Connected lighting system data analytics

Zhang, Y.

Award date:
2016

TECHNISCHE UNIVERSITEIT
EINDHOVEN
Computer Science and Engineering

Master Thesis
Connected Lighing System Data Analytics

Y.Zhang (0925713)
y.zhang@student.tue.nl

Supervisor:
Mykola Pechenizkiy
Dept. of CSE
Pierluigi Casale
Philips Lighting Research Eindhoven

September 8, 2016

Abstract

With the exponential increase of connected products and cheap digital data storage capabilities, applying data analysis and machine learning techniques on commercial datasets has been playing an important role to understand customer behavior. This understanding helps commercial decision-making in various aspects such as suitable time for product promoting or products placement in supermarkets.

Nowadays, Intelligent lighting systems generate large amount of data that include product information, usage patterns and timestamp, etc. Such data can be used to solve several interesting business problems.

In this study, three business questions are proposed based on business interests and data available, then answered by analyzing a sample dataset derived from real-world data of an connected lighting system. An ad-hoc data analysis framework is built for this aim. The framework is based on frequent pattern mining, which discovers frequent patterns in a given dataset. Three branches of frequent pattern mining: association rule mining, sequential pattern mining and frequent episodes mining are introduced to answer each question respectively. The generated patterns provide a clear understanding of the transactions happening in the data, which also provide a potential for decision making when using factual commercial and system data.

Disclaimer

The data and related results described in this thesis are generated from an anonymous dataset. The dataset is built for Scientific Research and Analytical explorations, it it not linked in any way to factual business and market facts. Conclusions derived in this Thesis are as such for the sake of Scientific Research, not for commercial or other employment.

Acknowledgment

Time flies, it has been almost a month since Ivan left; five months since I started internship; one year since the last time I came back from China; two years since the first time I landed Netherlands. It's just too fast. I am a happy person all the time, however not now, because I have to say goodbye to too many people whom I don't want to.

So many adorable people have appeared in my life during these two years, in university, in football pitch, In intern company, in parties and in PIC. Some of them gave me great help, and I really appreciate them. Some of them brought me happiness, which I will never forget. Some of them made me realize life can be such a beautiful thing.

Among those I want to say thank you, Mykola and Pierluigi are the people I first think of. Not being offensive, but I couldn't really understand what Mykola was talking in lectures during the first quartile in the first year, but he is a very responsible and careful tutor who helped our group a lot in his lecture. During the Master Thesis, he can always guide me to explore different ideas and techniques that I am not familiar with, however suitable for specific tasks. I just want to say thanks to him, and wish he has more time to take some rest. Pierluigi is another style, he helped me to get used to the new environment and makes my life at company a lot easier. I really enjoy having meeting with him (also with Tiblets), from the first time. Both me and Ivan find him really good at learning new knowledge and concluding other people's words. Apart from the instructing me about thesis and project, he taught me many useful skills of how to present my work to other people, which indeed help me. Another time, special thanks to both of them. Without them this thesis won't exists.

Emin, Jonas, Mikhail are my great team members and friends, we had great time together eating pizza at MF when heading to deadlines. Ting and Mengqi are the cutest roommates ever, I suppose they will never forget my scream at Efteling and Walibi. Xin is my 'boy friend' in many occasions, so sweet. Although I am not familiar with Adelize, she is so cute and crazy, can't help to like her. She made me know people can be so beautiful. I'm still missing the endless coffee break with Ivan, we are noob and nerd in each other's eyes. Can't forget the crazy ride in Frankfort with Dou, Dou Dou and Tiger. Last but not least, special thanks to Lv, thank you for making my life colorful, love you always. There are so many memories I want to share and so many people I want to say thank you, however I will stop here. Otherwise the acknowledgment will exceed the length of this thesis.

At last, thanks to my mom and dad.

Contents

List of Tables

List of Figures

1 Introduction

Since the advent of cheap computational power and massive data storage capabilities, companies have been interested in understanding customers' purchase behavior by analyzing customer transaction data. A good understanding of such behavior can help companies to address reasonable decisions which further improves customer purchase experience. For example, knowing the combinations of products that customers wish to have can help to prepare product bundles and save customers' time when choosing products. Furthermore, knowing the peak time of purchasing one kind of product can help to prepare promotion events for that specific product, enabling customers to know the products that best suit their expectation. These not only provide a better shopping experience for customers, but also enable higher profits for companies.

Pattern mining is an important topic in the data mining territory. Pattern mining can support the solution of many data mining problems, and it can also solve practical business problems. When pattern mining is used in basket analysis, this technique can help retailers to find customer purchase behavior, which may help to improve the customer oriented commercial strategy.

This thesis focuses on the practical problem of understanding the purchase behavior of consumers. Based on a sample dataset derived from real-word data of connected lighting systems, we aim to discover the purchase behavior of customers by leveraging data mining techniques. In Section 1.1, the business problem formulation of this work is present alongside 3 relevant questions of current interest in the business unit of the company. We translate these questions into scientific problems in Section 1.2. Then the rational of questions is stated in Section 1.3. Outline is provided in Section 1.4.

1.1 Business Problem Formulation

To infer and predict customer purchase behavior, historical record of products transactions needs to be used. By examining the transaction records, marketer and stakeholders are able to gain knowledge to increase customer satisfaction and make higher profits for the company. For example, repeat buyers identification can help marketers to recognize customers who continuously purchase products from a certain retailer and help them in their next purchase. On the other hand, knowing these customers with their behavior can help to preserve repeat buyers and scout for new such customers. In this study, we address and solve three business problems of interested listed in the following:

1. What types of products are frequently purchased together?
2. Are there any patterns reflecting purchase order of products? If there are, how to use the order to help decision making?

3. Are there any patterns reflecting time span between purchase behavior? If there are, how can this help to select proper products to promote within a given time window?

The study conducted in this thesis helps us to have a deep understanding of the customer purchase behavior from the transaction records, which will further provide a solid basis for decision making. In the following, we will explain how to translate these questions into a scientific problem. Afterwards, based on the business problem formulation and scientific problem formulation, rational of the questions are discussed.

1.2 Scientific Problem Formulation

Given the proposed business questions, the scientific problem is formulated as following

- Given a dataset of transactions, we want to discover frequent patterns that arise from the data. Proper techniques in frequent pattern mining territory shall be found to address each specific business question. The results of such techniques shall be well organized such that suit the expected answer of business questions.

The field of frequent pattern mining has been thoroughly studied during the last two decades and it contains problems from different perspectives (efficiently generating frequent itemsets [Agrawal dan Srikant, 1995], [Borgelt, 2003], multilevel frequent pattern mining [Han dan Fu, 1999], constrained based frequent pattern mining [Pei, *et al.*, 2001], etc.). We find some frequent pattern mining techniques are suitable to address the specific business questions proposed. For the first business question, frequently purchased product combinations are expected results. Such results are in the similar form as the results of association rule mining, where the results shall contain item (product) combinations that frequently appear. In addition, association rule mining takes care of conditional probability between items (products). For the second business question, ordered product sequences that appear frequently in customer transaction records are expected results. Such results are in the similar form as the result of sequential pattern mining, which can mine time ordered event sequences that appear frequently. For the third business question, frequently purchased product combinations within a certain time window are expected results. Such results are in the similar form as the result of frequent episodes mining, which can mine event combinations that appear frequently within a certain time window. Based on the discussion above, we state that:

- Association rule mining is a suitable technique to solve business question 1
- Sequential pattern mining is a suitable technique to solve business question 2
- Frequent episodes mining is a suitable technique to solve business question 3

These techniques have been integrated to work together as a coherent data analysis framework that will support the decision-making process of business unit. The data analysis framework has been developed based on frequent pattern mining, which is the fundamental problem in pattern

mining territory. Many studies contribute to this topic from different perspectives, such as frequent itemset discovery [Agrawal dan Srikant, 1995], frequent itemset generation without candidate generation [Han, *et al.*, 2000], multi-dimensional frequent sequence mining [Pinto, *et al.*, 2001], etc. In general, the problem is to find frequent itemsets, sequences that satisfy a preset minimum support, which are supposed to be interesting to business unit.

