning technique plays a very important role to get appropriate information. Possible machine learning tasks are classification, clustering, association rule, regression, sequential pattern discovery, etc. Among them, we need to define

appropriate techniques according to objectives in the first step. After selecting the machine learning tasks, we need to select particular algorithms related to that task. For example, if we are going to use sequential pattern mining, then, we need to decide which algorithms among existing are going to be implemented or make algorithms ourselves. A large number of machine learning algorithms are introduced and used in the last decades. The selection of machine learning algorithm totally depends on the dataset and result required. For example, for some dataset, one algorithm is appropriate, whereas for another dataset another algorithm may be appropriate.

Pattern evaluation: After getting a result of data mining or machine learning, we need to evaluate the result and approve whether the result is right or wrong. If we are using more than one algorithm, then it is important to evaluate and compare their performance so that we could select the best one. In Figure 1.1, pattern evaluation is included in the selection of data mining and machine learning task and algorithm.

Visualization or Interpretation: It is called post processing step of KDD where the extracted knowledge is represented in an appropriate way so that we can perceive knowledge easily from the result. Any graphical visualization can be done in this step. The KDD process is illustrated in the following figure. The KDD process begins from the selection step and proceeds until knowledge as shown in the figure below.

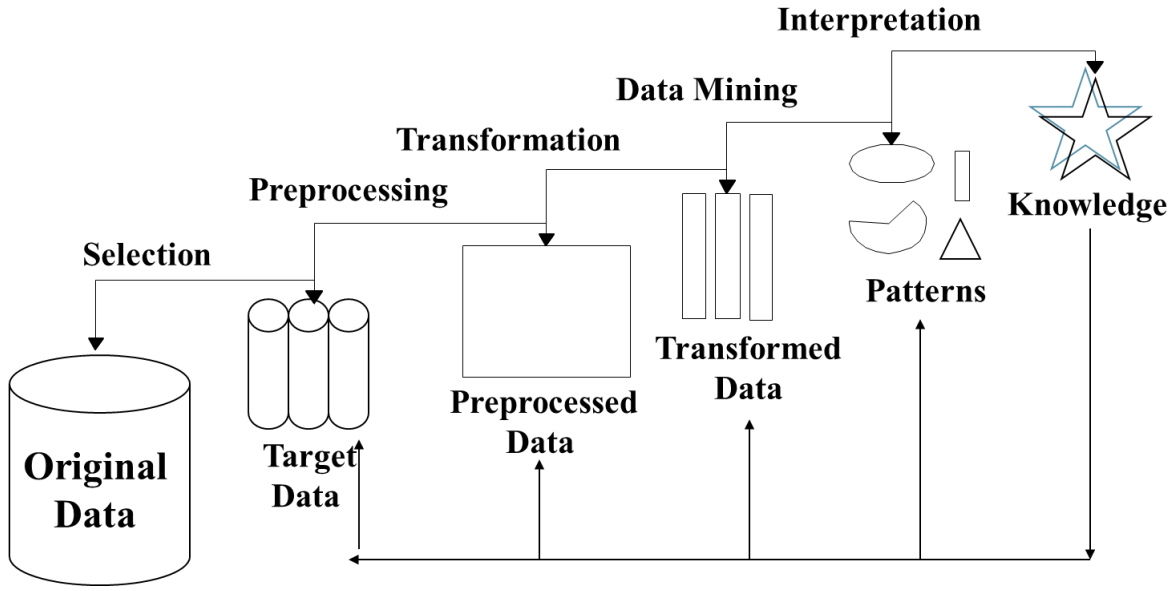

Figure 1.1: KDD process steps

In this thesis, real data from supermarket sales, collected by a company called Xhockware, is analysed by applying different types of machine learning techniques. Xhockware is a company that develops a system for supermarket retailers to create an innovative solution. The main objectives of supermarket retailers regarding customers are to satisfy them in their shopping. The factors affecting the satisfaction of customer are price, time and easy access. Finally, the objectives of the retailer are to increase their profit, which is directly proportional to customer satisfaction.

To help retailers achieve their goals, Xhockware develops a self-checkout system using smartphone technology. One of the main problems for customers in supermarkets is the waiting time in the queue for the payment in the checkout. The cashier also needs to spend a lot of time scanning the items. To overcome this problem, Xhockware also develops a mobile application which can be used by the customer to scan barcodes himself/herself with his/her mobile phone (a mobile application needs to be installed on their mobile phone).

The customers interact with this application in the following steps. In the first step, the customer starts the mobile application, selects the particular shop where he is going to shop by scanning the QR (Quick Read) code that is at the entrance of the store. After scanning the QR code, he/she or that mobile is registered in the system. In the second step, the client scans the product barcodes before putting them in the basket. In this way, he/she scans the barcodes of all items whatever they need and puts them in the basket. To increase usability, the mobile application has additional functionalities. For example, it allows increasing the number of items of the same product, editing a selected item, knowing the total price, discount information, etc. After collecting all the items, in the third step, the customer goes to a checkout system and scans the QR code at checkout. As soon as customer scan the QR code, his total price, and the overall bill are displayed in the POS (Point Of Sell) and he/she can pay by cash or by bank card or direct from mobile. To overcome the problem of theft of items, and to minimize errors in checkout, a subset of items may be randomly cross-checked.

1.2 Problem statement

To apply machine learning to supermarket data, it is important to map the business questions into the machine learning task. It is essential to define the types of data, as well as machine learning models that are appropriate for the particular problem. For example, to predict the sales of a particular month, it is necessary to select sales records from the previous months. Here, problem formulation can be done in two stages.

a) Business questions: A collection of all relevant business questions regarding the automatic checkout, sales, and purchase are taken from Xhockware where they collect business questions from the retailer as well from the customer using questionnaire or interview. The company collects all the business questions for retailer including customer satisfaction and increasing the profit. Some business questions are collected from the experts in the company.

b) Machine learning model design: Machine learning techniques like classification, clustering, association rule, sequential pattern mining, frequent sequence mining is used for data analysis, trying to provide answers to the business questions previously identified.

Some of the frequently arising business questions are as follows.

Which items are related to each other? Which items are bought together? This question is important for the arrangement of items in retailer house, advise the next item while preparing a shopping list or at the time of shopping and provide a discount coupon. For example, milk and bread are correlated, therefore it will be better to put bread near to milk. In addition, if a customer put milk on his shopping list, the system could advise him bread to put on the shopping list. This information can also be used to provide instance discount coupon for the particular customer while preparing a shopping list or while buying the items.

How can we classify a customer or an item or a brand? According to the shopping behavior of a customer, we can classify the customers so that we can predict customer's shopping behavior in future. In the same way, we can classify items/brand, which is the most demanded item/brand by customers, etc. This is important for prediction of sales for the target group of customer. Another application is clustering, where we can cluster the customer according to their shopping behavior. From the

shopping behavior of a particular cluster, the retailer can provide some beneficial schemes for that particular cluster group.

Which items are most likely for promotion or discount?

Which factors are important to define discount in items?

Which items are mostly returned or changed?

If a brand or particular item is not sold, then it is important to find the cause why that item or brand is not sold. Has a customer changed brand or stopped buying an item in the supermarket? For example, if a customer stops buying an item or brand (e.g. regular customers of Sagres Beer stop buying this brand). In this case, we can predict whether he/she is switching to another brand or stop buying this item.

Can we predict the sales for next month?

Which are most demanded items using the smart checkout system?

Which groups of people are most interested in the smart checkout system?

1.3 Motivation of study

Data mining is a milestone technology in the past three decades for knowledge extraction and decision making. The supermarket also has many relevant questions which can be solved by data mining and machine learning techniques. A large amount of data is collected in a supermarket from the sales record, items' information and customers' information from loyalty card. This data can be used to extract useful knowledge for customer satisfaction and to increase the overall profit of the retailer. Thousands of records are recorded in a supermarket system within an hour. Therefore, before applying knowledge extraction rules, it is important to organize the large data in a suitable manner in cloud storage. If we extract useful information in real time, then the speed of processing is also an important factor. Therefore, proper database management, data warehousing can be applied to solve this problem.

Customer satisfaction is an important factor for supermarket retailers to enhance their business. As the use of technology in each field is growing rapidly, supermarket retailers are looking for better technology in their business. *Therefore, the retailer has deployed kiosks, interactive displays, handhold shopping devices, and computer-enabled grocery carts to assist with store navigation, provide detailed product*

information, offer personalized product recommendations and promotions, and expand the available selection of merchandise (Burkeb, 2007).

An important factor of customer satisfaction is the time spent in the supermarket. Everybody wants to spend less time inside the supermarket retailer house; therefore, supermarket should always be careful about this concern of the customer. The time can be reduced in two ways; one is to reduce the queuing time in checkout and the other with the proper arrangement of items in shelves: items that are bought together should be arranged together.

We can extract very important information by analyzing sales information. In recent years, many research papers have been published which mainly focus on association rule, sequential pattern mining, clustering customer etc. although that do not solve all the business issues. Till now, all the researcher uses the dataset which does not keep the information of order in which clients pick items and put them in the basket. The order of items in which customer pick and put in basket plays very important role for analyzing shopping path, placement of item, etc. The dataset available in Xhockware has this type of information because the barcode of the item is scanned by user herself/himself. When a customer scans and selects/confirms that item to put in a basket, the detail information is automatically stored in the database so record in the database has information about the order of items.

1.4 Outline of thesis

The report is divided into seven chapters including this introductory chapter. Chapter two includes all the literature review. Some related theory and machine learning algorithms are described in chapter three. Main techniques and findings are mentioned in chapter four and chapter five. Chapter four is about methodology and data characterization. All the algorithms used to compare the performances are introduced in methodology chapter. All the findings and results are mentions in chapter five called result and finding. All the results of the experiments are described in tabular and graphical form. The discussion of results and findings are mention in chapter six. At the last, everything is concluded in the chapter seven.

Chapter Two: Literature Review

Extracting valuable information from existing data is one of the most important tasks in business organizations. In retail, machine learning techniques like classification, association rule, clustering, sequential pattern mining, etc., can be used to extract some hidden knowledge. All the steps performed to achieve the above goals from collecting raw data are called data mining. It is an ideal vision of maintaining a central repository of all organizational data. Before applying various techniques on data, we must be careful about the proper organization of data. Proper organization of data is achieved by data warehousing. Online transaction processing (OLPT) systems put and access data in the database quickly, but do not return the result in a good manner. The information stored by using OLPT can be analyzing KDD processes (Shivappa M Metagar, 2014).

The supermarket could find a more profitable way by focusing on customer satisfaction. Day by day, the number of retailers is increasing, but not the consumers. The supermarket should focus on a better selection of health foods, registered direction free advice, fast service checkout lines, better lighting and lower shelves. The purpose of customer profiling is to target valued customers for special treatment based on their anticipated future profitability to the supermarket. Data mining techniques are suitable for profiling the customers due to their proven ability to recognize and track patterns within the set of data (Min, 2007). We can classify customer according to their shopping behavior. It is observed that less than 20% of customers are worth 80% of the company's revenue: a precise characterization of the high-spending classes of customers, as well as other classes and niches, is, therefore, an essential piece of knowledge to develop effective marketing strategies (Giannotti, 1999).

The way of data collection: Data for knowledge extraction can be collected in different ways. A simple way to collect data is a questionnaire. Most of the important data in supermarket data can be collected from each transaction of goods that are recorded in databases. The data collected in a database can be sales or demographic data, or items information. One of the reasons that the data is getting bigger is the continuous generation from more sources and more devices. As a result, this data is often made up of volumes of text, dates, numbers, and facts that are typically free-

form by nature and cannot be stored in structured, predefined tables. The most important data can be POS data. All the sales information is recorded in the database while paying in POS. Each customer is identified by their identity in loyalty card. Although the purchasing data of a customer can be aggregated over several time levels such as day, week, and season, it is not easy to determine the most effective time level for conducting an analysis. The method of data collection would handle all necessary time levels such as years, seasons, and months, simultaneously, and can extract both independent time level patterns and interrelationship patterns among the time levels used (Morita, 2006).

The data can be in several different formats like a database table, CSV, text, excel etc. For the analysis purpose, different types of variables are considered; in terms of machine learning, variables are called features. In knowledge discovery process, the data should be in proper format. Converting the raw data into the proper format is the initial step of KDD process, called pre-processing. Variables related to the spending, frequency, promotion behavior and response on mailings all have a good predictive performance. The other variables, such as regency, inter-purchase time, the length of relationship, average spending per visit, returns of goods and distance to the store clearly exhibit lower univariate predictive performance (Wouter Buckinx, 2007).

Many hypothesis techniques can be used which validate some intuitive promises, casual inferences made by hypothesis testing may not be sufficient to predict behavioral patterns of grocery shopping. To overcome this, data mining techniques can be used (Min, 2007).

The simple technique for predictive accuracy is If-Then-Else rules. Using this technique, we can use a decision tree. For example, if the customer is more frequent, he/she tends to purchase fewer items per visit than does the less frequent shopper. If the customer is female or married, then she buys more glossary items. Therefore, customer's age, group-shopping behaviors, the frequency of visit the supermarket, the profession of the customer may affect the behavior of shopping. To retain the more frequent customer, supermarket management should cater their services amenities, e.g. greater availability of items, easy access to items, easy checkout, etc. The supermarket where unmarried or single customer arrives more most focus on the self-

checkout system. For example, the supermarket near campus should focus on self-checkout (Min, 2007).

Major finding of (Min, 2007) are:

- The supermarket target for married customer bases may offer a wide assortment of food and drinks that are catered to different needs to a married couple.
- The age group of less the 20 and greater than 50 needs assistance from supermarket employee; therefore, prefer cash payment.
- The age group of greater than 50 are more sensitive to price, therefore, offer more cost saving schemes.

2.1 Clustering and association rule

To increase sales and profit, the retailer needs to determine the proper arrangement of items so that the customer can easily access the items. The 70% of shopping decision depends on the display of the item. For the small retailer, we need island-type (all Item are located in the same location) shelf-space allocation whereas grid-type (items are located in the grid) allocation is suitable for big supermarkets. The shelf-space allocation can be achieved by mining the sales data. According to the correlation between commodities, we can form clusters that can be used for island-type allocation (Jiaguo, July 2010).

Clustering techniques can also be used to group customers of same shopping behavior or group the items. The promotions can be provided for a particular cluster of the customers so that overall profit of supermarket will increase. Seasonal promotions can be provided for particular groups like Deepavali festive season which can improve the purchase and the profit (Kumar, 2012).

Association rules are also important to optimize different factors like the cost for placement of items and to allocate a maximum number of items in the same area. The same technique can be extended including the concept of revenue optimization to make the same technique, even more, business perspective. Various optimization techniques can be used to find the most suitable patterns since the proposed technique mine a large number of sequential patterns (George Aloysius, 2012).

2.2 Sequential Pattern Mining

The order of an item in a basket plays a very important role in knowledge extraction; therefore, by using sequential pattern mining methods, we can extract additional information. Some literature is discussed below.