In different contexts, requirements of pattern mining are different. For example, in [Aloysius dan Binu, 2013], frequent pattern mining is used to find proper product placement in supermarkets. Thus mining frequent itemsets is enough to solve the problem. In other context, temporal information shall be considered. For instance, a food retailer may be interested in the question: In October, what kind of potato chips are frequently bought by customers. In this question, apart from itemsets (potato chips), temporal information (October) also attracts retailer. Temporal information is additional attribute in a transaction that gives more insight into customer behaviors[1].

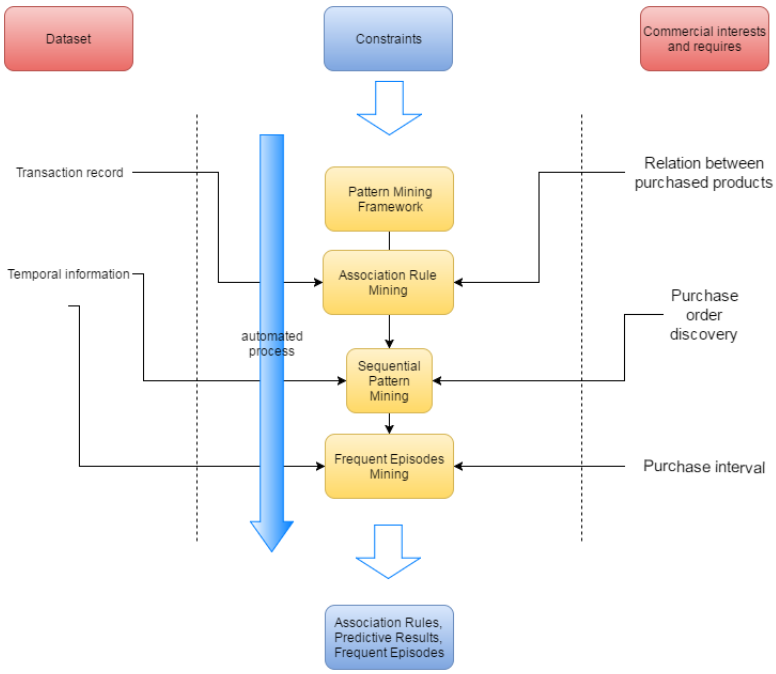


Figure 1: Data analysis frame work

We want to provide data driven support for decision making. Therefore, to build a framework according to this specific dataset is very important. The sample dataset contains products being used by customer, the time of purchase, customer usage data, etc. As shown before, the food

[1]Note that additional attributes are not restricted to temporal information, for example location, personal profile, etc.

retailer example, temporal information is considered in pattern mining. In our case, we also consider temporal pattern mining as a good method to solve our problem. Two branches of temporal pattern mining are used to discover temporal information: frequent sequential pattern mining and frequent episodes mining, which are discussed in Section 4 and Section 5 respectively. In addition, because of the sparsity of product's dimensions and the nature of products in our sample dataset, multilevel pattern mining is applied. With different level of products, users can analyze dataset from different point of views: from single products to groups of similar products. In the end, statistical evaluation is applied to check whether the generated rules are interesting w.r.t. correlations of itemsets, which is an important step to eliminate rules that are not interesting.

In general, we regard our problem as a frequent pattern mining problem, as shown in Figure 1. Given the input from users, we resolve the input as three different, but related questions to solve, association rule mining, sequential pattern mining and frequent episodes mining respectively. These three pattern mining problems can answer corresponding business questions from different perspectives. After above three process, we evaluate the results of each phase and provide output at last. Our work mainly focus on finding proper existing techniques and algorithms to solve our problem, and integrate these techniques in our data analysis framework.

1.3 Rationale of Questions

As mentioned above, three pattern mining techniques will be used for each business question. In this section, based on the business problem formulation and scientific formulation, we explain in detail why these business questions are proposed. The business questions are proposed based on business motivation, scientific techniques and the data available, where the first two are already discussed. As for the third, pattern mining techniques need certain features from data to generate results. Table 1 shows the data features needed for each pattern mining technique. This table also shows given data features and pattern mining techniques, what kind of business questions can be answered as well as the expect result. Each business question is detailed as follows:

Table 1: Business Questions according to data features

Data Features	**Pattern Mining Techniques**	**Business Questions**	**Expect Result**
Purchased products	Association rule mining	What kind of products are frequently purchased together?	Frequently purchased products combination
Purchased products & Purchase time	Sequential pattern mining	Are there any patterns reflecting purchase order of products? If there are, how to use the order to help decision making?	Frequently purchased products combination with order
Purchased products & Purchase time	Frequent episodes mining	Are there any patterns reflecting time span between purchase behavior? If there are, how can this help to select proper products to promote within a given time window?	Frequently purchased combination of products with order, within certain time window

What kind of products are frequently purchased together?: Given an example product set $\{B, C, A, D\}$, considering product bundles, companies are interested in which kind of bundles (e.g. $\{B, A\}, \{B, C\}, \{C, D\}, \{A, C, D\}$) cater customers' favor best, because knowing the favor of customers can help the company to emphasize promoting those bundles which are most likely to be purchased. Since the customer purchase record actually reflects the favor of customers, we can mine the history purchase record to find the potentially popular bundles. Therefore, to find the frequently purchased products combination from history purchase record is useful when trying to figure out popular products bundle. As shown in Figure 1, to answer this question, only the purchased products are needed. The expected results are frequently purchased product combinations, which are able to answer the first business question.

What is the purchase order of frequently purchased products?: Given a frequent purchased product set $\{B, C, A, D\}$, companies are interested in what products are likely to be purchased at the front, what products are likely to be purchased at the end, namely the order of purchase. Therefore, knowing the purchase behavior w.r.t order is useful when doing consecutive products promotion. As shown in Figure 1, purchased products and corresponding time of each purchase is needed to answer this question, and the expect result: Frequent purchased products combination with order are able to answer this question.

What kind of products are frequently purchased in order given a specific time window? Given an ordered frequently purchased sequence $\{A, B, C, D\}$ from a transaction dataset and a customer who just bought A, company may want to know when is the proper time to recommend product B to this customer. Therefore, to find a time window which can effectively restrict promotion time span is of great importance. As shown in Figure 1, purchased products and corresponding time of each purchase is needed to answer this question; the anticipate result: Frequently purchased combination of products with order, within certain time window (frequent episodes) are able to answer this question.

These three topics mentioned above are the business questions we would like to answer. In the following sections 3, 4, 5, each question is answered in detail. The reason that we come to these three business problem and the reason we exclude other possible business questions are explained as follows:

Normally, the ultimate goal to apply data analysis on product usage dataset is to understand the behavior of customers and provide better user experience. For example, Cao, *et al.* [2008] analyzed user click stream data gathered from a commercial search engine to provide context aware query suggestion. In our case, by analyzing customer usage data, we might be able to predict how customers use connected lighting system. For example, to analyze the brightness setting of certain lights may help to set the default brightness for different groups of customers, which make customers comfortable when they first use the lights. Therefore, to leverage customer usage data

to improve user experience of a system is worth investigate. Moreover, in our sample dataset, customer usage data is available to help conduct such research. However, a very big issue is that our usage data is not consecutive. In another word, for each customer, their information is recorded weekly, which make it hard to analyze user behavior. For example, a possible question to answer is when customers turn the light to the brightest mode. For this question, temporal data that granularity at least in hour is needed. Another problem for user experience improvement study is even tricky. Even if we have got a convincing result from our data, it is hard for us to evaluate the result. First of all, we need to collaborate with product designer to adjust products settings (e.g. default brightness) based on our analysis. This change shall be well validated and tested to make sure most customers like it, which will take a long time. After users have used the newly configured products for a while, feedback is requested from customers. The problems is that it takes too long a period to design, release, customer purchase, customer use, gather feedback. Based on the discussion above, we decide not to conduct study on user experience improvement.

Another possibility is to focus on products promotion. Promoting proper products to customer is an important task in market analysis. Namely the recommendation system. The performance of the recommendation system directly influence the purchase behavior of customer. Many companies have already established or have been improving their recommendation systems in recent years [Bennett dan Lanning, 2007][Linden, *et al.*, 2003].

Therefore, we want to find proper products and time for products promotion. Previous studies provide various ways to build recommendation system for product promotion. For example, [Linden, *et al.*, 2003] use collaborative filtering, which use information from many other customers to predict a certain customer's favor. Same as the example mentioned above, many recommendation systems do predictive analysis based on various information, including user profiles, opinion on specific products, purchase history, etc. However, in our sample dataset, apart from customer usage data and product information, no other information available. This makes hard to apply such product promotion methods.

However, another popular branch: pattern mining is suitable for our problem. As described in Introduction, this technique is firstly brought by Agrawal, *et al.* [1993] which aims to find valuable product relation from a transaction database. Later on, pattern mining evolves to satisfy many other practical applications [Wright, *et al.*, 2015],[Ang, *et al.*, 2013]. We find part of our dataset is very similar to the transaction dataset in [Agrawal, *et al.*, 1993], which enables us to use this technique. As described in Section 2, for every system/customer (these two words are interchangeably used), we have its full configuration file. We regard a new plugin of products as a transaction, where time of transaction is recorded weekly. Since people don't buy expensive products[2] as too often, thus using week as time granularity will not lose too much temporal information. With information of transaction and corresponding time, we are good to go.

[2]Products in this connected lighting system are not cheap

1.4 Outline

The rest content is organized as follows: We first introduce the sample dataset that we are working on in Section 2. In Section 3, we introduce the basis of frequent pattern mining, then we introduce (multilevel) frequent pattern mining to answer the business question: what products are frequently purchased together by customers. Interest measures of association rules are also introduced. Next, in Section 4, predictive analysis by using sequential pattern mining is detailed to answer the second business question. In Section 5, the concept and definition of frequent episodes mining are introduced, and third business question about the interval between customer purchase is detailed. The study is concluded in Section 6, future work is also mentioned here.

2 The Sample Dataset

The sample dataset is created using an extract of the logs of connected lighting system configuration files. In total, more than four thousand independent systems are recorded in this dataset, in which every record spans a whole year. Different products of the system may appear in the dataset. Third-parties compatible products may also appear in the dataset. In each record, features providing information of the system are present. For example, week number is recorded as temporal information, number and type of products are detailed, and products usage information are recorded as well.The records in this sample dataset are anonymous, which means no user profile can be identified or analyzed.

Time is discretized in week, which implies that the purchase of new products is recorded weekly. The company doesn't allow us to know the exact time of purchase of new products, because the privacy of users need to be protected. Moreover, usage information such as saturation, on/off and brightness are also recorded weekly.

The sample dataset results to be very sparse in terms of dimension and time. Frequent pattern mining, sequential pattern mining and frequent episodes mining are usually applied to dense dataset such as mailing campaign[Wong, *et al.*, 2005], web usage data [Mobasher, *et al.*, 2001], inaugural address and novel [Tatti dan Cule, 2012]. Although many products may appear in the dataset, customers don't buy all of them. Many customers just buy few types of products, some even buy only one type of product. This behavior results in the dimension sparsity found in the data. Products purchase data are also sparse in terms of time since customers may not buy products on a weekly or even monthly base. Many customers might just buy once. Because of the sparsity of time, we modify the input data for association rule mining as we will explain in Section 3.2. Because of the sparsity of dimension, we introduce multilevel association rules to reduce dimension. This will be detailed in Section 4.

Based on the information publicly available on Internet, there exists an explicit taxonomy of connected lighting system products as shown in Figure 2. A 3-level hierarchy explicitly shows how products [3] are grouped together. The products reside at the leaf level of this tree-like hierarchy. For example, *ColorS1, Ewhite1, white2* resides in the leaf level. The middle level consists of different groups of similar products based on products' attributes or special usage. For example, the group *ColorS* contains both *ColorS1, ColorS2*, because they both belong to a special type of products. The highest abstraction level describes the functionality that the light supports. For example, *white*, which supports groups, scenes, on/off and dimming. This *is-a* relationship is used as the basis for multilevel association rules mining.

[3] 'item' and 'product' both mean the products in this dataset

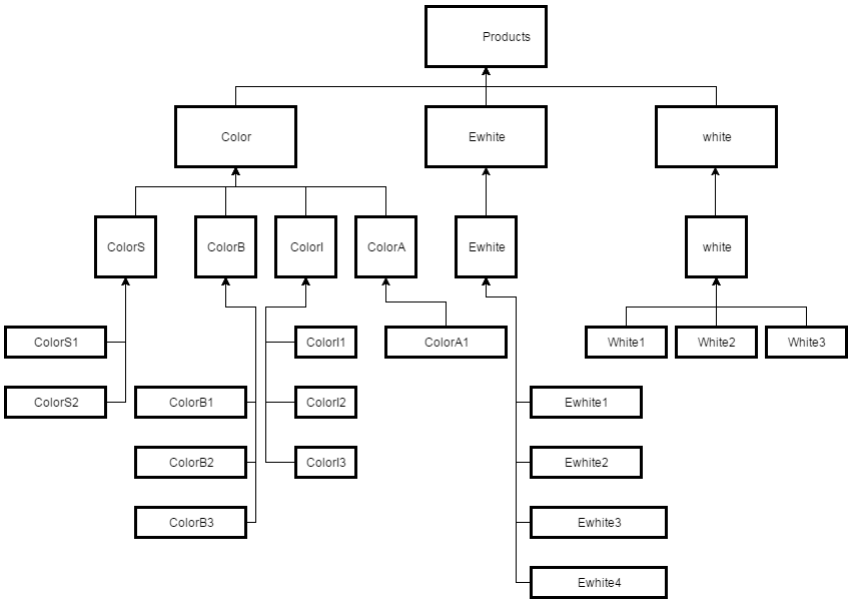


Figure 2: *is-a* relationship in sample dataset

Three levels are presented in this *is-a* relationship (We do not count root node as a level). Leaf-level represents products, middle level represents different groups of similar products, and the high level describes the functionalities that a group of lights support.

As shown above, many attributes are worth investigating in this dataset. For example, we might use customer usage data to improve user experience for certain lights or certain groups of customers. In addition, and temporal information may help to decide when to promote certain types of products. With such informative a dataset, many research topics can be derived. In this work, we investigate the business questions satisfy the company's needs, as previously described.

3 What types of connected lighting system products are frequently purchased together?

Finding frequently purchased products combination is of great interest to company. This helps to recommend products bundle based on customers' favor. In order to find such combinations, we apply Association Rule (AR) mining on our sample dataset. In addition, multilevel frequent pattern mining is introduced for two reasons:

1. Connected lighting system products have inherent taxonomy, which is worth investigating.
2. The sample dataset is sparse in terms of products dimension, thus the dimensions need to be reduced.

Moreover, not all generated results are interesting for the company. To eliminate uninteresting results, we apply different interesting measures and select proper ones to answer our question.

3.1 Association Rule Mining

In this section, we first introduce the concept and definition of association rule mining. The problem is formally defined as follows:

Let $\mathcal{T} = \{I_1, I_2, ..., I_m\}$ be an itemset. Let $\mathcal{D}$ be a set of transactions, for each transaction T in the dataset $\mathcal{D}$, $T \subseteq \mathcal{T}$. We call a transaction T *supports* an item $x \in \mathcal{T}$ if x is in T. [4]

An itemset X is one item x or a set of items $x_i, ..., x_j$, where $x_i, ..., x_j \in \mathcal{T}$. Similarly, we call a transaction T *supports* an itemset $X \subseteq \mathcal{T}$ if T *supports* every item $x \in X$.