Generalized Sequence Pattern

The association rule and sequential pattern mining model were first proposed by (Rakesh Agrawal, 1995). The main steps for the sequential pattern mining are: Find frequent items, generate candidate sequences and check which candidates are frequent itemset. The authors proposed two algorithms, AprioriAll and AproiriSome, for sequential pattern mining. AprioriSome checks all the candidates while AprioriSome checks only the maximal sequences. AprioriSome algorithm is suitable when the minimum number of customers that must support a sequential pattern is low. In this algorithm, the candidate generation does not have a pruning step, therefore it has more search space and slower performance. If a customer buys items after six months of another item, then it does not indicate that these items are sequential. So, the application of a sequential pattern mining algorithm to these types of sequences does not have any significance rather than occupying more memory and slower the speed of execution.

In 1996, the same author of Apriori algorithm proposed another algorithm which considers some constraints. The proposed algorithm is called GSP (Generalized Sequence Pattern). They consider three constraints; time, sliding window and taxonomy. This algorithm has some improvements in candidate generation and a pruning phase. While joining the candidate, it joins only those sequences which has contiguous subsequence's; it also can apply time constraints, sliding window and taxonomy in this phase. Next step is counting phase where sequences are stored in the hash tree, making it easier and faster for sharing the sequence whether exit or not in sequence. The check whether a sequence is in another sequence or not is performed in two phases: forward and backward. It is up to 5 times faster than Apriori algorithm. The performance depends on the number of users and the number of transactions; the scan-up is linear (Ramakrishnan Srikant, 1996).

Vertical database search

Another milestone in sequential pattern mining was the work by Zaki (ZAKI, 2001). It uses vertical searching instead of using horizontal search. It uses vertical id-list database format, a lattice-theoretic approach to decompose the original search space. It not only reduces the IO (Input Output) cost by reducing the database scans but also minimizes computational cost by using efficient search schemes. The dataset contains customerID, timestamp, and sequence of items. It does not need a hash tree and complete search techniques. It has a very good scan up ratio. The performance of algorithm is better for higher number of items.

Even though considering some factors to speed-up the execution and reduce the memory occupied, the above mentioned algorithm does not consider how the obtained sequence are important. It does not consider whether or not an obtained sequence gives meaningful information. (Cláudia Antunes, 2004) proposes an approach that considers a factor that filters the important patterns. This paper proposes an algorithm called ε-accepts. While generating the k-sequences from the candidate sequences, besides applying time constraints, sliding window, and taxonomy, they proposed a constraint which is inputted from the user. The input from user depends on the requirements of output from the database. It increases the speed of operation and removes unnecessary sequences from the databases. The approximated constraints are expressed as deterministic finite automata (DFA) and they proposed an algorithm called ε-accepts. To check whether a sequence is accepted or not, the algorithm calculates the generation cost. If the generation cost is less than the constraint input by the user, then that sequence is considered as candidate sequence however other constraints like time constraints, sliding windows, and taxonomy can be applied simultaneously. DFA is very an innovative terminology in pattern mining.

Projected Database

The main factor impacting on execution speed and memory occupation is the number of candidate generation from the frequent items or frequent itemset. If we generate less number of candidates, then the performance will drastically change. (Jian Pei, 2004) proposed one of the most effective approach which controls the number of candidate generation. In this model, instead of creating the candidate sequences, it creates projected database and searches on the basis of prefix and postfix of itemset.

It describes three approaches, freescan, prefixscan and pseudoprojection methods. All the methods use projected database; the only difference is the way of creating the projection database. These methods can also apply some constraints. Sequence databases are recursively projected into a set of smaller projected databases based on the current sequential pattern(s), and sequential patterns are grown in each projected database by exploring only locally frequent fragments.

The speed of execution and memory occupation mainly depend on data structure and data representation in memory. If we represent data in a binary or bitwise format, the speed will dramatically increase. We can use the algorithm using a different data structure (Jay Ayres, 2002). The search strategy plays a very important role in speeding up the execution. It is the depth-first search strategy that integrates a depth-first traversal of the search space with an effective pruning mechanism. The concept used here is a representation of the database in a bitmap format. It generates sequence tree that contains null as root item and its extended sequence in a child. For example, if we generate the sequences of a and b, its' children may be aa, ab, (a, b). This algorithm was compared with SPADE and PrefixSpan and got better result with all conditions in SPADE and better with some conditions in PrefixSpan. This algorithm is better for the larger database but has similar results for the small database. We can also represent data in tree data structure. (Kirti S. Patil, 2013) proposed a model which use Apriori algorithm and FP-tree for candidate generation. Not only minimum support but also maximum distance (Maximum sequence between items) can also be considered to check whether a particular sequence occurs or not. There are three types of patterns; periodic, statistically significant, and approximate patterns. In the periodic pattern, a period of occurrence of the sequence is considered. In statistical significant pattern, we need to calculate the probability of occurrence of items whereas in the approximate pattern, a compatibility matrix is constructed. to compatibility matrix, sequential patterns are determined (Gabor, 2010).

The number of candidate generation and number of frequent pattern generation depend on the minimum support considered. We can also consider more than one minimum support that depends on its natural frequency. (Ya-Han, 2013) proposes PLMS-tree (Pre-order Linked Multiple Supports tree) that is a compact tree constructed from the entire database.

We can also apply extra information on sequences like time interval between items in sequences which plays important roles on data analysis and knowledge extraction. For example, if a person buys a digital camera, then, he/she will buy memory card within one month. To find the time interval between every pair of items, we can use clustering algorithm. By using normal Apriori for generation of candidate sequences and use mining algorithm using both minimum support and time interval between items. The main concept of this algorithm is to reverse the original sequence dataset to enhance the efficiency for searching the target patterns. In addition, the clustering analysis is used to automatically generate the suitable time partitions between successive itemset so that the traditional algorithms can be extended to discover the time-interval sequential patterns without pre-defining any time partition (Chueh, 2010).

Another model in sequential pattern mining is consideration of profit in the model. The final aim of any research or knowledge discovery is to increase the overall profit, therefore, it is important to consider the most profitable sequences in sequential pattern mining. High-profit items may not be listed in frequent sequences because of its frequency. According to the profit of individual item, utility measure is calculated and a high utility sequential pattern from the quantitative sequential database is found (Guo-Cheng Lan, 2014). When the profit value of sequence is higher, the sequence is being given importance irrespective of number of items.

(George Aloysius, 2012) proposed two stages model which mainly considers the profit of items and this model can be implemented for the product placement.

Another model is the high utility sequential pattern considering the profit of each sequence which is calculated from profit table and a number of items bought by the customer (SHOW-JANE YEN, 2013). The sequence's utility is compared with minimum utility threshold. The algorithm is based on PrefixSpan which generates projected databases recursively.

(Chandni Naik, 2014) proposed RFM-Q (Recency, Frequency and Monetary-Quantity) algorithm that discovers a quantity of items which is purchased by the customer satisfying certain criteria. The extracted knowledge can be used for sales promotion. It compares the result with RFM-based sequential pattern mining. RFM sequential pattern mining does not consider the quantity of items which is used in RFM-Q. It also considers the prefix scan but the projection database is different from the normal

prefixscan. The proposed algorithm takes little more time but produces less number of sequential patterns and patterns are more valuable. A real life data set was used in their experiments. The result shows that the proposed method is better than the RFM-based sequential pattern mining in terms of a number of generated patterns and also retains more meaningful results for the user.

By using sequential pattern mining techniques, a large number of frequent sequences are generated, therefore, it is very difficult to find a most valuable patterns from them. (Philippe Fournier-Viger, 2014) propose pruning candidates based on the study of co-occurrences. They present a new structure named CMAP (Co-occurrence MAP) for storing co-occurrence information. The main theme of CMAP is that the generated frequent sequence will not be the subsequence of any other frequent sequence. They explain how this information can be used to prune candidates in SPADE, SPAM, and ClaSP algorithms.

Chapter Three: Machine Learning Algorithms and Data Characterization

3.1 Clustering

Clustering is an unsupervised machine learning technique where objects are grouped in such a way that objects of the same cluster have more similar features than objects from different clusters. It is done without any learning or training of instances. K-means algorithm is one of the most common algorithms for clustering. K-means clustering partitions n observations into *k* clusters in which each observation belongs to the cluster with the nearest mean. The total number cluster indicated by K is defined at the beginning of algorithm. An object belongs to that cluster defined by centroid which has the least distance to all the clusters' centroid.

Consider a set of observations ($\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$), where each observation is a *d*-dimensional real vector, *k*-means clustering aims to partition *n* observations into k ($\leq n$) sets $\mathbf{S} = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares (WCSS) (sum of distance functions of each point in the cluster to the K centre). In other words, its objective is to find:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 \qquad \text{…..……...(3.1)}$$

where $\boldsymbol{\mu}_i$ is the mean of points in S_i.

3.2 Association rule

The relationship between goods can be measured by using association rule. Association distance of particular item with other items can be calculated using the concept of how often these items are bought together. If beer and peanut are often bought together, we can say that these items are associated with each other. The Apriori algorithm is the mostly used association rule mining algorithm. Mathematically, the association distance between item A and B can be calculated using formula.

$$\text{dist (A, B)} = 1-\left(\left(\frac{count(A,B)}{count(D)}\right)\right)^2 \text{ ……..…… (3.2)}$$

Where dist (A, B) is association distance of item B from A.

count (A, B) occurrence of both A, B in D number of transactions.

count(D): Number of transactions

Association rules be not only applied to locate the item but also to promote the items. The promotion (discount) of one item can increase sales of another item. For example, if supermarket provides a discount in beer, then, sales of associated items such as peanuts, fried potato, meatball etc. will increase and finally overall profit will be more than without discount.

The implication expression of form X=>Y, where X⊂I and Y⊂I and X and Y are disjoint sets. The strength of association can be measured in terms of support and confidence. The rule X=>Y holds in transaction D with confidence C and support S where c% of transactions in D that contains X also in Y and S% of the transaction in D contains X⊂Y (Abdulsalam, 2014).

Association rule mining process could be divided into two phases (S.O. Abdulsalam, 2014).

1. Frequent item generation: Find all the frequent items from a transaction database.
2. Rule generation: This is to extract the entire high confidence rule from frequent item set found in the first step. This rule is called strong rule.

3.3 Sequential pattern mining algorithms

Sequential pattern mining is a data mining technique finds the most frequent subsequences from a large set of sequences. The main application fields of sequential pattern mining DNA analysis, web page access pattern analysis, text mining etc. In the supermarket, sequential pattern mining is applicable for product placement, analysis of shopping behavior of customer, shopping path analysis etc. In the supermarket, a list of items from the basket is taken as sequences and frequent subsequences are searched in those sequences. We can also apply some constraints like profit, gap between sequential patterns, etc.

Sequential pattern mining is a mining technique concerned with finding statistically relevant sequences from the database containing sequences of items. The data used

for sequential pattern mining must be discrete. The main purposes of sequential pattern mining are finding frequently occurring patterns, comparing sequences, clustering patterns, recovering missing sequence members, etc. The main steps followed by common algorithms (but not all algorithms) are: find frequent items with length-1 or frequent items, then generate the candidate sequences of length-2, and check each candidate whether it is frequent of not and so on.

Many algorithms are proposed and implemented in sequential pattern mining. Among them, only seven algorithms are selected to compare performance in this thesis. The performance of algorithm depends on the dataset used. An algorithm may give better performance for particular dataset whereas the same algorithm may not give good performance for another dataset. The dataset used and expected output in this thesis are different from the dataset which are commonly used. Therefore, it is important to compare algorithms with real data of Xhockware. The brief introduction and psuedocode of some relevant algorithms are listed below.

3.3.1 PrefixSpan Algorithm

This algorithm works on the projected database. A new database is created from a sequential database so that the search space will be smaller in each step. The database projection concept can be applied in three different ways; freescan, prefixscan and pseudoprojection. Each algorithm may use different way of creating projection database. Some constraints like time gap, taxonomy can also be applied in its extended form of algorithm. In each iteration, projected database is created from the sequential database. The size of projected database is decreasing in each iteration therefore the search space is reduced. The projected database depends on frequent items created in previous step. General steps are:

1. Create length-1 frequent items.
2. Find projected databases for each frequent item.
3. Scan only projected database for a particular frequent item or itemset to find lenth-n+1 frequent sequence pattern.
4. Similarly, for lenth-n+1 patterns, find projected database and scan corresponding projected database.

5. repeat step 3 and 5 until the frequent item becomes empty.

Main differences in three algorithms of prefix method is the way of creating the projection database of each frequent item or itemset. In freespan, for each length-n item or itemset, find projected database where non-frequent items are removed from the sequences and need to find new projected database for length-n+1 database.

In prefixscan, the projection database for length-n+1 sequences are created from prefix items of the length-n item so that projected database will be less complex.

In pseudoprojection, the index is used, from that index position, the search operation starts to count support. The pseudocode is given below (Jian Pei, 2004).

PrefixSpan Algorithm

PrefixSpan (α', l+1, S|$_{α'}$).

The parameters are

α is a sequential pattern;
l is the length of α; and
S|$_α$ is the α -projected database if α≠<>, otherwise, it is the sequence database S.

Method:

Scan S|α once, find each frequent item, b, such that

a) *b can be assembled to the last element of α to form a sequential pattern; or*
b) *<b> can be appended to α to form a sequential pattern.*

For each frequent item b, append it to α to form a sequential pattern α', and output α'.

For each α', construct α' projected database S|α', and call ***PrefixSpan (α', l+1, S|α').***

3.3.2 SPADE

It uses vertical database format where it maintains disk based id-list of each item or itemset. In first step, horizontal database format is converted into vertical database format. Then, it creates the id-list of each item where a id-list is represented in (sid, eid) pair, where sid is sequence id and eid is event id of a particular sequence. In each iteration, it creates the id-list and count of the occurrence and then it decides whether a particular item or itemset is frequent or not. The iterations progress until the frequent item becomes empty. This method reduces the IO cost and search space. The pseudocode of algorithm is given below (ZAKI, 2001).

SPADE algorithm

```
SPADE (min_sup, D)
        F1 = {frequent items or 1-sequences};
        F2 = {frequent 2=sequences};
        Ɛ = {equivalent class [X]θ1};
        For all [X] ϵ Ɛ Enumerate-Frequent-Seq([X]);

Enumerate-Frequent-Seq(S):
For all atoms Ai ϵ S do
        Ti -= Φ;
        For all atoms Aj ϵ S, with j≥I do
                R = Ai V Aj;
                If (Prune(R) == FALSE) then
                        L(R) = L(Ai) ∩ L(Aj);
                        if σ(R) ≥min_sup then
                                Ti = Ti U {R}; F|R|U {R};
                        End
                        If(Depth-first-Search) then Enumerate-Frequent-Seq(Ti);
                end
                if(Breadth-First-Search) then
                        for (Breadth-First-Search) then
                                for all Ti ≠ Φ do Enumerate-Frequent-Seq(Ti);
                        end
                end

Prune (β):
        for all (k-1) subsequences, α< β do
                if([α1]) has been processed, and α ≠ Fk-1) then
                        return TRUE;
                return FALSE;
```

Based on this algorithm, many algorithms were proposed and proved the better performance than this. One of the most important algorithms is CM-SPADE. It prunes co-occurring sequences which reduces the search space, as well as only necessary sequences will be generated. If a sequence occurs as a sub-sequence of another sequence with the same support, then that sequence is pruned.