The absolute support of an itemset X in dataset $\mathcal{D}$ is the number of transactions in $\mathcal{D}$ which supports this itemset X. The relative support of an itemset X, $\sigma(X/\mathcal{D})$ in a dataset is the absolute support of itemset X versus the total number of transactions in $\mathcal{D}$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset \mathcal{T}, Y \subset \mathcal{T}, X \neq \emptyset, Y \neq \emptyset, X \cap Y = \emptyset$. The confidence of an association rule $X \Rightarrow Y$ in dataset $\mathcal{D}, \varphi(X \Rightarrow Y)/\mathcal{D}$ is the ratio of $\sigma(X \cup Y/\mathcal{D})$ versus $\sigma(X/\mathcal{D})$, that is,

$$\sigma(X \Rightarrow Y) = P(X \cup Y)^5 \tag{1}$$

$$\varphi(X \Rightarrow Y) = P(Y|X) \tag{2}$$

The confidence of rule $X \Rightarrow Y$ can also be written as the following equation (3); Based on this equation, the confidence of the the association rule can be easily calculated by the support of X

[4] This definition of support can be extended easily with ancestors and descendant to multilevel pattern mining

[5] Note that $X \cup Y$ is the situation X and Y both appear in one record

and $X \cup Y$. This implies that the problem of mining association rules can be viewed as mining frequent patterns.

$$\varphi(X \Rightarrow Y) = P(Y|X) = \frac{\sigma(X \cup Y)}{\sigma(X)} \tag{3}$$

In frequent mining applications, users or experts set a minimum support (σ_{min}) and a minimum confidence (φ_{min}) to generated association rules. We call association rules that satisfy both σ_{min} and φ_{min} strong association rules. In general, the objective of frequent mining is to find proper strong association rules in the given dataset.

Frequent pattern mining is studied thoroughly during the last two decades, which has become an important branch in data mining territory. This topic is initially discussed in [Agrawal, *et al.*, 1993]. Later on, many variants are studied extensively, for example, FP-growth algorithm for fast frequent itemsets generation [Han, *et al.*, 2000], multi-dimensional frequent sequence mining [Pinto, *et al.*, 2001], mining multilevel association rules based on a progressively support decrease algorithm: ML_T2L1 [Han dan Fu, 1999], pushing constraint into FP-growth algorithm [Pei, *et al.*, 2001]. In the next section, we discuss the concept of multilevel frequent pattern mining.

3.1.1 Multilevel Frequent Pattern mining

As mentioned before, products taxonomy is worth investigating and the product sparsity need to be taken care by reducing dimensions. Therefore, we introduce multilevel frequent pattern mining to study the taxonomy of products and to reduce dimension. The technique is detailed as follows:

Multilevel frequent pattern mining is an important branch of frequent pattern mining. This level wise pattern mining is applied on the dataset with abstraction levels. (i.e. a taxonomy on items, also called *is-a* hierarchy). For example, a taxonomy as shown in Figure 3, which says 1. Soccer *is-a* outdoor sport, *is-a* outdoor activity, 2. Picnic *is-a* outdoor activity. Given such taxonomy, We may want to know the probability that a person who has picnic also plays football. This example can be seen as a rule: $do(X, soccer) \Rightarrow do(X, picnic)$, where X is the person, picnic and soccer are activities.

There are two preliminaries for multilevel frequent pattern mining[Han dan Fu, 1999]. Firstly, a dataset whose items have an implicit/explicit taxonomy. For example, as shown in Figure 3, we define a hierarchy based on facts, where *outdoor activities* is the highest level; *outdoor sports, picnic* are the middle level and *soccer* is the leaf level. Other techniques such as hierarchy clustering may also help to discover implicit hierarchy in the dataset. Secondly, proper mining techniques shall be used to efficiently find interesting frequent patterns. Previous studies [Han dan Fu, 1999], [Srikant dan Agrawal, 1996] proposed efficient methods to conduct multilevel frequent pattern mining, and to prune uninteresting patterns.

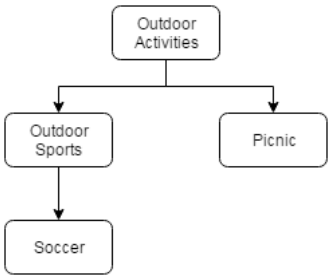


Figure 3: An example is-a relation

There are two preliminaries for multilevel frequent pattern mining[Han dan Fu, 1999]. Firstly, a dataset whose items have an implicit/explicit taxonomy. For example, as shown in Figure 3, we define a hierarchy based on facts, where *outdoor activities* is the highest level; *outdoor sports, picnic* are the middle level and *soccer* is the leaf level. Other techniques such as hierarchy clustering may also help to discover implicit hierarchy in the dataset. Secondly, proper mining techniques shall be used to efficiently find interesting frequent patterns. Previous studies [Han dan Fu, 1999], [Srikant dan Agrawal, 1996] proposed efficient methods to conduct multilevel frequent pattern mining, and to prune uninteresting patterns.

For most applications, multilevel FP mining starts from high abstraction level. However, many of these rules show common sense, which are neither detailed nor interesting enough to contribute to further work [Han, *et al.*, 2011]. Instead of only looking for frequent patterns in higher abstraction level, drilling down to lower levels can help to discover more informative and interesting patterns. In addition, frequent patterns that cross levels are also interesting, because such patterns show relationship between individual items and group of items. Therefore, we apply two related but different approaches to mine frequent patterns from different abstraction levels. The first approach aims to solve cross-level frequent pattern mining based on a straight forward thought: in each transaction record, replace the descendant items with the corresponding ancestor, and apply frequent pattern mining algorithm on this modified transaction dataset. The second approach is a progressively deepening method introduced by [Han dan Fu, 1999], which uses an encoded transaction dataset to progressively mine patterns and eliminate uninteresting patterns.

In the following content, an extended strong association rule definition is introduced for progressively deepening frequent pattern mining. Then we explain two approaches that are used for multilevel frequent pattern mining. The experiment results of these two approaches are presented in Section 3.3.1.

Definition 3.1. Extended strong association rule definition: Given σ_l at level l, an itemset X in $\mathcal{D}$ is frequent if the number of transactions T that supports X is no less than σ_l. A rule $A \Rightarrow B$ in $\mathcal{D}$ is strong if, each ancestors of every item in A, B is frequent at corresponding level (i.e. the level that ancestors resides in), itemset $A \cup B$ is frequent at current level. A preset minimum confidence is satisfied. [Han dan Fu, 1999]

Progressively deepening frequent pattern mining. Definition 3.1 implicitly states that only frequent patterns, that for each items in the pattern whose corresponding ancestor is frequent, can be generated. The implementation of this definition can effectively prune many uninteresting patterns by eliminating infrequent ancestors and their descendants, thus, uninteresting patterns of descendants will not be generated. In the following content, we use an example to illustrate how this approach works in a concrete way for ease to understand.

Table 2: Transaction dataset

TransactionID	Items
T1	ColorS1, ColorB2, Ewhite1, ColorB1
T2	ColorS1, ColorS2, ColorB2
T3	ColorI1, ColorS1, ColorS2,ColorB1
T4	white1, ColorS2, ColorB2, ColorB3
T5	white1, white2, ColorS1
T6	white1, white2, ColorS1, ColorI1
T7	ColorA1, ColorA1
T8	Ewhite1, Ewhite1

Table 3: Encoded Transaction dataset

TransactionID	Items
T1	ColorColorSColorS1, ColorColorBColorB2, EwhiteEwhiteEwhite1, ColorColorBColorB1
T2	ColorColorSColorS1, ColorColorSColorS2, ColorColorBColorB2
T3	ColorColorIColorI1, ColorColorSColorS1, ColorColorSColorS2,ColorColorBColorB1
T4	whitewhitewhite1, ColorColorSColorS2, ColorColorBColorB2
T5	whitewhitewhite1, whitewhitewhite2, ColorColorSColorS1
T6	whitewhitewhite1, whitewhitewhite3, ColorColorSColorS1, ColorColorIColorI1
T7	ColorColorAColorA1, ColorColorAColorA1
T8	EwhiteEwhiteEwhite1, EwhiteEwhiteEwhite1

Given a hierarchy of items, as shown in Figure 2, and a example transaction dataset shown in Table 2. we encode this transaction dataset to the form as Table 3. For each item in the Table 2, we encode the item name as: $highlevel+midlevel+leaflevel$. For example, *ColorS1* is encoded as *ColorColorSColorS1* , where *Color* represents the highlevel group of products, *ColorS* represents midlevel group of products and *ColorS1* is the item itself.