3.3.3 SPAM

The main innovative concept used in this algorithm is the data representation in memory. The transactional database is represented by vertical bitmap format. For each transaction, search the occurrence of each item, if the item is present, then, represent it with binary 1 or 0 otherwise. Finally, all the transactions are represented

in bitmap format. From that bitmap transactional database, count the support of each item or itemset and create candidate sequences of length n+1. From the candidate sequence find the frequent items of length n+1. Repeat the steps till occurrence of frequent items. The main advantage of this algorithm is speed of execution and memory occupy. The pseudocode is given below (Jay Ayres, 2002).

SPAM algorithm

SPAM (SDB, minsup)

Scan SDB to create V(SDB) and identify F_1, the list of frequent items.
for each item sϵF_1,
SEARCH (<s>,F_1,{eϵF_1 | e > $_{lex}$ s},minsup).

SEARCH (pat,S_n, I_n, minsup)
Output pattern pat.
S_{temp} := I_{temp} := Φ
FOR each item j ϵ S_n,
IF the s-extension of pat is frequent THEN S_{temp} := S_{temp} U {i}.
FOR each item j ϵ S_{temp},
SEARCH(the s-extension of pat with j, $S_{temp,}$ {e ϵ S_{temp} | e>$_{lex}$,j},minsup).
FOR each item j ϵ I_n,
IF the i-extension of pat is frequent THEN I_{temp} := I_{temp} U {i}.
FOR each item j ϵ I_{temp},
SEARCH(i-extension of pat with j, S_{temp}, {eϵI_{temp} | e > $_{lex}$ j}, minsup).

Based on this algorithm, (Philippe Fournier-Viger, 2014) propose an algorithm with pruning co-occurring sequences. The pruning mechanism is exactly similar to CMSPADE. The proposed algorithm is called CMSPAM. This algorithm is also selected for the data analysis.

3.3.4 BIDE+

This algorithm finds frequent sequences based on PrefixSpan algorithm with some improvements. It generates closed sequences from the database. Closed sequences are those which are not subsequences of any other sequences with the same support. It is not only faster but also generates only necessary sequences that makes easier to analyse the result (Han, 2004). The pseudocode of BIDE+ algorithm is given below.

BIDE+ Algorithm

BIDE (SDB, min_sup, FCS)
INPUT: an input sequence database SDB, a minimum support threshold min_sup
OUTPUT: the complete set of frequent closed sequences, FCS
FCS = Φ;
F1 = frequent 1-sequences (SDB,min_sup);
for (each 1-sequenc f1 in F1) do
    SDB^{f1} = psuedoprojected database (SDB);
for (each f1 in F1) do
    if (!BackScan(f1, SDB^{f1}))

        BEI = backward extension check (f1, SDB^{f1});

        Call bide(SDB^{f1},f1, min_sup, BEI, FCS);

return FCS;

bide (S_p_SDB, S_p, minsup, BEI, FCS)

Input: a projected sequence database S_p_SDB, a prefix sequence S_p,a minimum support threshold min_sup, and the number of backward extension items BEI

Output: the current set of frequent closed sequences, FCS

LFI = locally frequent items (S_p_SDB);

FEI = | {z in LFI | z,sup = sup^{SDB} (S_p)}|;
if((BEI+FEI) == 0)
FCS = FCS U {S_p};
for(each I in LFI) do
$S_p{}^{I} = <S_p,i>$;
SDB^{spi} = pseudo projected database (S_p_SDB, $S_p{}^{i}$);
for (each I in LFI) do
    if(!BackScan($S_p{}^{i}$, SDB^{spi}))
    BEI = backward extension check ($S_p{}^{i}$, SDB^{spi});
    call bide(SDB^{spi}, $S_p{}^{i}$,min_sup, BEI, FCS);

3.3.5 MaxSP

(Philippe Fournier-Viger, 2013) proposed an algorithm with maximal sequential pattern mining. It selects only limited and most important sequences from the sequential database. It is not necessary to store intermediate candidates in main memory, therefore, it saves the memory occupation. The pseudocode for this algorithm is given below.

MaxSP algorithm

MaxSP (a sequence database SDB, a threshold minsup, a prefix P initially set to ⟨⟩)
largestSupport := 0.

Scan SDB once to count the support of each item.

FOR *each item i with a support ≥ minsup*

P' := Concatenate(P, i).

SDBi := DatabaseProjection(SDB, i).

IF the pattern P' has no maximal-backward-extension in SDBi THEN

maximumSupport := MaxSP (SDBi , minsup, P').

IF maximumSupport < minsup THEN OUTPUT the pattern P'.

IF support(P') > largestSupport THEN largestSupport := support(P')

RETURN largestSupport.

3.3.6 VMSP

Maximal sequential pattern mining based on vertical search approach is proposed by (Philippe Fournier-Viger, 2014). This algorithm is similar to MaxSP except it uses vertical mining. The performance is better than MaxSP due to vertical searching strategy. The pseudocode is given below.

VMSP Algorithm

PATTERN-ENUMERATION*(SDB, minsup)*
Scan SDB to create V(SDB) and identify S_{init}, the list of frequent items.
FOR *each item s ∈ S_{init},*
SEARCH *(⟨s⟩ , S_{init}, the set of items from S_{init} that are lexically larger than s, minsup).*

SEARCH *(pat, S_n, I_n , minsup)*
Output pattern pat.
Stemp := Itemp := Φ
FOR *each item j ∈ S_n,*
IF *the s-extension of pat is frequent THEN S_{temp} := S_{temp} ∪{i}.*
FOR *each item j∈ S_{temp},*
SEARCH *(the s-extension of pat with j, S_{temp} , elements in S_{temp} greater than j, minsup).*
FOR *each item j ∈ I_n,*

IF *the i-extension of pat is frequent THEN* $I_{temp} := I_{temp} \cup \{i\}$.
FOR *each item* $j \in I_{temp}$,
SEARCH *(i-extension of pat with j,* S_{temp} *, all elements in* I_{temp} *greater than j, minsup).*

3.4 Data Characterization

The data is imported from the Xhockware database, all transactions from 2015-10-01 to 2016-03-30. The complete database of the system contains many tables and schema. The selection of appropriate tables from such a large set of tables of this huge database is a very important task before applying machine learning techniques. After selecting the tables, we have to decide which tables are appropriate to answer each of the business questions. The primary task of this thesis is to find the association of sales items using automatic checkout system. So, we select all the tables belonging to sales information like sales id, date, items, shopping times, the number of items, price discount, etc., from the database, dumping of all records from October 2015. A single table of the database is not enough for data analysis, therefore, we need to select many related tables. A lot of pre-processing of data has been done before applying the data analysis. For the selection of appropriate fields and records from the different tables, SQL query is selected with PostgreSQL database management system. The steps of pre-processing on data are as follows:

- Select all the necessary tables from the relational database
- Apply different types of JOIN like INNER JOIN, OUT JOIN, etc.
- Select the fields or aggregation of fields from the joined tables
- Apply the selection condition: select only those records which are collected on date after October 2015, only for one store or only specific store id, only the records which are from the youbeep application (Xhockware application to manage the shopping process) or automatic checkout application, etc.
- Type casting: the item name in the database is string variable but for the processing, it is easier to use in numeric datatype, therefore convert this field from string data type to numeric datatype. But for most of the experiments, barcode is considered instead of item name.

After performing these all steps, the dataset is formatted with a table containing transaction id, transaction date, shopping duration, the number of products, total price,

and a list of items or list of barcodes. The table in this format can be used for clustering and association rule mining, classification, sequential pattern mining. This is even not a final dataset for each experiment. This dataset format is again modified in the appropriate format before applying different types of machine learning techniques.

In some cases, barcodes are selected instead of using item names because it will be faster to use numeric variables than string variables. All the processing is done in numeric values therefore at the end (post-processing or visualization) barcodes are converted into item name with the help of another table.

Especially, in the case of sequential pattern mining, many algorithms are implemented with many conditions, therefore, the direct use of the complete dataset takes too much time too run all the analysis. To overcome this problem, an arbitrary dataset is generated using an application called SPMF. It is an open-source data mining library written in Java, specialized in pattern mining. By using arbitrary dataset, some of the relevant algorithms are shortlisted and validated with a real dataset. The properties of the arbitrary dataset are described later in the corresponding section where they are used because arbitrary dataset has various data formats.

Chapter Four: Methodology

The primary purpose of the retailer is to increase the customer's satisfaction and finally enhance the overall profit on the supermarket. To achieve these objectives, many machine learning techniques are essential for different types of data. Among machine learning techniques, clustering, association rule mining and sequential pattern mining are important machine learning techniques in supermarket data.

In the supermarket, there are different types of the tables in the database. The data related to sales history is considered for data analysis. Generally, sales history table contains the information about customer id, price, date, time, time spent inside retailer house, time spends to payment, the sequence of items taken etc. Beside this table, some other related tables are important for data transformation and visualization. For the data analysis purpose, database dump from October 2015 until the end of March 2016 is taken into consideration and required tables and features are selected using PostgreSQL queries. Of course, enough time for pre-processing should be spent for pre-processing data since the complete system has huge and complex database schema.

Instead of using real dataset directly, some arbitrary dataset is used in an experiment to test the performance of algorithms and validate the result of some shortlisted algorithm with a real dataset.

4.1 Clustering and association rule

Clustering is a process of grouping instances without using any knowledge or training. Many algorithms are used in clustering. K-Means clustering algorithm is most common and relevant algorithm to find clusters of transactions. K-means algorithm is considered for clustering the instances. Instances are clustered with two features, total time spent inside retailer house and the total price of one basket. The dataset of Xhockware has a unique information which might be very important for the analysis of customer behavior that is information about total time spent inside retailer house. The shopping behavior might be different for the customer who spend more time for

shopping with the customer who spend less time for shopping. Therefore, before testing their behavior with association rule mining, the transactions are clustered with total time and total price features. Only shopping duration, total price and barcode list from the table are selected before applying K-means clustering. Matlab tool is used with two fields shopping duration and total price and adds cluster index value to the corresponding row. Finally, all the transactions (barcode list) are divided into three group which are three clusters.

All the records are divided into three clusters according to shopping duration and total price then association rule are applied in each cluster with the Apriori algorithm. The analysis of the outcome, especially the differences between the three groups, we can be used for different goals. This result can be applied for placement of items, to offer a discount for each customer, to provide the list of items while preparing shopping list, etc. For example, when a customer selects milk in his/her shopping list then we can remind him/her about another associated item in his shopping list. The overall operation is represented in block diagram presented in the Figure 4.1.

Sequential pattern mining technique is also applied in the same dataset and same clusters. The R package is used for sequential pattern mining purpose. R has a package called ‘arulesSequences’ which is an implementation of the SPADE algorithm in R. Like association rules, it is applied in each of the three clusters. In this experiment, the itemset is considered as unordered. For the sequential pattern mining purpose, data format is different from association rule, therefore, customer id, shopping date and time, and list of barcodes are taken from the table for each cluster.

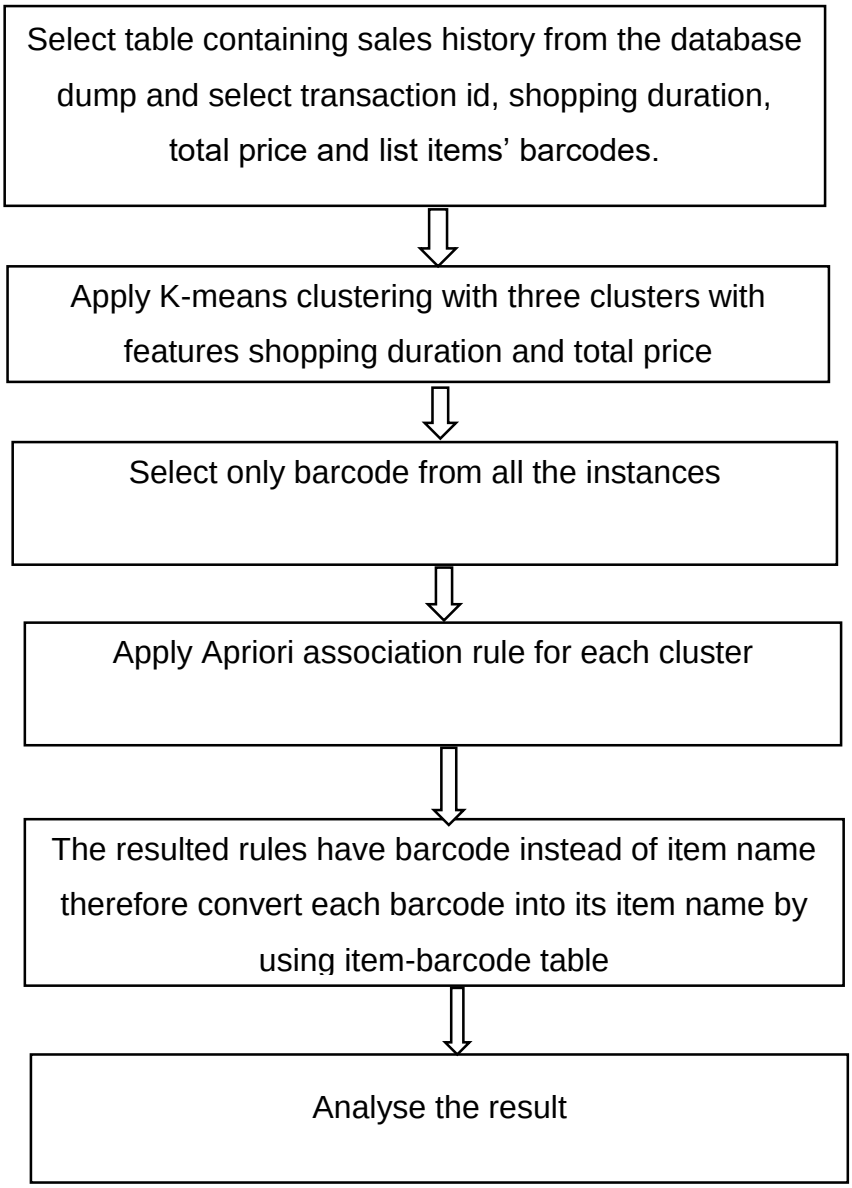

Figure 4.1: Block Diagram for Clustering and Association Rule

4.2 Sequential Pattern Mining with real and unordered itemset

Sequential pattern mining is a process of finding most frequent patterns from a list of sequences. Suppose, we have the sequences of <2 4 5>, <2 5 8>, <1 2 5> then the sub-sequence <2 5> occurs in each sequences therefore it is a frequent sub-sequence with support 3 or 100%. In the case of market basket analysis, we can find frequent sequences with a list of more than one item. Let us take an example of transaction database with following data.