Given preset minimum support for highlevel $\sigma_{high} = 4$, by scanning Table 3 once, the highlevel frequent 1-itemset table $\mathcal{L}$[high, 1] is derived as shown in Table 4. Similarly, highlevel 2-itemset table $\mathcal{L}$[high, 2] is derived, where only 1 frequent itemset is generated.

According to Definition 3.1, midlevel items, other than the descendants of frequent highlevel items (i.e. Color** and White** in 4), are eliminated. Therefore, we use Table 4 $\mathcal{L}$[high, 1] to eliminate those midlevel items and the transactions that do not contain frequent highlevel items. Table 6 shows newly generated encoded transaction dataset. Compared with the original encoded transaction dataset in Table 3, encoded item *EwhiteEWhiteEwhite1* is removed from transaction *T1* as shown in Table 6; transaction *T8* is eliminated, because it does not contain any frequent highlevel items. This newly generated transaction dataset will be used as the original transaction dataset for

Table 4: $\mathcal{L}$[high, 1]

Support	Itemset
7	Color**
4	white**

Table 5: $\mathcal{L}$[high, 2]

Support	Itemset
4	white**, Color**

Table 6: Encoded Transaction dataset midlevel

TransactionID	Items
T1	ColorColorSColorS1, ColorColorBColorB2, ColorColorBColorB1
T2	ColorColorSColorS1, ColorColorSColorS2, ColorColorBColorB2
T3	ColorColorIColorI1, ColorColorSColorS1, ColorColorSColorS2,ColorColorBColorB1
T4	whitewhitewhite1, ColorColorSColorS2, ColorColorBColorB2
T5	whitewhitewhite1, whitewhitewhite2, ColorColorSColorS1
T6	whitewhitewhite1, whitewhitewhite3, ColorColorSColorS1, ColorColorIColorI1
T7	ColorColorAColorA1, ColorColorAColorA1

Table 7: $\mathcal{L}$[mid, 1]

Support	Itemset
6	*ColorS*
4	*ColorB*
3	*white*

Table 8: $\mathcal{L}$[mid, 2]

Support	Itemset
3	*ColorS*, *ColorB*
3	*ColorS*, *white*

Table 9: Encoded Transaction dataset leaflevel

TransactionID	Items
T1	ColorColorSColorS1, ColorColorBColorB2, ColorColorBColorB1
T2	ColorColorSColorS1, ColorColorSColorS2, ColorColorBColorB2
T3	ColorColorSColorS1, ColorColorSColorS2,ColorColorBColorB1
T4	whitewhitewhite1, ColorColorSColorS2, ColorColorBColorB2
T5	whitewhitewhite1, whitewhitewhite2, ColorColorSColorS1
T6	whitewhitewhite1, whitewhitewhite3, ColorColorSColorS1

midlevel frequent pattern mining. In addition, {White**, Color**} is this level's frequent itemset.

Similarly, we apply frequent pattern mining on midlevel encoded transaction dataset. We set the minimum support $\sigma_{mid} = 3$. We scan the transaction dataset as shown in Table 6, and generate $\mathcal{L}$[mid, 1], $\mathcal{L}$[mid, 2], as shown in Table 7, Table 8 respectively. Infrequent midlevel items are eliminated, as well as transactions that do not contain frequent midlevel items (e.g. transaction $T7$).

At last, we scan Table 9 given minimum support $\sigma_{leaf} = 2$. Then generate the frequent leaflevel items and itemsets, as shown in Table 10, Table 11 respectively. Notice that in Table 11, all the absolute support of 2-item itemsets are relatively low $\sigma = 2$, which shows the advantage of using progressively deepening method: allow patterns in low level with lower support to be generated.

Table 10: $\mathcal{L}$[leaf, 1]

Support	Itemset
5	**ColorS1
3	**ColorS2
3	**ColorB2
2	**ColorB1
3	**white

Table 11: $\mathcal{L}$[leaf, 2]

Support	Itemset
2	**ColorS1, **ColorB1
2	**ColorS1, **ColorB2
2	**ColorS2, **ColorB2
2	**ColorS2, **ColorS1
2	**white, **ColorS1

Table 12: Example of cross-level Transaction dataset

TransactionID	Items
T1	ColorS, ColorB2, Ewhite1, ColorB1
T2	ColorS, ColorS, ColorB2
T3	ColorI1, ColorS, ColorS,ColorB1
T4	Ewhite1, ColorS, ColorB2, ColorB3
T5	Ewhite1, white2, ColorS
T6	white1, white3, ColorS, ColorI1
T7	ColorA1, ColorA1
T8	Ewhite1, Ewhite1

Table 13: 1-item cross level frequent itemsets

Support	Itemset
6	ColorS
4	ColorB2
3	white1

Table 14: 2-item cross level frequent itemsets

Support	Itemset
3	ColorS, ColorB2
3	ColorS, white1

The output patterns of this process are $\mathcal{L}$[high, 1],$\mathcal{L}$[high, 2],$\mathcal{L}$[mid, 1],$\mathcal{L}$[mid, 2],$\mathcal{L}$[leaf, 1],$\mathcal{L}$[leaf, 2], in which the 2-item itemsets (note that in real datasets, k-item itemsets is possible to appear, where $k \geq 3$) can be used to generate association rules.

Cross-level association rule mining. In previous subsection, we explained the design of level wise frequent pattern mining. However, this method can only discover patterns in the same level (i.e. for each pattern, only items from same level included); only association rules that every item in the rule comes from same abstraction level can be generated. If we want to know the relation between *ColorS* and *White1*, such patterns can not provide enough information. Therefore, in this subsection ,we propose a cross-level frequent pattern mining method which can discover association rules cross-level.

The idea behind this approach is very straightforward. Since we want to find how ancestors of certain groups of descendants behave against other descendants, we simply replace those groups of descendants with their ancestors in the transaction dataset. Then apply frequent pattern mining algorithms on the processed transaction dataset. The output frequent patterns can be used to generate cross-level association rules. As an example, given a transaction dataset as shown in Table 2, a taxonomy on items as shown in 2, we want to discover how *ColorS1* and *ColorS2* behave as a group against other items in this dataset. First, for each transaction in this dataset, *ColorS1* and *ColorS2* are replaced by their ancestor *ColorS*, as shown in Table 12. Then we apply frequent pattern mining algorithm on this dataset. Given minimum support $\sigma = 3$, following frequent itemsets are generated, as can be seen in Table 13 and Table 14.

3.1.2 Interesting measures

In the previous subsection, two approaches to generate multilevel frequent pattern are introduced. The first approach progressively generate frequent pattern from top level to bottom level and prune potential uninteresting patterns in advance. The second approach enable cross-level discovery of frequent patterns. By using the results above, we are able to generate association rules for multi-

level analysis needs.

In this section, we first explain how to generate strong association rules from previous results. Afterwards, a case is shown to illustrate why strong association rule is not good enough for further use. At last, several other evaluation techniques of association rules are introduced, and we explain how this statistical based evaluation process can eliminate many uninteresting association rules.

I. Strong association rule. Strong association rules are derived from the frequent patterns. According to Equation 3, the confidence of association rule $\varphi(X \Rightarrow Y)$ can be easily calculated given the support of LHS [6] $\sigma(X)$ and the support of RHS $\sigma(X \cup Y)$. The derived strong association rules satisfy preset minimum frequency and minimum confidence. However, the preset minimum frequency and minimum confidence shall be carefully chosen, for example, too small a minimum frequency may results in too large a rule set and too small a confidence results in less strong rules. Moreover, given a proper minimum frequency and minimum confidence, many uninteresting rules can still exist. Therefore, in this subsection, we show the an example of uninteresting rule using the example from [Han, *et al.*, 2011], and provide other interest measures that help to exclude uninteresting rules.