SID	TID	ITEMS
1	1	2 3 4
1	2	1 3 5
2	1	3 5
2	2	4
3	1	2 5 7

(a)

SID	ITEMS
1	< (2 3 5 4) (1 3 4)>
2	< (3 5) 1>
3	<2 5 7>

(b)

Table 4.1: Sample transaction database with sequence id, time and item list

In table (b), each row represents a sequence. we can say that the sequences < (3 5) 1> occur two times. The main difference between normal sequence database and the transactional database is that sequential database contains only one item in itemset but transaction database contains a set of items in the itemset.

There are many algorithms in sequential pattern mining. SPADE algorithm is applicable to all transactional datasets. In this experiment, even though the transaction items are ordered and unsorted, SPADE algorithm use unordered and sorted item. For example, <2 4 3> and <2 3 4> are considered as the same sequence because each transaction sequence is sorted in ascending order before processing. The new property of Xhockware is not implemented in this case. The knowledge we are interested in extracting from this experiment is that what item will be bought by a particular customer in the next transaction.

Sequential pattern mining techniques are used to extract hidden knowledge from the real sales dataset using the youbeep application. For the sample data, we use only the data collected about the sales transaction in the LIDL retailer shop located in Cascais, Portugal. Appropriate tables and fields are taken from database dump of all sales records of all the retailer who are using the Youbeep application. Youbeep application is a mobile application developed by Xhockware. At the beginning, we select all the sales transaction from 1st October 2015 to 30 March 2016. A dataset of this time period is taken because the buying pattern may be similar within the winter season. The buying pattern may be different in summer and winter. Device-id (customer id), a combination of date and time, the total number of products and list of

items in each transaction is taken to find appropriate fields. The identity of the customer can be identified from their device-id from which they are scanning the items. The transaction id is device-id, event-id is a combination of date and time in integer datatype, size is a total number of product in each transaction and list of all items are taken by aggregating the items.

In this experiment, only the frequent sequences are found. The performance of the algorithm is not considered. The performances of each algorithm are compared in next section by using the SPMF tool.

4.3 Sequential Pattern Mining with arbitrary dataset

For the experimental purpose, some arbitrary dataset with different properties is generated with the SPMF tool. The generated the dataset contains 10,000, 20,000, 30,000, 40,000 and 50,000 sequences with 1,000 distinct items with 30 items in a sequence and one item in a itemset. Using this type of dataset, some of the algorithms are implemented and compare with other in different conditions. The main difference here is a dataset has only one item in an itemset. In dataset that is implemented in this experiment, each item is ordered and not sorted. PrefixSpan, BIDE+, SPAM, CMSPAM, MaxSP, SPADE, CMSPADE, VMSP, FEAT algorithms are considered for the performance analysis with real and arbitrary data.

4.3.1 Sequential pattern mining with sequential database

We have selected some algorithms with the arbitrary dataset as describe in the previous section. Each dataset contains 1,000 distinct items, 30 items per sequence and one item per itemset. These algorithms are executed with five different datasets with the same number of distinct items and the same number of items in the sequences. For each dataset, these algorithms with minimum support 0.0002, 0.0004, 0.0006, 0.0008 and 0.001 are implemented and record time of execution, memory used and a number of sequences generated. Finally, we plot and analyze the results. At the end, we run the experimental setup using the real dataset. In all cases, the number of item in an itemset is always one. It means that only the most frequent sequences from the sequential database are generated. The items in the sequences are ordered but not sorted. For example, <3 4 2> and <4 2 3> are different sequences. But all the cases of above-proposed algorithm consider these sequences as the same

sequence. In above example, <4 2> has support 2 and there are no more sequences of length-2 with support more than 1.

4.3.2 Sequential pattern mining in sequential database with MaxGap

Consider a customer who went to the supermarket and bought bread and after 5 months he went again to the supermarket and bought butter. In this case, there will be no significance of considering <bread, cream> as a sequence. So, if there is a very long gap between items, then, we can prune that type of sequences from the candidate sequence or it can also be pruned at the time of candidate generation. Two of sequential pattern mining algorithms consider this type of constraints in their algorithm. Therefore, only two algorithms called VMSP (Vertical mining of Maximal Sequential Pattern) and CMSPAM (Co-occurrence Map Sequential Pattern mining with Bitmap representation) are considered that support the maximum gap between items. The working mechanism and pseudocode were already described in the previous section. Comparisons of these two algorithms with an arbitrary dataset that is generated with following properties.

Number of sequences = 50,000

Number of items in sequences = 30

Number of distinct items = 1000

Number of items in an itemset = 1

At the end, we verify the result with real dataset.

In each case, the result is recorded and presented in tabular as well as graphical form. The performance of the algorithm is compared in terms of execution time, maximum memory used, and the number of candidate generation.

4.4 Sequential Pattern mining with real dataset

All the above cases are validated using a real dataset from Xhockware. All the steps from pre-processing to visualize result (post-processing) is given below.

i. Choose sales_history and barcode_product_name table from the whole database.

ii. Select list of items in each transaction.

iii. Assign each item with an integer value which is an index value of that item from the barcode_product_name database.

iv. Apply sequential pattern mining algorithms which are implemented in the arbitrary dataset in previous sections.

v. Plot the result in different conditions to compare the performance of each algorithm with other.

vi. The result of step (iv) has integers instead of item's name, so, convert these integers with corresponding item name for the proper visualization.

Similar to the arbitrary dataset, three types of experiments are performed to compare the performance of algorithms with a real dataset.

In the first case, the dataset is just a sequential dataset. Each item is considered as itemset and we find the most frequent sequences. It does not contain the information of the time of shopping. We select only shortlisted sequential pattern mining algorithms. PrefixSpan, BIDE+, CMSPAM are the shortlisted algorithm for this type of dataset. Each basket is considered as a sequence. This type of implementation can be applied for the best path analysis, spot suggestion of item, sport discount offer while buying inside supermarket, etc.

In another case, the algorithms which considered MaxGap constraints are selected for the experiment. We select VMSP and CMSPAM algorithms with a real dataset. In this case, the dataset is a sequential dataset. Each item is considered as itemset. In this case also, we tabulate the execution time, maximum memory used and a number of candidate sequences and finally plotted the result in a line chart.

In the third experiment, we consider transactional database instead of a sequential database. Customer Id is considered as sequence id and represents dataset in transactional database format. All the transactions of a customer are supposed to be in a sequence and each basket of that customer is considered as an itemset of that sequence. Most important feature of this type of dataset is that itemset is ordered and not sorted.

4.5 SPADE Algorithm for ordered itemset

We modified the SPADE algorithm with sequential pattern mining for ordered itemset. Let us describe SPADE algorithm in detail before applying modification. The

sequential database is formatted in vertical format. It is better to take an example of it how it works. Let us consider the sequential dataset as shown in the following the table.

SID	sequence
10	<a(dac)(abc)d(dcf)>
20	<(cad)c(bca)(adeb)>
30	<(efb)(cdab)(dfa)cb>
40	<egd(caf)c(ba)d>

Table 4.2: Sequential Database with SID and Sequences

Let us implement SPADE algorithm to find sequential patterns from above sequential database. Above sequential database is converted into the vertical format as shown in the table below.

SID	EID	Itemset
10	1	a
10	2	dac
10	3	abc
10	4	d
10	5	dcf
20	1	cad
20	2	c
20	3	bca
20	4	adeb
30	1	efb
30	2	cdab
30	3	dfa
30	4	c
30	5	b
40	1	e
40	2	g
40	3	d
40	4	caf
40	5	c
40	6	ba
40	7	d

Table 4.3: Vertical Sequence of database

1. Find occurrence of each item with SID, EID format

a		b		c		d		e		f		g	
SID	EID	SID	EID	SID	EID	SID	EID	SID	EID	SID	EID	SID	EID
10	1	10	3	10	2	10	2	20	4	10	5	40	2
10	2	20	3	10	3	10	4	30	1	30	1		
10	3	20	4	10	5	10	5	40	2	30	3		
20	1	30	1	20	1	20	1			40	4		
20	3	30	2	20	2	20	4						
20	4	30	5	20	3	30	2						
30	2	40	6	30	2	30	3						
30	3			30	4	40	3						
40	4			40	4	40	7						
40	6			40	5								

Table 4.4: Vertical sequential database of each item

From the above table, the sequence of the item whose occurrence is less than minimum support is omitted. Suppose minimum support is 2, then the item g is removed from the list. So, frequent length-1 items are a, b, c, d, e and f.

Now, find the candidate sets from a single item. Finding candidate set is just combining items to each other. In this example, the combinations will be <aa>, <ab>, <(ab)>, <ac>, <(ac)>, <ad>, <(ad)>, <(ae)>…. To find its vertical table of occurrence join the table. For example, if we join a and b, we find all the occurrences with equal SID and check its EID. If EID of b is greater than EID of a with equal SID than that combination or sequence is accepted. For <(ab)>, both the SID and EID must be equal to be accepted. Let us find some sample combinations.

<ab>			<(ab)>			<(ac)>		
SID	EID(a)	EID(b)	SID	EID(a)	EID(b)	SID	EID(a)	EID(c)
10	1	3	10	3	3	10	2	2
20	1	3	20	3	3	10	3	3
30	2	5	30	2	2	20	1	1
			40	6	6	30	2	2
						40	4	4

Table 4.5: Length-2 sequences in vertical table format

We can make all the combination and find the support for each combination. Like in step 2, omit those sequences which have less support than minimum support. In this step, we can find all the length-2 frequent sequences.

In this way, we can generate sequences of length-3 from the length-2. In general, generate length-n sequences from length-(n-1) frequent sequences. Candidate generation process is similar to Apriori algorithm. Repeat these steps until the frequent sequences will be empty.

Modification of SPADE algorithm

The dataset for the original SPADE algorithm has unordered items in each itemset and it is sorted in ascending order. But in the proposed model, the items in itemset are ordered and not sorted. So, the proposed model can be applied for this type of dataset. From this model, we can extract two types of information; the sequential pattern of the overall transaction as well as frequent sequences from the itemset. In market basket analysis, we can use this type of information for path analysis, discount analysis, item arrangement, etc.

Let us take the same example from the above section. Consider the transaction database as shown in Table 4.2. The vertical sequential database for each item will be modified as shown below table.

a			b			c			d			e			f		
SI	EI	II	SI	EI	II	SI	EI	II	SI	EI	II	SI	EI	II	SI	EI	II
10	1	1	10	3	2	10	2	3	10	2	1	20	4	3	10	5	3
10	2	2	20	3	1	10	3	3	10	4	1	30	1	1	30	1	2
10	3	1	20	4	4	10	5	2	10	5	1	40	2	1	30	3	1
20	1	2	30	1	3	20	1	1	20	1	3				40	4	3
20	3	3	30	2	4	20	2	1	20	4	2						
20	4	1	30	5	1	20	3	2	30	2	2						
30	2	3	40	6	1	30	2	1	30	3	1						
30	3	3				30	4	1	40	3	1						
40	4	2				40	4	1	40	7	1						
40	6	2				40	5	1									

Table 4.6: Vertical sequence database of each item with SID, EID, and IID

Here, each item is represented by SID, EID, and IID, where IID indicates the index in the itemset. Now, we join each item with each other and make a table for length-2

sequences. Resulting table for length-2 sequences in above example will be as shown in the table below.

<ab>					<(ab)>				
SID	EID(a)	IID(a)	EID(b)	IID(b)	SID	EID(a)	IID(a)	EID(b)	IID(b)
10	1	1	3	2	10	3	1	3	2
20	1	2	3	1	20	3	3	3	1
30	2	3	5	1	30	2	3	2	4
					40	6	2	6	1

Table 4.7: Length-2 sequences in vertical table format in modification of SPADE

The formation of <ab> is same as an original algorithm but the formation of <(ab)> is the different from original SPADE algorithm. It prunes more candidates than original algorithm. In original SPADE algorithm, a sub-sequence occurs if the SID is equal and EID is in ascending order. Similarly, in the case of <(ab)>, both the SID and EID must be equal. But in our proposed model, in the case of <(ab)>, SID and EID must be equal and IID must be in ascending order. In above example, row 2^{nd} and 4^{th} are not considered as occurring sub-sequence.

Chapter Five: Experimental analysis

5.1 Clustering and Association Rule

The association rule and frequent sequence pattern depends on the shopping duration. We found that the association rule and frequent patterns is different for different cluster. The clustering was done with the features total time inside retailer house and total price.

As described in chapter four, we apply association rules in four types of item-set including all transactions and three clusters. We have 25,942 transaction records which are taken from only one retailer of LIDL, Cascais, Lisbon. We select the record only after November that means we analyze only for this winter. At first, the association rule (Apriori Algorithm) for all the transactions are implemented. From this experiment, the main finding is that the frequent itemsets are vegetables. The main fact behind this result is that, this item has only one brand and have only one barcode for one product. But other items like milk, rice, etc., may have many barcodes that depend on the company, therefore, these barcodes occur less frequently. The main limitation on the real dataset is that item is not categorized.

Let us now analyze each cluster with minimum support 0.03 and minimum confidence 0.1. The transaction or customer who spend more time inside retailer and spent more price are grouped into this cluster. Among 25,942 transactions, only 916 transactions are clustered in cluster one. The frequent items are more expensive items. Generally, clothes, cleaning items, kitchen items, etc. are more frequent items in this cluster. If a person buys a cloth, then he/she will buy the cleaning items like detergent. From this result, we can say that clothes and cleaning items can be arranged nearby. Another finding is that a customer bought clothing items will less likely buy grocery items; therefore, expensive items can be put nearby like clothes, items for decoration, cleaning items, etc.

To increase the profit for the supermarket, this result can be applied for the discount coupons. If a customer is in this cluster, discount coupons should be provided for other non-related items like vegetables, meat, and other glossary items because he will buy these related items without discount too.

In the second cluster, the customers buy random items but less expensive one like sneakers, candies, especial items for natal (Christmas), etc. This is a season of Christmas, therefore, the customer will buy gift items. If a person buys candies, then he will buy a watch or any other gift items.

In the third cluster, the most frequent item is bread, butter, etc. If a person buys bread, then he/she will buy butter. And if a person buys bread and butter then he/she will buy sausages. This cluster's clients spend less time in retailer house. So, the customer who stays less time in the supermarket will buy fast foods. It is a very interesting finding.

5.2 Clustering and sequential pattern mining

From the experiment of sequential pattern mining of transactional database, it is found that the sequences contain the same items. It means that if a person buys vegetable, then, most probably he/she will buy the same item in next visit. Besides this, some interesting frequent patterns are also generated. Let us describe some of the findings of each cluster in detail.

Cluster 1: This cluster contains only 913 transactions among 25812 transactions with maximum time spent inside the retailer. It gives 277 frequent sequential pattern rules and only 12 rules with confidence. In this cluster, most frequent sequences are vegetables and fruits. This cluster’s customer buys random items so it is very difficult to analyze.

Cluster 2: It contains 16,457 transactions with middle time spent inside retailer house. A total number of rules generated without confidence are 309. The result is a little bit different from cluster one. The most frequent items and sequences are Banana, Plastic bag, Butter, eggs, bread, water, etc. For example, if a person buys banana then he/she will buy a banana in future. If a customer bought bread, then he will buy butter in future and then eggs. With minimum confidence, it only produces 112 rules.

Cluster 3: It contains 8,442 transactions with minimum time spent inside the retailer. A total number of rules generated in this cluster are 1056 without confidence. Therefore, the customer in this cluster bought similar items frequently. The most frequent items are similar to cluster two. If the customer bought banana or fruits, then he/she will buy bread in future. With minimum confidence, it generates 596 rules.

The following table compares the three clusters according to the number of transactions, a number of items and execution time according to SPADE algorithm.

Cluster	Number of Transactions	Number of items	Execution time
1	913	3223	0.11s
2	16457	4128	0.18s
3	8442	5145	0.37s

Table 5.1: Number of transactions, Number of Items and Execution time for each cluster

From the above table and below figure, it concludes that the execution time in SPADE algorithm depends on a number of items, not in the number of the transaction. The number of transaction in the cluster is almost double than cluster three but the execution time is almost half of cluster three; it is because the number of items in cluster two is less.

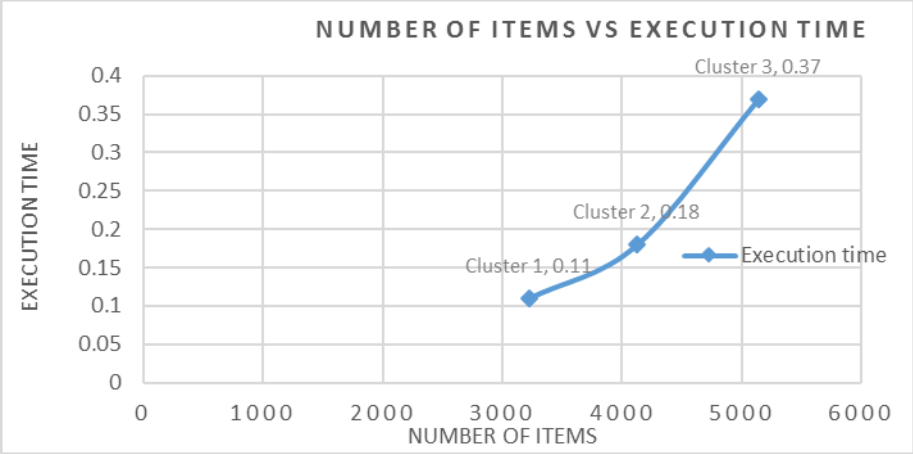

Figure 5.1: Number of Items and Execution time for each cluster

5.3 Sequential pattern mining with arbitrary dataset

In this experiment, we generate the dataset as described in chapter Methodology. Two types of the dataset are generated; sequential database and transactional database. In the case of a sequential database, each item is taken as itemset and find only frequent sequences from all the transaction. But in a transactional database, transaction id, event id and items are considered. Here, each itemset is a set of items in a basket.

5.3.1 **Sequential pattern mining with sequential database**

In this experiment, each itemset contains an item. After the experiment, each algorithm is compared with each other. we also compare algorithms with different values of sequences. Here, we calculate execution time, the number of frequent items, and maximum memory used for 50,000, 40,000, 30,000, 20,000 and 10,000 sequences in the sequential database. The execution time is measured in millisecond and maximum memory used in megabytes. Some findings are given in following table. Each result is tabulated in the table as well as draw in graphical charts. For each experiment, first, find the result and tabulate and then draw the result for proper visualization with line chart or graphical charts.

Let us find how execution time varied with variation in a number of sequence for PrefixSpan algorithm.

Min Sup/size	50,000	40,000	30,000	20,000	10,000
0.0002	5372	4781	3661	2800	1759
0.0004	904	774	686	654	683
0.0006	941	667	570	389	289
0.0008	873	724	579	367	222
0.001	893	731	577	372	218

Table 5.2: Execution time for different size databases in various minimum support

Let us draw a bar diagram for some algorithm how the execution time differs with different values of minimum support. We select PrefixSpan algorithm to draw the bar diagram with five different values of minimum support for five different datasets. The bar diagram is shown below. The discussion of each experiment is mentioned in next chapter.

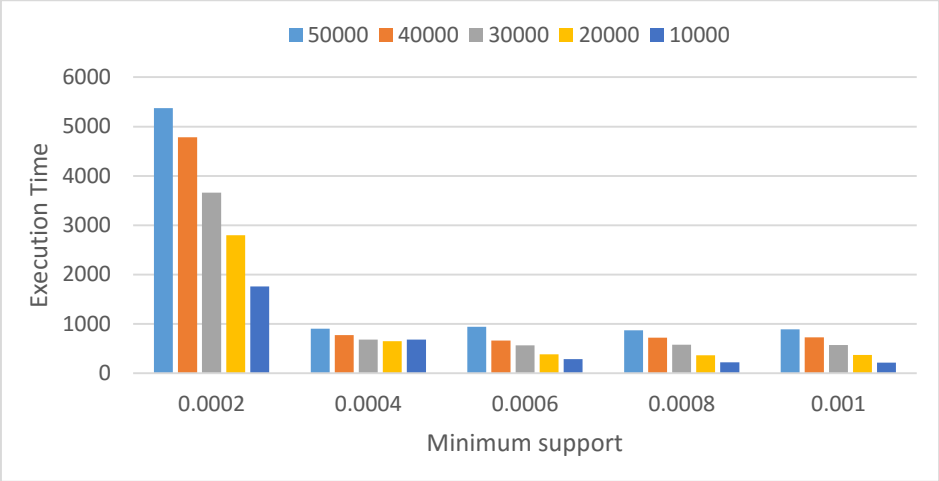

Figure 5.2: Bar diagram for execution time vs. minimum support for different dataset in PrefixSpan algorithm

Now, we compare seven algorithms in the dataset containing 50,000 sequences in the sequential database, 1000 distinct items, 20 items in sequences and one item per sequence. In some cases, the execution time is very high and seems to unbalance the diagram, therefore such values are omitted. For example, the execution time for SPADE and CM-SPADE is very high for minimum support 0.0002 and 0.0004 therefore these values are not considered in the table as well as in line chart. This case occurs in the case of less amount of minimum support. Like the previous experiment, we execute each algorithm with five different values of minimum support and their execution time is recorded in the table as shown in the table below. Finally, this tabular data is plotted in a line chart to have proper visualization. Execution time is plotted in different values of minimum support for each algorithm and plot as shown in Figure 5.3

Support	PrefixSpan	BIDE +	CMSPAM	MaxSP	SPADE	CMSPADE	VMSP
0.0002	5372	12163					
0.0004	904	981	3957	4629			4149
0.0006	941	874	3169	7136	15253	1602	3238
0.0008	873	863	3301	6781	3405	589	3379
0.001	893	886	3273	6400	807	485	3151

Table 5.3: Execution time vs. minimum support for different types of algorithms

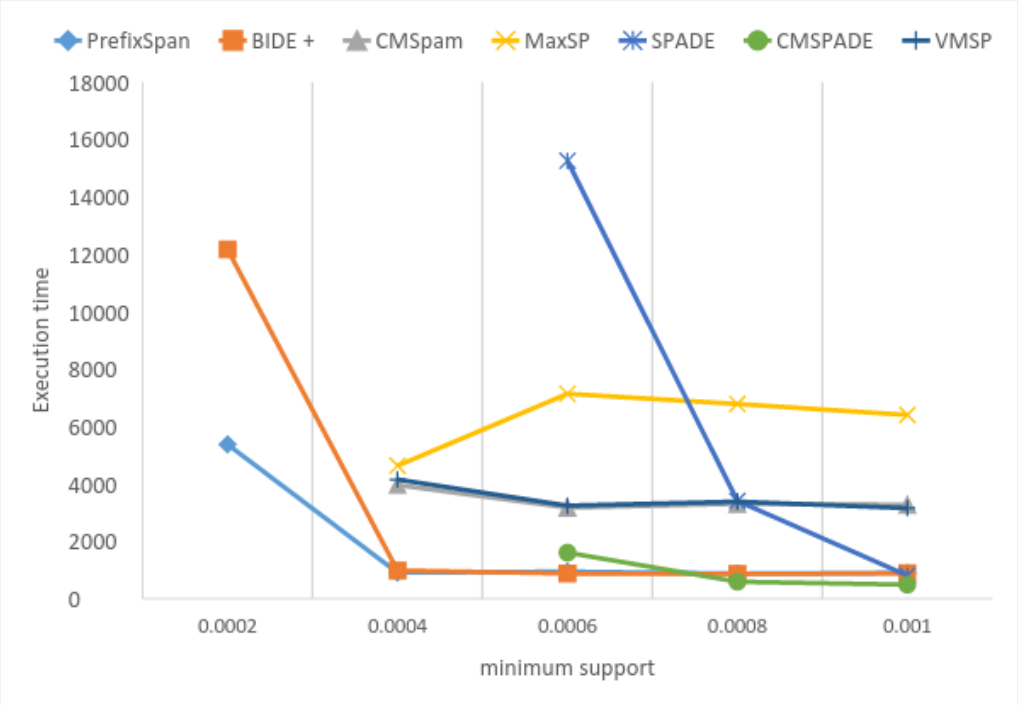

Figure 5.3: Execution time vs. minimum support for different types of algorithms

From above figure, it is found that each algorithm has different patterns. Among them, prefixscan and BIDE+ algorithm give similar patterns because they are working on the same principle. For the larger value of minimum support, CMSPAM is better. These three algorithms can be taken into consideration to be implemented in real data.

In the same way, we calculate a total number of candidate generation using different values of minimum support. The dataset is same as above experiment. The result does not seem to very interesting. In most of the cases, it calculates only the length-1 frequent sequences.

	PrefixSpan	BIDE +	CMSpam	MaxSP	SPADE	CMSPADE	VMSP
Support							
0.0004	2717	2717	2717	1772			1792
0.0006	1000	1000	1000	1000	13347	13347	1000
0.0008	1000	1000	1000	1000	1713	1713	1000
0.001	1000	1000	1000	1000	1030	1030	1000

Table 5.4: Comparing algorithms with different values of number of frequent sequence generation vs minimum support

Let us plot this result with line chart for better visualization.

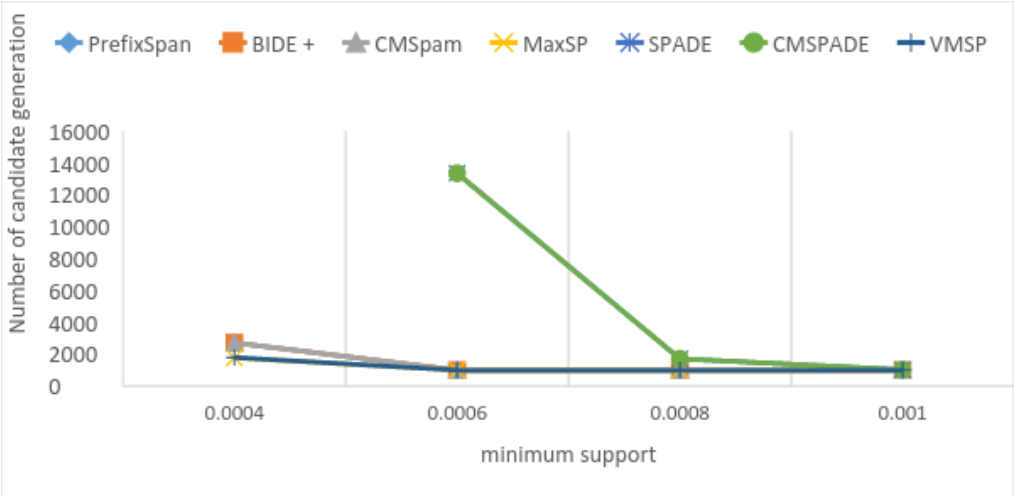

Figure 5.4: Comparing algorithms with different values of number of frequent sequence generation vs minimum support

In terms of a total number of candidate generation, the same algorithms are again the best performing. They produce less candidates than other algorithms. In this way, let us find some result for maximum memory used. All the properties of the dataset are same as in the above example.

	PrefixSpan	BIDE +	CMSpam	MaxSP	SPADE	CMSPADE	VMSP
Support							
0.0002	701.14943	1929.9		770.43			
0.0004	701.149	1929.94	1269.41	1556.86	1355.83	886.53	507.3
0.0006	701.149	1929.94	1154.57	1370.15	566.76	685.76	1070.4
0.0008	701.14	1929.97	1147.91	1154.184	804.77	631.81	948.79
0.001	701.149	1929.97	769.353	928.017	573.88	593.22	785.87

Table 5.5: Comparing algorithms with different values of for maximum memory used

The line chart for above table is given below.

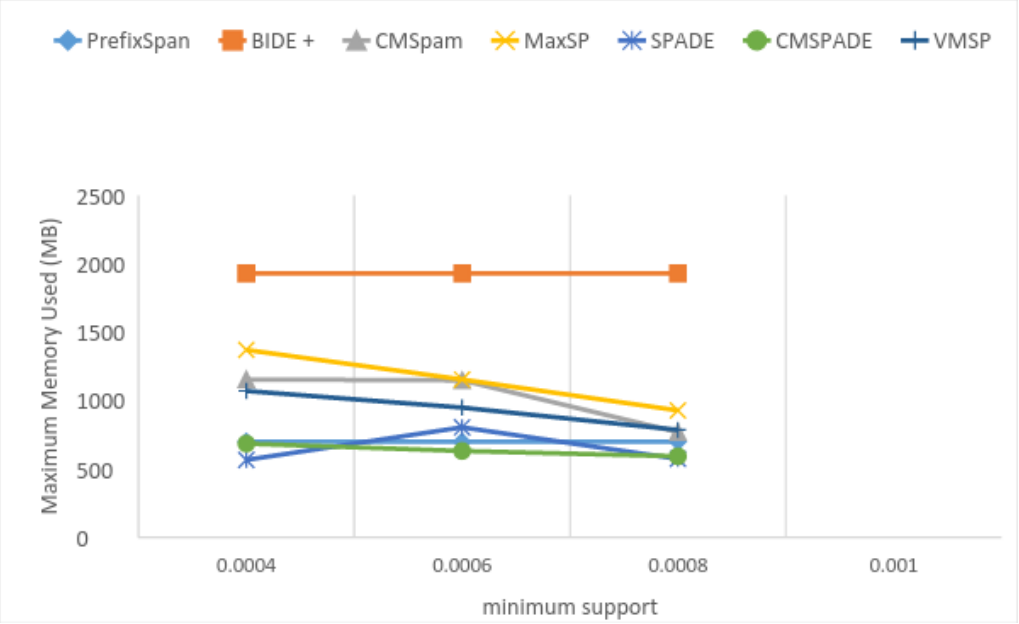

Figure 5.5: Comparing algorithms with different values of support vs maximum memory used

From above experiment, it is found that PrefixSpan, SPADE, and CMSPADE algorithm use less memory while generating frequent sequence patterns.