II. Minimum frequency and Minimum confidence is not enough. Suppose a transaction dataset with 10000 records in which 6000 records include *PC games*, 7500 records include *videos* and 4000 include both. Given preset minimum support and minimum confidence 30% and 60% respectively. We have an association rule:

$$PCgames \Rightarrow videos \quad (support = 40\%, confidence = 66\%) \tag{4}$$

Obviously, this is a strong association rule, which is interesting according to the definition of strong association rule. However, notice that the support of purchasing *videos* is 75%, which is higher than 66%. This indicates that these two products are negatively related and the appearance of *PCgames* will decrease the appearance probability of *videos*. Therefore, this rule shall not be recommended. To eliminate such rules, other interest measures should be introduced.

III. Correlation Analysis. Some previous researches only use support and confidence thresholds to generate association rules for applications. For example, Mobasher, *et al.* [2001] introduce an association rule based approach for Web personalization which only considers support and confidence to filter rules. However, as shown above, strong association rules may lead to unreasonable decisions. Because the correlation between items/itemsets is not considered in support and confidence thresholds. Therefore, several correlation analysis criteria are discussed here to take care of correlation between LHS and RHS of association rules.

[6] LHS refers to left hand side of a rule, for example: X is the LHS of rule $X \Rightarrow Y$. Similarly, RHS is the right hand side of a rule

IV. Lift. We first introduce correlation measure *Lift*, as shown in equation 5. In our case $P(A \cup B)$ is the support of $A \cup B$ $\sigma(A \cup B)$. Similarly, $P(A)$ and $P(B)$ is the support of A and B: $\sigma(A)$ and $\sigma(A)$ respectively.

$$lift(A, B) = \frac{P(A \cup B)}{P(A)P(B)} = \frac{\sigma(A \cup B)}{\sigma(A)\sigma(B)} \tag{5}$$

If $lift = 1$, the purchase of A is independent from the purchase of B, if not, A and B are dependent. Notice that A,B are not restricted to items; this definition can be extended to itemsets. If $lift > 1$, A and B have a positive relation, which means the purchase of A or B will increase the probability of purchasing another. In contrast, if $lift < 1$, A and B have a negative relation, which means the purchase of A or B will decrease the probability of purchasing another.

Consider the previous example: *PCgames, videos*. Since the probability of purchase *PCgames*,*videos* and the probability of purchase both are given, the lift of rule 4 can be calculated by using equation 5. Thus, $lift(PCgames, videos) = \frac{P(PCgames \cup videos)}{P(PCgames) \times P(videos)} = \frac{0.40}{0.60 \times 0.75} = 0.89$. Based on the introduction above, because $lift(PCgames, videos) < 1$, we say that the correlation between purchase of *PCgames* and purchase of *videos* is negative. This statistic result can be used in real lift problem, for example, *PCgames* and *videos* shall not be placed together.

V. Interest measures for association rule. In the previous section, we introduce *lift* to solve the problem of misleading strong association rules. *Lift* is capable to correctly identify the correlation between LHS and RHS, however, not in all situation. In [Han, *et al.*, 2011], the author shows that lift is not capable to correctly identify correlation when the number of *null transaction*[7] outweigh the number of target transaction. To correctly identify correlation of certain rules in a transaction dataset with large amount of *null transaction*, several other statistics are considered.

In [Omiecinski, 2003], *all confidence* is introduced as an alternative measure, which is defined below:

$$all\_confidence(X, Y) = \frac{\sigma(X \cup Y)}{max\{\sigma(X), \sigma(Y)\}} \tag{6}$$

In [Tan, *et al.*, 2004], *cosine* measure is introduced, which is defined below:

$$cosine(X, Y) = \frac{P(X \cup Y)}{\sqrt{P(X) \times P(Y)}} = \frac{\sigma(X \cup Y)}{\sqrt{\sigma(A) \times \sigma(B)}} \tag{7}$$

In [Wu, *et al.*, 2007], *Kulczynski measure* is introduced based on the finding of S. Kulczynski (1927). The definition is shown below:

$$kulc(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{2}(\frac{1}{\sigma(X)} + \frac{1}{\sigma(Y)}) \tag{8}$$

[7] *null transaction* are transactions do not contain target items/itemsets

These measures are proved *null-invariant* in [Han, *et al.*, 2011]. We say a measure is *null-invariant* if the result do not suffer from *null transaction* [Tan, *et al.*, 2002]. Therefore, we integrate the interest measures above in our frequent pattern mining framework. In addition, range of the measures above is [0,1], where close to 0 indicates a negative correlation and close to 1 indicates a positive correlation.

Apart from *null transaction*, imbalance of LHS and RHS also influences interest measures. Here, imbalance means the huge difference between the number of LHS and RHS. To distinguish those association rules that may suffer from imbalance, Wu, *et al.* [2010] introduces *imbalance ratio* to calculate the imbalance of LHS and RHS. Formal definition as follows:

$$imbalance\_ratio(X, Y) = \frac{|\sigma(X) - \sigma(Y)|}{\sigma(X) + \sigma(Y) - \sigma(X \cup Y)} \tag{9}$$

The result range is [0,1), where 0 indicates a balance rule; more closer to 1 indicates more imbalance the rule is. An example is provided in [Han, *et al.*, 2011] to explain this measure in detail; the author also shows that *kulc* can generate reasonable results given imbalance association rules. Based on the discussion above, the author select *kulc* and *imbalance ratio* as two major measures for frequent pattern mining tasks.

The statistics discussed above are essential for discovering interesting association rules. In Section 3.3.1, we show the experiment result of association rule mining.

3.2 Data wrangling

Based on the dataset introduced in Section 2, we first pre-process this dataset. For each independent system (customer), whenever the number of products in this system changes, we extract the record of this system at that time; then concatenate these records together as one record. After we have this change-record, we further extract only the information related to time, changed products and user ID.

In the end, the pre-processed dataset is organized as follows: a nested list, in which each sub-list is a user's change-record. First item in a change-record is userID, second item is week number, and third item is the product purchased in that week. Table 15 shows an example record of user 1. First column of this table is userID, second column is week number and the third column is the products that are newly added to the system at that week.

Table 15: Example of pre-processed data

User ID	week number	products
1	0	white1, white1
1	4	ColorS1
1	18	Special1

Based on the pre-processed dataset introduced above, further processing is applied to make the dataset suitable for association rule mining.

Product information is extracted from each system's all records and organized as a new record for every system. These records contain all the products each customer (system) has ever bought (contain products being removed). We concatenate the transactions of a user's purchase history instead of using single transactions because most of the transactions only include one product. As can be seen in Figure 4, more than 600 transactions only contain one product.

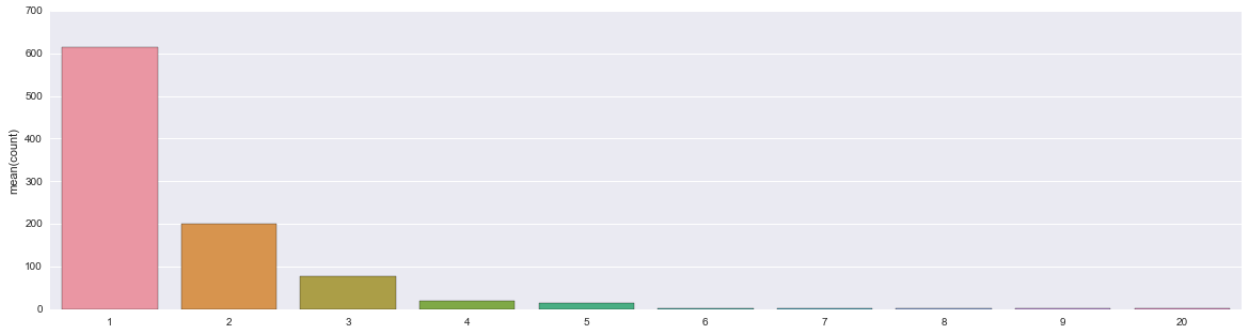


Figure 4: Number of products purchased in transactions

Therefore, the data format is described as follows: a nested list where each sub-list is a user's transaction record which contains all the products this user has ever bought. In total 4208 users' records are included. Table 16 shows an example of input dataset for association rule mining. Each row in this table represents all the products that have been added to the dataset.