5.3.2 Sequential Pattern with MaxGap

We select only two algorithms for sequential pattern mining with MaxGap. The gap between items is an important factor to extract important information from the transaction database. The output with different algorithms are recorded in tables in different conditions. In this experiment, we select VMSP and CMSPAM algorithms and we calculate the execution time in millisecond, the number of candidate generation, and the maximum memory used in MB with five different values of the maximum gap between items and minimum support values. Like an earlier experiment, we recorded execution time, maximum memory used and a total number of candidate generation as shown in the figure below. Finally, we plotted the result in line chart for better visualization. In case the value is very high as compared to other values, that values are omitted to display properly in the line chart. If some values are very high, then the chart will not reflect the result properly. Tabular result and graphical results are shown below.

VMSP Algorithms

CMSPAM Algorithms

Execution time

Support	1	3	5	7	Inf
0.00006	120360	153423	252848	402962	
0.00008	110575	173068	262656	441289	
0.0001	20745	22680	22284	28020	27222
0.0003	5554	5164	6393	6244	6082

(a)

Execution time

Support	1	3	5	7	Inf
0.00006	134364	183077	249226	349083	1160160
0.00008	130732	183032	247984	344922	1098974
0.0001	22675	24713	25359	26381	29962
0.0003	5509	5559	6651	6202	5899

(b)

Memory

Support	1	3	5	7	Inf
0.00006	9879	80064	199782	352738	
0.00008	9879	80064	199782	352738	
0.0001	91	2332	9254	21816	245678
0.0003	5000	5000	5000	5000	5000

(c)

Memory

Support	1	3	5	7	Inf
0.00006	15255	88254	213123	373804	1813225
0.00008	15255	88254	213123	373804	1813225
0.0001	5091	7332	14254	26819	250693
0.0003	5000	5000	5000	5000	5000

(d)

Number of sequences

Support	1	3	5	7	Inf
0.00006	1617.25	1625.23	1716.46	1790.63	
0.00008	1637.7	1627.93	1711.3	1847.83	
0.0001	1612.19	1616.448	1617.306	1545.88	1662.13
0.0003	983.427	1164.668	1162.79	1038.5	1034.9

(e)

Number of Sequences

Support	1	3	5	7	Inf
0.00006	1619.96	1598.39	1613	1700.24	1855.169
0.00008	1663.89	1562.56	1656.99	373804	1861.363
0.0001	1613.89	1562.31	1608.26	1631.88	1631.25
0.0003	1124.459	1074.92	1021.83	991.86	1069.72

(f)

Table 5.6: Execution time, memory and frequent sequences for different values of MaxGap for VMSP and CMSPAN algorithm

With this tabular result, a line chart is drawn for the execution time, the maximum memory used, and the total number of the frequent sequence generated. The

execution time for different values of minimum support and MaxGap is plotted as shown in the figure below.

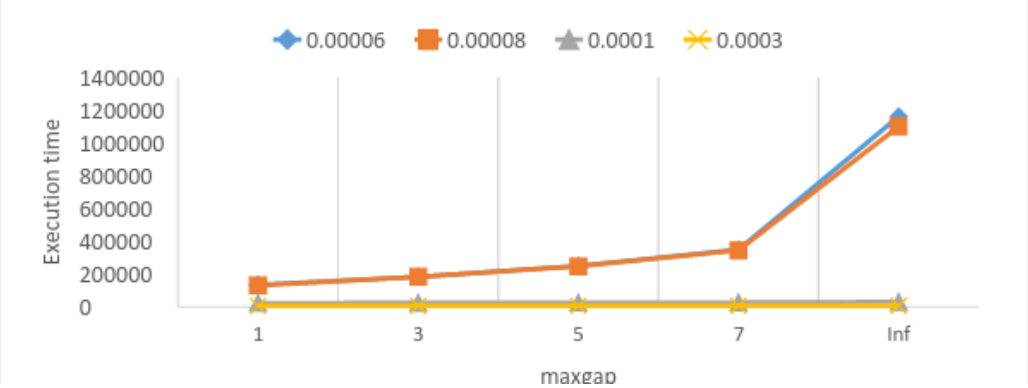

Figure 5.6: Execution time plot of CMSPAM with different Maxgap and minimum support

From above figure, we can say that the execution time is increasing with increase in MaxGap. If we do not consider MaxGap, then MaxGap value will be considered as infinity. It also found that, for the value of minimum support greater than 0.0001, execution time do not vary whatever the MaxGap value.

Similarly, let us find how execution time varies with the variation of MaxGap in VMSP algorithm. In this case, we consider different values of minimum support.

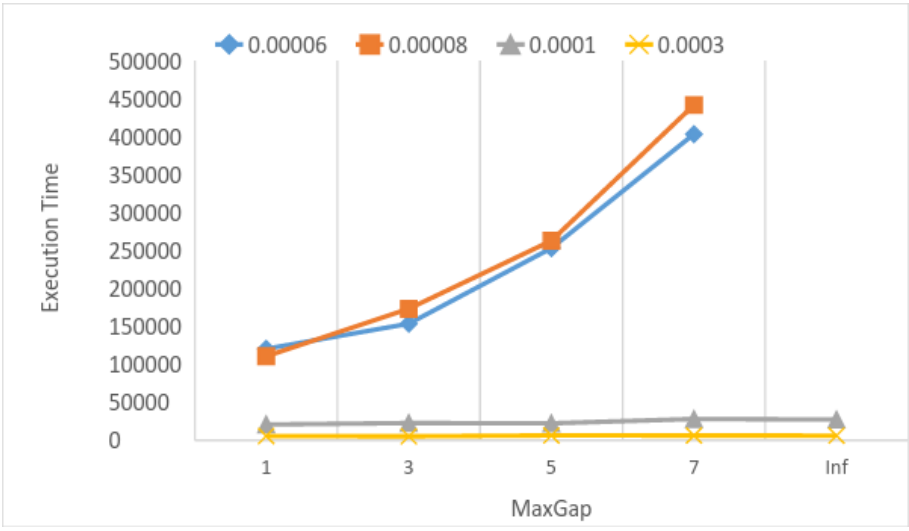

Figure 5.7: Execution time of VMSP algorithm with different values of Maxgap and minimum support

The pattern is almost similar in CMSPAM and VMSP algorithm, as expected since they both work on the same principle.

Let us compare both algorithms with minimum support value 0.0001. we use same tabular data to plot. We compare the performance of VMSP and CMSPAM algorithm in terms of the execution time with different values of MaxGap. We can see in the figure below, the performance of VMSP and CMSPAM do not have big difference in terms of the execution time. Therefore, to analyze the data with sequential pattern mining with MaxGap CMSPAM or VMSP both can be used.

In this way, we compare the maximum memory used for different values of MaxGap in both cases. In this case we again consider minimum support of 0.0001.

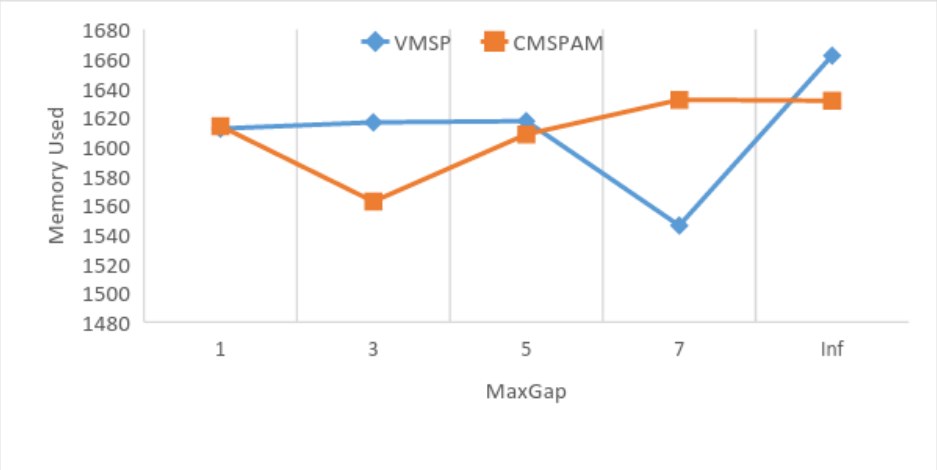

Figure 5.8: Comparing maximum memory used in different values of MaxGap for VMSP and CMSPAM algorithm

From above figure, there is no significant difference between both algorithms in terms of maximum memory used.

Finally, we can say that CMSPAM and VMSP do not have big differences in performance in all the conditions. In some cases, CMSPAM has better options for the application of constraints, therefore, in this thesis, most of time in later experiments, this algorithm is used.

5.4 Sequential pattern mining for real data

We compared the performance of seven sequential pattern mining algorithms with the arbitrary dataset in the previous section. Now, only three most relevant algorithms are selected with a real dataset of Xhockware. The dataset is converted to the format which is supported by SPMF application. This dataset also has two formats; sequential

database format and transactional database format. In each database format, experiment with and without MaxGap is performed and compare the result for selected algorithms.

5.4.1 Sequential pattern mining for transactional database

We choose only PrefixSpan, FEAT, and CMSPAM and compare them according to the execution time, the total number of frequent sequences generated and the maximum memory used. These algorithms use a transactional database where each item is represented as a number. The dataset contains all the transactions during the period indicated above. After the experiment, all the results are tabulated in the table and then in a graphical diagram.

Support	PrefixSpan	FEAT	CMSPAM
0.01	10302		9036
0.03	587		909
0.05	175	2983	537
0.07	113	1158	278
0.09	65	340	184

Table 5.7: PrefixSpan, FEAT, and CMSPAN algorithm with minimum support vs execution time

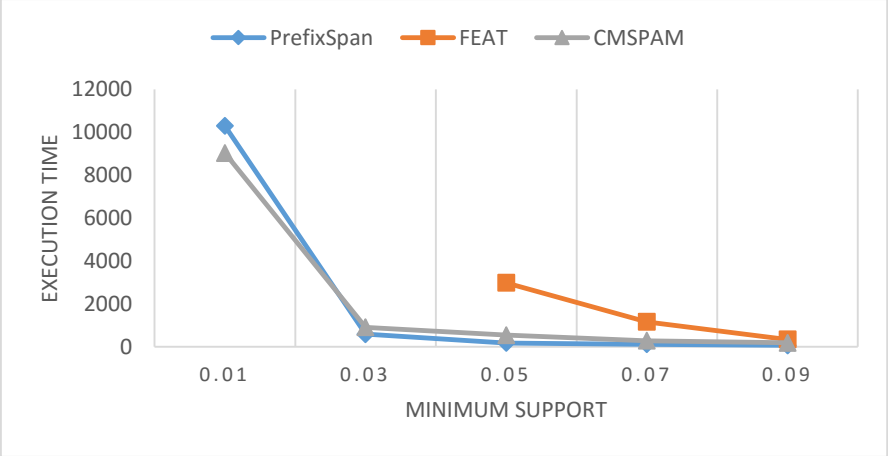

Figure 5.9: Comparing PrefixSpan, FEAT, and CMSPAN algorithm with execution time in different values of minimum support in real data of Xhockware

From above Figure 5.9, we can say that PrefixSpan and CMSPAM algorithm is the best algorithm in terms of the execution time. Both give a similar result. In this type of dataset, we can apply either PrefixSpan or CMSPAM algorithm to find frequent sequences from a transactional database.

In a similar way, we try to compare three algorithms in terms of the number of frequent items. Similar to the previous experiment, we represent the result in tabular form and then graphical form for better visualization

Support	PrefixSpan	FEAT	CMSPAM
0.03	1049		1048
0.05	251	254	251
0.07	119	121	119
0.09	64	67	64

Table 5.8: Comparing PrefixSpan, FEAT, and CMSPAM with number of sequence vs minimum support for real data

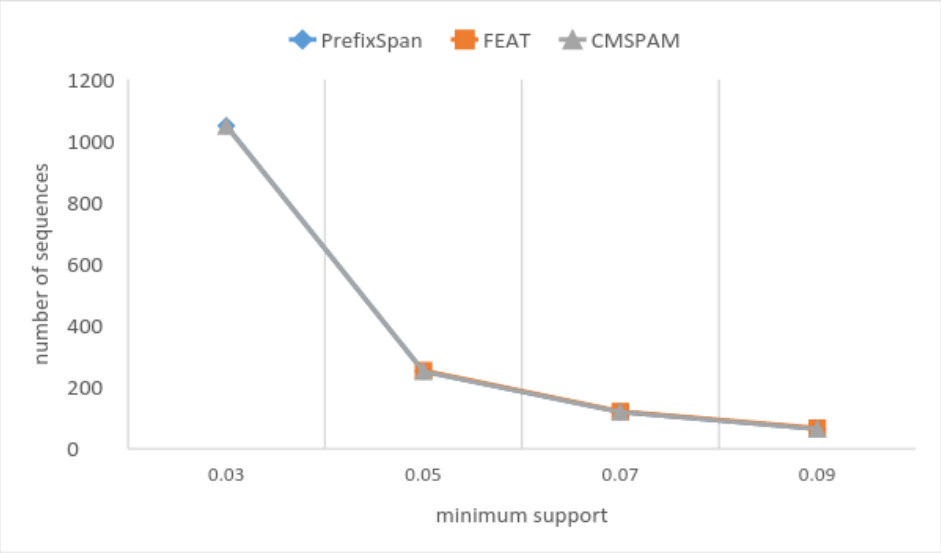

Figure 5.10: Comparing PrefixSpan, FEAT and CMSPAN algorithm for real data

From above figure, Number of frequent sequences generated is almost equal for all the algorithms in different values of minimum support.

5.4.2 Sequential Pattern Mining with sequential database

Here, the data is same as section 5.4.1 but the format is different. It is not the transactional database. It is only sequential database. Each transaction is taken as a sequence and we find the most frequent sequences of items. It contains total 25,925 sequences each sequence represents a basket. We choose only selected algorithms that are already verified from arbitrary dataset. Only PrefixSpan, BIDE+, and CMSPAM are taken for the experiment. The execution time for the sequential dataset

is tabulated as shown in the following the table. The minimum support values are 0.0001 to 0.0009 with five different values.

Support	PrefixSpan	BIDE +	CMSPAN
0.0001	631	2033	
0.0003	414	508	5351
0.0005	216	328	2253
0.0007	188	245	1247
0.0009	170	200	1054

Table 5.9: Comparing PrefixSpan, BIDE+, and CMSPAM with execution time vs minimum support

The line chart plotted for above tabular data is shown below.

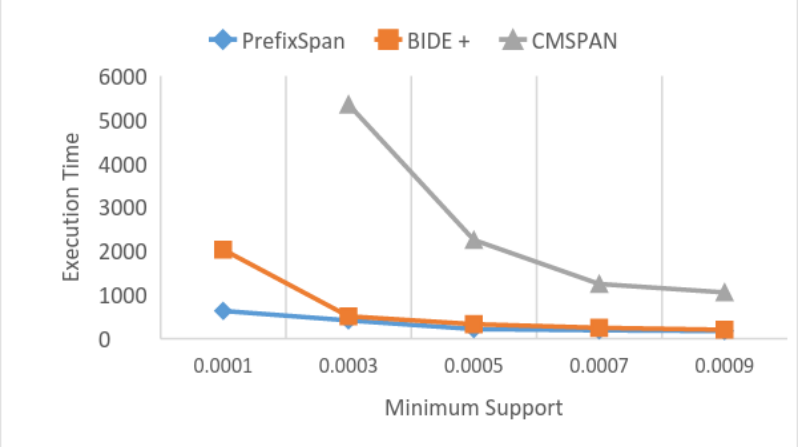

Figure 5.11: Comparing PrefixSpan, BIDE+ and CMSPAM with execution time

From above table and line chart, PrefixSpan algorithm is the best algorithm to find most frequent sequences from the sequential data set. But MaxGap is not considered in PrefixSpan algorithm.

Support	PrefixSpan	BIDE +	CMSPAN
0.0001	98267	96066	98267
0.0003	13596	13596	13596
0.0005	5606	5606	5606
0.0007	3030	3030	3030
0.0009	2155	2155	2155

Table 5.10: Comparing PrefixSpan, BIDE+, and CMSPAM with number of frequent sequences

With the same dataset, same conditions and same algorithms a total number of frequent sequences generated is recorded in tabular form and plotted in a line chart.

The line chart is drawn as shown in figure below.

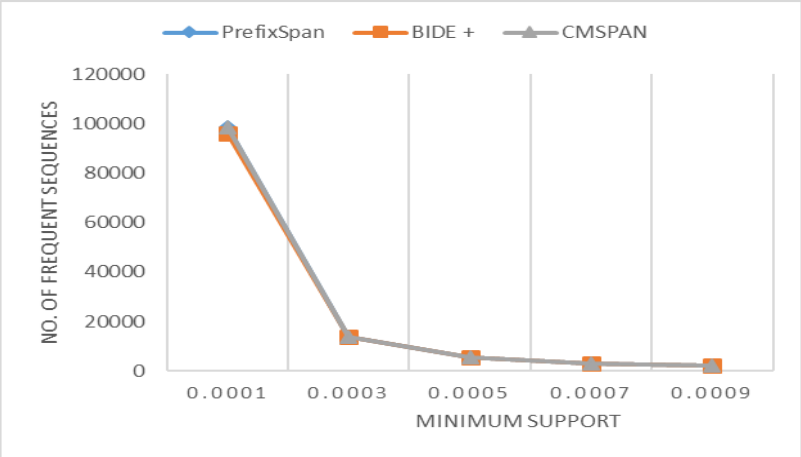

Figure 5.12: Comparing PrefixSpan, BIDE+, and CMSPAM with number of frequent sequences

The result is initially tabulated in a table and plotted in line chart for better visualization. From above table and line chart, it is shown that all the algorithm has the same number of pattern in each case with different values of minimum support.

Similarly, let us draw the result in term of the maximum memory used in the same case with the same dataset.

Support	PrefixSpan	BIDE +	CMSPAN
0.0001	277.78	677.58	1149.86
0.0003	429.19	677.58	1224.71
0.0005	429.19	677.58	1091.03
0.0007	437.27	684.77	946.79
0.0009	437.27	684.77	621.78

Table 5.11: Comparing PrefixSpan, BIDE+, and CMSPAM with Maximum Memory used

The result of above-tabulated data is drawn in following line chart. The maximum memory used in each case is different therefore they have different performance.

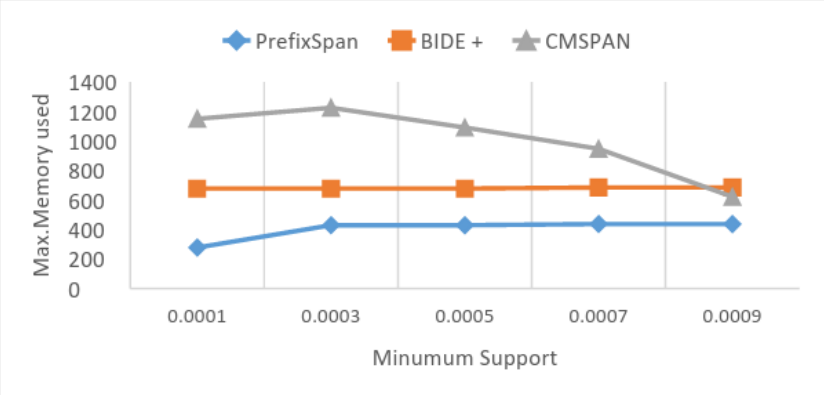

Figure 5.13: Comparing PrefixSpan, BIDE+, and CMSPAM with Maximum Memory used vs minimum support for real data

5.4.3 Comparing algorithms with different values of MaxGap for real dataset

Here, we select only VMSP, CMSPAM and VDEN algorithms. The dataset is only the sequential database in the same condition as above. The performance of each algorithm with respect to the execution time, the number of the generated sequences and the maximum memory used are evaluated. The minimum support value for each case is considered as 0.0005.

MaxGap	VMSP	CMSPAM	VGEN
1	1600	1663	1483
3	1715	1680	1671
5	1971	1702	1952
7	1884	1890	2034
Inf	2522	2106	2215

Table 5.12: Comparing VMSP, CMSPAM, and VDEN algorithm with different MaxGap

As in above experiment, the tabulate data is plotted in line chart as shown in the figure below.

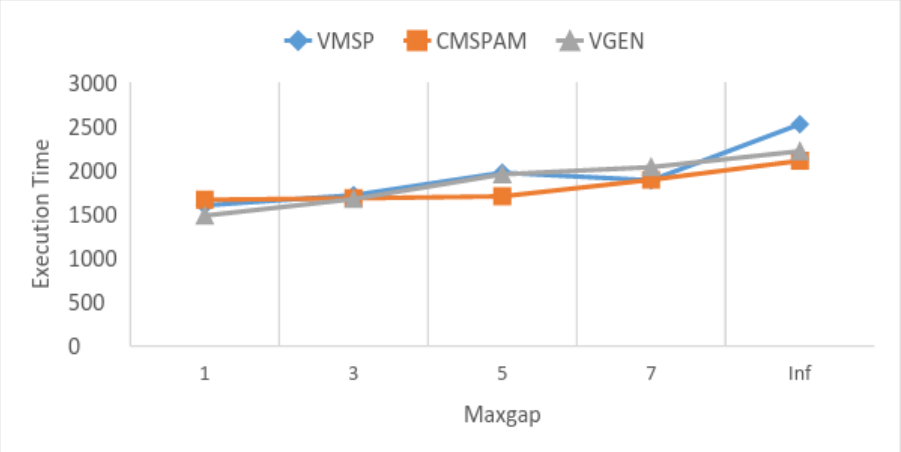

Figure 5.14: Comparing VMSP, CMSPAM and VDEN algorithm with different values of MaxGap factor

From above experiment, it is shown that all the algorithm has almost similar performance in terms of the execution time with different values of MaxGap values.

In a similar way, we tabulate a number of the frequent sequences generated in same dataset and same features. The minimum support is 0.0005 for each case. Like an earlier experiment, we tabulate the data and then draw line chart.

MaxGap	VMSP	CMSPAM	VGEN
1	1800	2081	2082
3	2292	2573	2574
5	2814	3099	3100
7	3316	3605	3606
Inf	5282	5606	5607

Table 5.13: Comparing VMSP, CMSPAM, and VDEN algorithm with Number of sequences

In this experiment, a number of candidate generation of each algorithm is almost similar in nature.

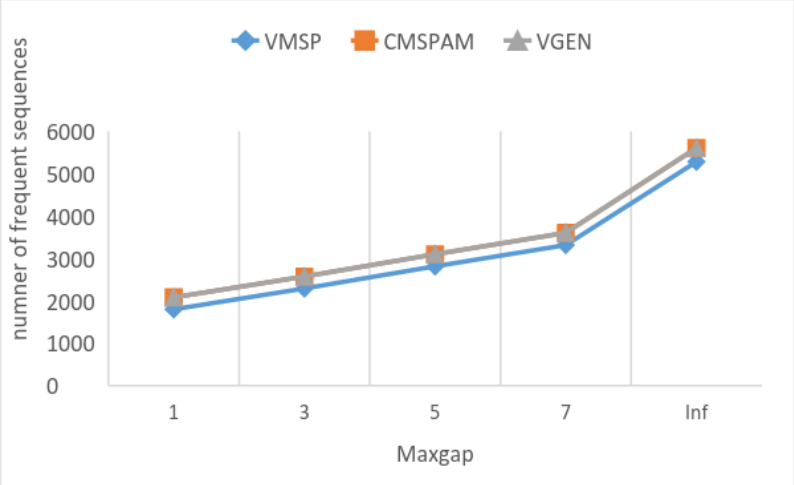

Figure 5.15: Comparing VMSP, CMSPAM and VDEN algorithm how they change in Number of candidate generation in different values of MaxGap factor

Now, let us compare performance in terms of maximum memory used. All the features and dataset is similar to above cases.

MaxGap	VMSP	CMSPAM	VGEN
1	955	900.32	1569.29
3	719.32	641.41	1278.86
5	739.69	641.99	954.07
7	722.66	722.99	756.05
Inf	787.53	755.03	869.37

Table 5.14: Comparing VMSP, CMSPAM, and VGEN with Maxgap vs minimum support

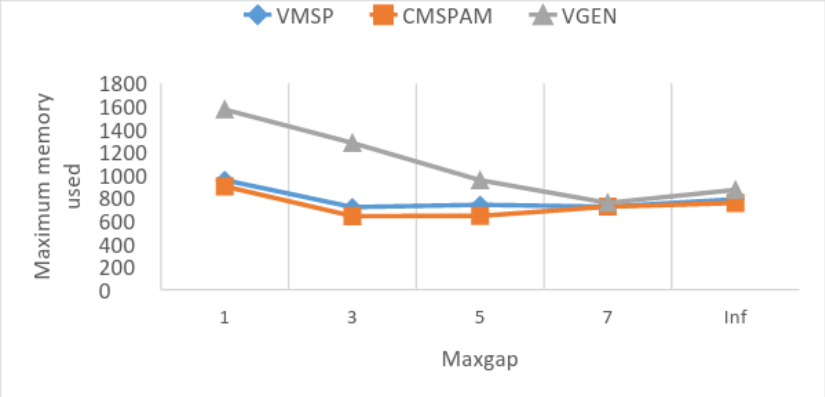

Figure 5.16: : Comparing VMSP, CMSPAM, and VDEN with maximum memory used vs Maxgap

In terms of the memory used, CMSPAM is the best algorithm as compared other two algorithms.

Let us analyze how the execution time varies with different values of minimum support with the same value of MaxGap. Here, we consider MaxGap value of 5.

Support	VMSP	CMSPAM	VGEN
0.0001	43583	27737	32703
0.0003	4182	3889	3952
0.0005	1971	1702	1952
0.0007	1393	1289	1119
0.0009	1163	1123	1275

Table 5.15: Comparing VMSP, CMSPAM, and VDEN algorithm with Execution time

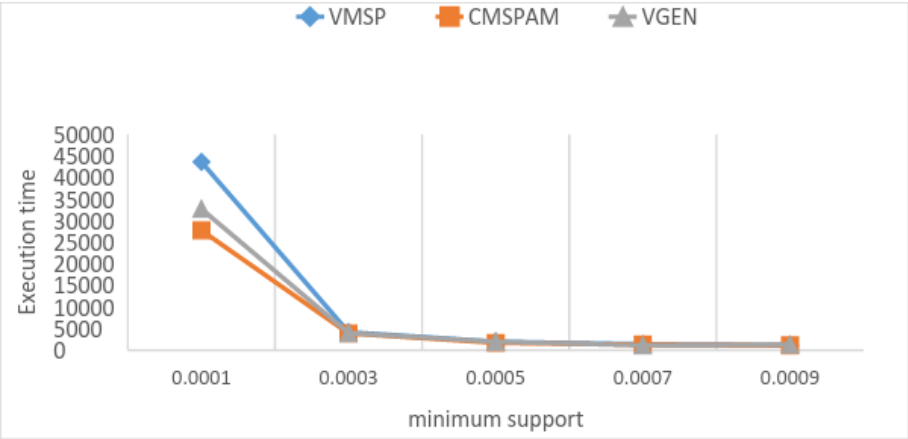

Figure 5.17: Comparing VMSP, CMSPAM and VDEN algorithm with Execution time in different values of minimum support Considering MaxGap value 5

5.4.4 Comparing CMSPAM algorithm with different values of minimum support and MaxGap

Here, we select only CMSPAM because it is only one algorithm which supports MaxGap factor with transaction database. The performance of this algorithm is also very good when compared with the other algorithms. Let us check performance with respect to execution time, a number of the sequence generated and maximum memory used. This experiment is a very important to find best value of MaxGap. The maximum gap between two item is a number of times a particular customer miss to buy that item. For example, if a customer buy butter today and he will buy bread after 5th visit, then the gap between this two item is 5.