Table 16: Example of input data from association rule mining

changed products
white1, white1, ColorS1, white3
ColorS1,ColorS1,ColorS1
Special1,white2, Ewhite1, Ewhite1

3.3 Experiment setting

All experiments are conducted on windows 7 64-bit Operating system, Intel(R) Core(TM) i5-5300U CPU 2.30GHz, 4.00 GB RAM.

3.3.1 Association rule mining experiment

We first conduct the experiment of association rule mining. In this experiment, we define objectives as follows:

1. Pick proper minimum support and minimum confidence, then compare the results generated by different support and confidence settings.

2. Calculate interest measures of association rules, show how these measures eliminate uninteresting rules.

In the latter part of this sub-section, both objectives are discussed extensively with results generated from the sample dataset. A python package Orange3-Associate [8] is used in our study, which provides a fast frequent pattern mining algorithm FP-growth (without candidate generation) [Han, *et al.*, 2000],[Han, *et al.*, 2004]. We explain both the reason to use FP-growth for our task and the algorithm itself in Appendix A.1.

I. Pick proper support and confidence value. To generate a strong association rule, minimum confidence and minimum support have to be defined before running the algorithm. However, there is no standard way to pick a proper minimum support and minimum confidence for each specific problem. For example, in [Sandvig, *et al.*, 2007], the robustness of a recommendation algorithm based on association rule mining is studied; the author uses relative support: $\sigma = 0.1$ to generate frequent patterns; in [Wong, *et al.*, 2005], to find so called 'respond' rules, which is relatively rare, the minimum support value is set below 5%. Examples above shows that people select proper minimum support in different context, which is also true for minimum confidence. Therefore, minimum support and minimum confidence shall be carefully selected such that enough amount of association rules can be generated and these generated association rules are interesting to users.

Previous research also suggest options to use dynamic minimum support and minimum confidence. For example, use multiple support value for web page with different importance. In [Mobasher, *et al.*, 2001], to reserve particular important information of content-oriented pages (situated deeply in the website, thus quite rare), a minimum support of 0.01 is assigned to these pages. In contrast, less informative pages are assigned minimum support of 0.1. Another example is progressively decrease minimum support for different abstraction level, which is introduced in Section 3.1.1.

To evaluate the selected minimum support and minimum confidence, we compare number of association rules generated given different combination of support and confidence value, where the minimum support is 842, 421, 30, 5 and the minimum confidence is 0.8, 0.6, 0.4, 0.3, 0.1. A heat map is depicted to show the result. Information of input dataset is introduced in Section 3.2.

II. Evaluate the effectiveness of interest measures. In the previous section, we introduced strategies to set minimum confidence and minimum support. Many uninteresting rules have been filtered out when generating strong association rules. However, not all strong association rules are good enough for further use; some of these rules are misleading [Han, *et al.*, 2011] as introduced

[8] http://orange3-associate.readthedocs.io/en/latest/scripting.html

before. Therefore, we conduct experiment to find interesting rules.

In this experiment, we first explain how dominating items influence the result. Then we implement interest measures: *kulc, imbalance ration, lift, all-confidence, max-confidence*. These interest measures are applied on the result of AR mining to find interesting AR. The effectiveness of each interest measures mentioned above is evaluated and those measures that are suitable for our sample dataset is selected. In the end, we use the selected interest measures to filter AR. The filtered rules are evaluated by domain experts to see if added value exists in our results.

3.3.2 Experiment of cross-level multilevel frequent pattern mining

As introduced in Section 3.1.1, multilevel frequent pattern mining is another important branch in this knowledge domain. The main contribution of this type of FP is that it enables users to have an insight of a dataset from different abstraction levels. Some AR might be 'invisible' before, however can be find when those rare items collected as a group (the support value of this group is the sum of all rare items). Although the rare items do not directly constitute the newly discovered AR, these AR do show information of rare items. In this subsection, we conduct an experiment to evaluate cross-level FP mining.

Each time we group one set of similar products to the corresponding higher abstraction level, for example, *white1, white2, white3* to *White*. This process is applied for every middle level *White, ColorB, ColorI, ColorS, ColorA, Ewhite*. Then we compare the number of AR generated given different middle level with number of AR generated without abstraction. The reason of number difference will be discussed, and the generated cross-level AR are evaluated by domain experts.

3.3.3 Experiment of progressively deepening frequent pattern mining

In Section 3.1.1, progressively deepening multilevel frequent pattern mining (PDMFPM) is detailed with a concrete example. In this section, we conduct an experiment to evaluate the benefit of using this approach to generate multilevel FP. There are two major benefits of using this method. First, this approach can automatically prune uninteresting rules. Second, by using different support value for different abstraction level rather than same support value for all levels, interesting rules from lower levels are reserved, uninteresting rules from higher levels are eliminated. In order to validate the benefits mentioned above, two sub experiments are designed.

In the first experiment, PDMFPM's ability to eliminate uninteresting rules is examined by using two different rule generation strategies. We use PDMFPM as first strategy, with gradually decrease support value $\sigma_{high}, \sigma_{middle}, \sigma_{leaf}$ for each abstraction level. We use another normal FP strategy as a control experiment, using same support value for each abstraction level. The results of both strategies are compared: the number of generated AR are compared to validate whether PDMFPM can efficiently eliminate uninteresting AR; the eliminated AR are checked to see whether

those rules are uninteresting.

In the second experiment, we show how using different σ for different abstraction level helps to eliminate uninteresting rules and reserve interesting rules. Three strategies are applied here. The first strategy use PDMFPM with gradually decrease support value; the second strategy use PDMFPM with same support value σ_{high}; the third strategy use PDMFPM with same support value σ_{leaf}. Generated AR from different strategies are compared to check whether gradually decrease support helps to reserve interesting rules as well as eliminate uninteresting rules.

In the end, the generated multilevel AR are evaluated by domain experts.

3.4 Result discussion

3.4.1 Association rule mining experiment

I. Pick proper support and confidence value. As can be seen in Figure 5, y-axis shows minimum confidence and x-axis shows minimum support. The degree of color shows number of association rules generated. Different threshold for minimum confidence and minimum support are set. For minimum confidence, we set 0.1, 0.3, 0.4, 0.6 and 0.8; for minimum support, we set 842[9], 421, 30 and 5 (absolute support). In this picture, less transparent part indicates more rules are generated using the corresponding support/confidence setting. It is quite obvious that lower minimum confidence/support result in more association rules generated. Given $\sigma = 842$ or $\sigma = 421$, no association rules are generated. Given $\varphi = 0/3$ and $\sigma = 5$, more than 200 association rules are generated. Given $\varphi = 0.1$ and $\sigma = 5$, more than 500 association rules are generated.

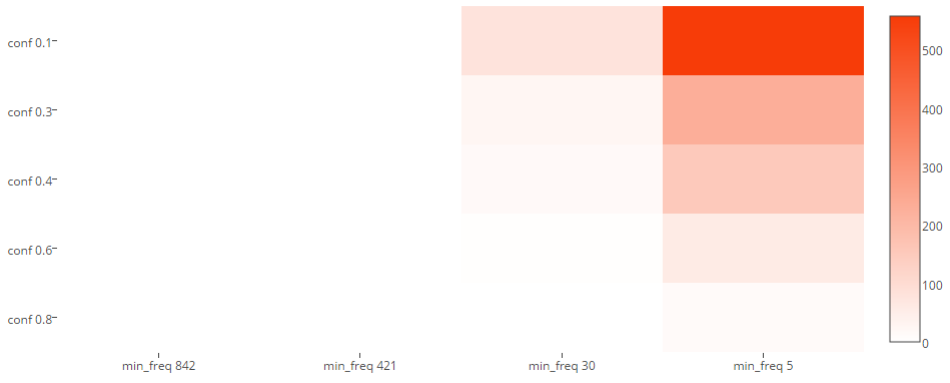


Figure 5: Number of association rules generated given different minimum confidence and minimum support

[9]minimum support 842 is relative support 0.2 in our case

From Figure 5, it is clear that a too high minimum support can hardly help to generate association rules. Setting confidence as 0.1 and minimum support as 5 do generate more than 500 rules, however the confidence is too low, which means the conditional probability is too low to be convincing. Therefore, based on number of generated rules given certain confidence and support threshold, we decide to use $\varphi_{min} = 0.2$ and $\sigma_{min} = 5$ for further analysis.