Execution Time

Support	1	3	5	7	Inf
0.01	3999	5858	7848	9381	9869
0.03	887	901	947	955	1006
0.05	449	435	450	500	402
0.07	303	274	276	274	317
0.09	178	181	185	193	183

(a)

Frequent Sequence Count

Support	1	3	5	7	Inf
0.01	1973	5454	9438	13420	35126
0.03	370	476	591	955	1048
0.05	173	186	193	206	251
0.07	100	104	106	110	119
0.09	59	59	62	63	64

(b)

Max. Memory(MB)

Support	1	3	5	7	Inf
0.01	678.58	663.38	612.53	606.49	618.37
0.03	578.24	202.45	228.92	246.57	429.19
0.05	315.77	326.99	482.7	321.94	186.81
0.07	527.04	548.47	285.13	463.18	397.29
0.09	154.62	337.07	466.64	253.18	133.56

(c)

Table 5.16: Execution time, Number of frequent, and Max. memory with minimum support vs Maxgap

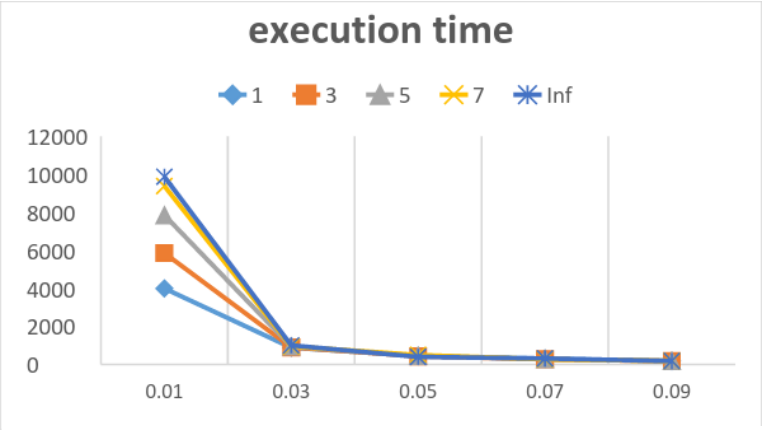

Figure 5.18: Execution time with different values of MaxGap and minimum support for CMSPAM algorithm

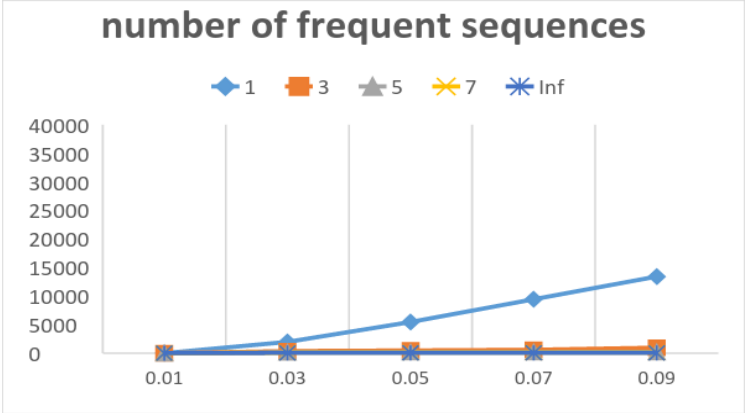

Figure 5.19: Number of frequent sequence count with different values of MaxGap and minimum support for CMSPAM algorithm

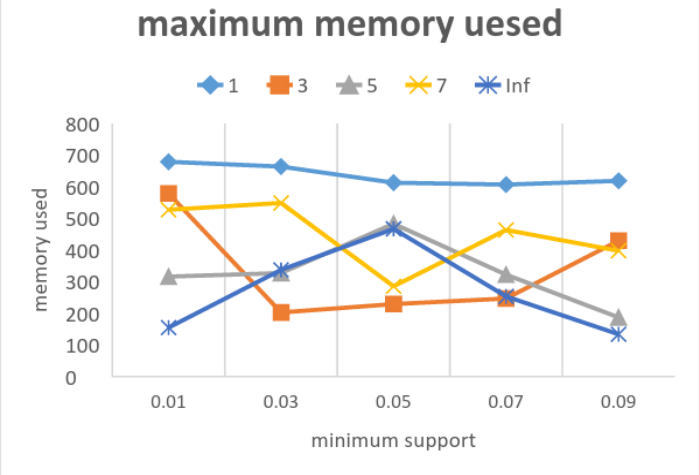

Figure 5.20: Maximum memory used with different values maxgap and minimum support for CMSPAM algorithms

From all the above three experiments, the minimum gap between item is three. To find the sequential patterns from the sequential database, the best value of maximum gap is three.

Chapter 6: Discussion and Future Work

6.1 General Discussion

Although we could not answer all the research questions, we try to answer some of them which are most relevant. From some experiments of clustering, and sequential pattern mining, we are able to answer some relevant queries from the business fields. The arrangement of items inside retailer house, offering the discount coupon for customer, prediction of next item from the sequence of the itemset, suggesting particular item while preparing shopping list, spot suggestion of items while buying the items are the implementation of findings of this research. Main machine learning task used are clustering, association rule, and sequential pattern mining. Mainly, we focus on sequential pattern mining.

6.2 Clustering and association rule

The dataset that we are analyzing have very important information which might contain some hidden knowledge. From the clustering and association rule mining, the important finding is that the cluster of customers or transactions who spent more time inside supermarket buy vegetables, fruits, meat, bread etc. most frequently. The cluster of medium time spent customers buy occasional items like candy, gift etc. and the cluster of customers with very high time spent customer buy expensive items like clothes. And another finding, the customer who spend maximum time inside retailer does not buy a higher number of the item as compared to the customer who spend medium time.

To increase the profit of the supermarket, this result can be applied in the discount coupons. Discount coupons for customer should be provided to other non-related or non-frequent items. For example, if a clustered customer buys vegetables, meat, and other grocery items then he/she will buy these related items without discount too therefore the discount can be given for other non-related items like cloths, cleaning items etc. Most of frequent items in supermarket are grocery items like bread, butter and vegetables. One of important finding is that if a person buys bread and butter then

he/she will buy sausages. The customer who spent less time in the supermarket will buy fast foods. It is very interesting finding.

6.3 Comparing algorithms for sequential pattern mining

Many researchers propose many algorithms in the field of sequential pattern mining and they also proof better performance with other. Almost all the sequential pattern mining algorithms are tested with their dataset. The dataset of Xhockware is totally different from the normal dataset. In their dataset, an itemset contains one or more than one items. But in real dataset, each itemset has only one item. Real itemset contains the sequences with ordered items and they are not sorted. Therefore, the performance may be different from their own dataset. We select seven algorithms to plot the performance in different graphical images. In overall algorithms, Prefixspan is the best algorithm for our real dataset. We compare these algorithms on the basis of the execution time, the number of frequent sequences generated and the total memory used. The result of sequential pattern mining can be used for different purpose. We can predict the next buying items for the particular customer, we can suggest/offer some product for the customer. We can apply this information for the best path analysis of the customer. We can arrange the items properly in the best path. We can put new items in the initial paths. By applying the profit on the frequent sequence, we can find the best path according to shopping behavior.

The experiment is done with two types of the dataset; one is sequential dataset and another is a transactional dataset. In the transactional dataset, only the frequent sequences of itemset from ordered itemset sequences are found. Some very interesting result are found in the sequential dataset. Generally, if a person buys the milk of one brand, then he/she will buy same items in next shopping. From this experiment, the customer normally does not switch the brand. Common grocery items like milk, bread, vegetables, fruits are common sequences appearing in many transactions. If a customer buys an egg, he/she willing to buy bread in next shopping.

The frequent sequences are found from the sequential database. The sequences are ordered and unsorted, therefore, it is different from the association rule. Frequent sequences in unordered and sorted sequences are similar to association rule. Of course, the initial items which is taken or scan is near to entrance but the next item may be different. Normally, the customer follows the same path while collecting the

items in the supermarket. The sequential pattern mining experiments show that the customer first picks butter and then pick bread. From this finding, we can advise supermarket to put some new items on this path. other frequent paths are cheese>meat>butter, bread>salad>sugar>, bread>egg.

6.4 Problem faced and its solution

While applying association rule on these data, we faced many problems and solve them as well. The main problem faced is handling big data. A table contains more than 12GB of data and contains more than two millions of records. To solve this problem, we used database management system for different types which are possible to do in database query because SQL (Structure Query Language) operations are faster than Matlab operation. For example, to replace barcode with item name, we used SQL query. Another problem is processing string variables. To solve this problem, we convert these variables to numeric variables, process it and finally again converted to a string.

To speed up the operation, all the pre-processing is done in PostgreSQL query and imported manually in the Matlab workspace. After Matlab operation, the result is stored in a Matlab variable and exported to database for the visualization of the result.

We use two methods to speed up the execution in Matlab.

1. Pre-allocation the variable: - The variable used in the loop may change the size of that variable in each iteration and need to copy in new location therefore before starting the loop, we allocate the tentative size of that variable and after loop ends, we again shrink that variable.

2. Use Parallel execution: We use par for loop which executes loops in parallel and gives the faster result.

3. We use numeric variables instead of using string variable: Association rule is generally applied in item name which is a string variable. If we process item name directly, the program will be more complex and also execute very slowly therefore we use barcodes for processing and finally, at the time of display the result is converted from barcode to item name.

Another problem is the language. The item name in the database are in the Portuguese language, therefore, it was difficult to understand for me. To solve this problem, we took the help from the Portuguese friends.

6.5 Future work

The supermarket is a very wide area where the machine learning techniques can be implemented. Beside supermarket, same techniques can be implemented in many fields. We can implement this dataset to classify customers according to their biographical information.

In this thesis we addressed the supermarket data analysis with sequential pattern mining techniques. Sequential data analysis is addressed in other fields like disease prediction, web page data analysis, genomics, etc. We can benefit from ideas developed within these fields, adapting them for our problem.

Our proposed model can be implemented not only in the supermarket but also in many areas as indicated in above paragraph. Particularly, in the supermarket for such a dataset that we have, we can use some other constraints to apply sequential pattern mining techniques and develop better algorithms with that constraints. For example, if we know the layout of the supermarket, then we can effectively find the best path and we can arrange items according to frequent items.

In this thesis, we only show with the help of one example that SPADE algorithm with ordered dataset gives only useful ordered sequences. But can compare the performance of this algorithm with other algorithms. We can also extend this method with some constraints like profit, Maxgap, etc.

Chapter 7: Conclusion

The main purpose was to use the newly generated dataset in machine learning. The real dataset that we used in our implementation section has totally new pattern and keeps new information. We used this information and able to extract new knowledge from that dataset. Clustering according to shopping time, finding frequent sequences in ordered and unsorted data set etc. are new findings in the supermarket sales dataset. The association of items depends on how much time spend inside the supermarket. It is not better to suggest the same item for all customer because the association of item depending on their shopping time. The result we draw from dataset is very different from another normal dataset. We found Prefixscan sequential pattern mining algorithms is the best one in terms of execution time, memory usage and number of frequent sequences found in the case of sequential pattern mining with ordered and unsorted itemset containing only one item in the item set in sequence. Our modified algorithm also achieves better results in terms of number of frequent sequence generation than the original method with the dataset containing ordered and unsorted real transaction dataset.

The performance of sequential pattern mining algorithms is highly dependent on the dataset they are applied. Important features of a sequential pattern mining dataset are the number of sequences, the number of items per sequences, the number of items in an itemset, etc. Therefore, before applying for a sequential pattern-mining algorithm, compare alternatives on your dataset to know which is best for you. To find the association rules do not consider all the baskets as the same category. Before applying association mining techniques categorize each basket and apply association rules so that the association of items depends on other shopping behavior.

Another important finding from this research is purposing new algorithms for the unique dataset. This research not only focuses on supermarket sales dataset but also on any fields like medicine image processing, the web, etc., which has this type of information. This search is just an opening research for this type of dataset so many types of knowledge can be extracted from this type of dataset.

References

Burke, (2007). Consumer choice of retail shopping aids. *Journal of Retailing and Consumer Services*, 339–346.

Chandni Naik, P. A. K., Niyanta Desai. (2014). Knowledge Discovery of Weighted RFM-Q Sequential Patterns from Customer Sequence Database*. Paper presented at the International Conference on Data Mining and Intelligent Computing (ICDMIC),* New Delhi, India.

Chueh, H.-E. (2010). Target oriented sequential patterns with time interval. *International journal of computer science & information Technology 2*(4), 113-123.

Cláudia Antunes, A. L. O. (2004). Sequential Pattern mining with approximated constraints*. Paper presented at the IADIS Applied Computing*, Lisbon, Portugal.

Gabor, M. E., &. F. (2010). Finding Sequential Patterns from Large Sequence Data. *International Journal of Computer Science, 7*(1), 43-46.

George Aloysius, D. B. (2012). An approach to products placement in supermarkets using PrefixSpan algorithm. *Journal of King Saud University - Computer and Information Sciences, 25*(1), 77–87.

Giannotti, F., Manco, G., Nanni, M., Pedreschi, D., & Turini, F. (1999). Integration of Deduction and Induction for Mining Supermarket Sales Data. *SEBD*, 117-131.

Guo-Cheng Lan, T.P. H., Vincent S. Tseng and Shyue-Liang Wang. (2014). Applying the maximum utility measure in high utility sequential pattern mining. *41*(11), 5071–5081.

Han, J. W., &. J. (2004). BIDE: efficient mining of frequent closed sequences. *Paper presented at the Data Engineering, 2004. Proceedings. 20th International Conference on.*

Jay Ayres, J. G., Tomi Yiu, and Jason Flannick. (2002). Sequential PAttern Mining using A Bitmap Representation*. Paper presented at the Proceeding KDD '02 Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM New York, NY, USA.

Jiaguo, L. (July 2010). A Solution based on Data Mining to Shelf Space Allocation*. Paper presented at the Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on (Volume:2),* Chengdu.

Jian Pei, J. H., B. Mortazavi-Asl, Jianyong Wang, H. Pinto, Qiming Chen, U. Dayal and Mei-Chun Hsu. (2004). Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach. *IEEE Transactions on Knowledge and Data Engineering, 16*(11), 1424 - 1440. doi:10.1109/TKDE.2004.77

Kirti S. Patil, S. S. P. (2013). Sequential Pattern Mining Using Apriori Algorithm & Frequent Pattern Tree Algorithm. *IOSR Journal of Engineering 3*(1), 26-30.

Kumar, D. L., C. A., M. C. & A. (2012). Market Basket Analysis for a Supermarket based on Frequent. *IJCSI International Journal of Computer Science Issues, 9*(5), 257-264.

Min, H. (2007). Developing the profiles of supermarket customers through data mining. *Service Industrial Journal 26(7):747-763, 26*(6), 747-763.

Morita, T. N.& H. (2006*).* Pattern Mining in POS Data using a Historical Tree*. Paper presented at the Sixth IEEE International Conference on Data Mining*, Hong Kong, China.

Philippe Fournier-Viger, A. G., M. C. &. R. T. (2014). Fast Vertical Mining of Sequential Patterns Using Co-occurrence Information. Tainan, Taiwan: *Springer International Publishing.*

Philippe Fournier-Viger, C.-W. W., Antonio Gomariz, Vincent S. Tseng. (2014). VMSP: Efficient Vertical Mining of Maximal Sequential Patterns *Advances in Artificial Intelligence* (Vol. 8436, pp. 83-94). Montréal, QC, Canada: Springer International Publishing.

Philippe Fournier-Viger, C.-W. W., Vincent S. Tseng. (2013). Mining Maximal Sequential Patterns without Candidate Maintenance *Advanced Data Mining and Applications* (Vol. 8346, pp. 169-180). Hangzhou, China: Springer Berlin Heidelberg.

Rakesh Agrawal, R. S. (1995). Mining Sequential Patterns*. Paper presented at the Proc. of the Int'l Conference on Data Engineering (ICDE),* Taipei.

Ramakrishnan Srikant, R. A. (1996). Mining Sequential Patterns-Generalizations and Performance Improvements *Advances in Database Technology—EDBT'96* (pp. 1-17): Springer Berlin Heidelberg.

S.O. Abdulsalam, K. S. A., A.G. Akintola and M.A. Hambali. (2014). *Data Mining in Market basket transaction: An Association rule mining approach.* Paper presented at the Foundation of Computer Science FCS, New York, USA.

Shivappa M Metagar, P. D. H., Anil S Naik. (May 2014). Case Study of Data Mining Models and Warehousing*. Paper presented at the International Journal of Innovative Research in Computer and Communication Engineering*, Bangalore, India.

SHOW-JANE YEN, J.-Y. G., & YUE-SHI LEE (2013). Mining Sequential Purchasing Behaviors from Customer Transaction Databases*. Paper presented at the 2013 IEEE International Conference on Systems*, Man, and Cybernetics, Manchester.

Wouter Buckinx, G. V. &. D. V. (2007). Predicting customer loyalty using the internal transactional database. *Expert Systems with Applications*, 125–134.

Ya-Han HuFan Wu, Y.-J. L. (2013). An efficient tree-based algorithm for mining sequential patterns with multiple

ZAKI, M. J. (2001). SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning 42*(1-2), 31-60.