II. Evaluate the effectiveness of interest measures. As shown above, we generate 359 strong association rules given $\varphi_{min} = 0.2$ and $\sigma_{min} = 5$. Table 17 shows the result, where total number of AR is 359, among which 116 have *white1* in RHS. Such AR account for 32.3% of total amount, which is interesting as well as confusing. Therefore, we first discuss the topic: dominate item.

The absolute support value of *white1* is $\sigma = 2998$; the corresponding relative support value is $\sigma = 0.71$, which is very high. In this context, We call such a item dominate item if this item appears in most transactions. Apparently, *whtie1* is a dominate item.

Table 17: Association rule experiment result

number of FP	number of AR	number of RHS has *white1*	$kulc \geq 0.5$	$IR \leq 0.5$
262	359	116	8	15

Consider equation 3, if dominating item serves as LHS, we have:

$$\varphi(white1 \Rightarrow Y) = \frac{\sigma(white1 \cup Y)}{\sigma(white1)}$$

Since $\sigma(white1)$ is very big compared with $\sigma(Y)$, and $\sigma(white1 \cup Y)$ is even smaller, thus the confidence $\varphi(white1 \Rightarrow Y)$ is very small. Therefore, we can infer that AR $white1 \Rightarrow Y$ can not be a strong AR.

If dominating item serves as RHS, we have:

$$\varphi(Y \Rightarrow white1) = \frac{\sigma(white1 \cup Y)}{\sigma(Y)}$$

This AR do not suffer from low confidence value. However, even if this AR is a strong rule, it is not interesting to users. For example, in our case, *white1* is bought by most customers. If we want to recommend products to customers, *white1* is not the right one.

Based on the discussion above, we eliminate all the AR whose RHS is *white1*. For those AR with *white1* as LHS, none of them are strong AR, thus already eliminated. Table 18 shows the result after elimination of the case discussed above. As can be seen in Table 18, number of AR drop from 359 to 270; only 27 AR with *white1* in RHS remains; number of AR ($kulc \geq 0.5$) slightly decrease and number of $IR \leq 0.5$ remains unchanged.

Table 18: Association rule after eliminate *white1* as RHS

number of FP	number of AR	number of RHS has *white1*	$kulc \geq 0.5$	$IR \leq 0.5$
262	270	27	5	15

In [Han, *et al.*, 2011], the authors suggest that *kulc* and *imbalance ratio (IR)* shall be used to measure which ARs are interesting. Here, we investigate that among those interest measures we introduced in 3.1.2, which measures are suitable for our problem.

In Table 17 and Table 18, we notice that given 359 and 270 ARs, in both cases, only 15 of them satisfy $IR \leq 0.5$. This indicates that most ARs suffer from skewness. As announced by the authors of [Han, *et al.*, 2011], *kulc* performs better under skewness compared with other interest measures such as *lift, all-confidence, max-confidence, etc.* In an extreme skewness situation, the value of *kulc* should be around 0.5. To have a clear view, another experiment is conducted by using more strict thresholds.

Table 19: Evaluate *kulc* together with *IR*

number of AR	$kulc \geq 0.4$	$IR \geq 0.9$
270	9	144

As shown in Table 19, with 144 ARs that $IR \geq 0.9$, only 9 ARs' $kulc \geq 0.4$, which is strange, because as mentioned above: given extreme skewed AR, the *kulc* value should be around 0.5. Although we decrease the threshold to 0.4 (in order to contain ARs whose *kulc* value slightly below 0.5), only 9 ARs remain. Therefore, we look into the results to investigate why *kulc* is not working well. In addition, we evaluate other measures to find proper ones for selecting interesting rules.

Two randomly picked ARs are presented here:

$$\{white2, ColorI2, ColorS1\} \Rightarrow \{white1, white2\}$$

$$(\varphi = 42.9\%, \sigma(LHS) = 14, \sigma(RHS) = 318)$$

$$\{ColorI2, ColorB1, Special1\} \Rightarrow \{ColorS1\}$$

$$(\varphi = 50\%, \sigma(LHS) = 32, \sigma(RHS) = 670)$$

Given the information above, we are able to calculate the statistics as shown below:

Table 20: Rule statistics

AR	kulc	IR	σ	$\sigma(LHS)$	$\sigma(RHS)$	lift	P(RHS)	φ	max confidence	all confidence
Rule 1	0.22	0.93	6	14	318	5.67	7.6%	42.8%	43%	1.8%
Rule 2	0.26	0.93	16	32	670	3.14	15.9%%	50.0%	50%	2.4%

From Table 20, we can easily find out that $\varphi > P(RHS)$, which means the presence of LHS lift/increases the probability that RHS presents. This indicates that the correlation of this AR is positive. However, *kulc* cannot show this positive correlation, because both *kulc* values show negative correlation. In contrast, both *lift* values show positive correlation. From the perspective of skewness, it is hard to say one lifts another in an AR (although the results support positive correlation). Therefore, statistics showing neutral correlation are reasonable as well. In Table 20, both *max confidence* show neutral correlation.

We further investigate why *kulc* behaves bad in our case. Consider equation 8, given a skewed AR, we assume $\sigma(X) \gg \sigma(Y)$. We have:

$$kulc(X \Rightarrow Y) \approx \frac{\sigma(X \cup Y)}{2 \times \sigma(Y)}$$

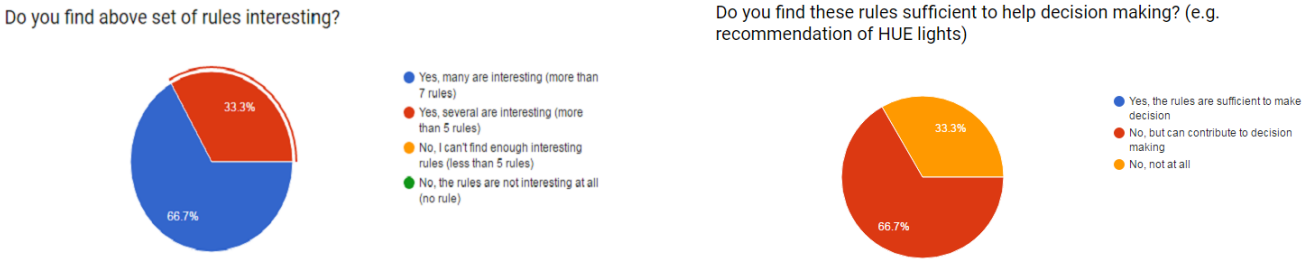
Since $\sigma(X \cup Y)$ is even smaller than $\sigma(Y)$, therefore the *kulc* value can hardly reach 0.5 (neutral). Based on the discussion above, we decide to use *lift, max confidence, imbalance ration* as three proper interest measures for our particular problem.

III. Expert evaluation. $Lift > 1$ when $IR < 0.5$, and $0.4 < maxconfidence < 0.6$ when $IR \geq 0.5$ are applied as interest measures to find interesting ARs. Table 21 shows the generated rules using interest measures above, where the rules are ordered by their rank value $\sigma * \varphi$. The generated rules are evaluated by domain experts to see if added value exists in our results.

We select 2 groups of association rules (satisfy above interesting measures) for evaluation, where each group contains 10 rules. After viewing these two groups of rules, domain experts have to answer 2 questions.

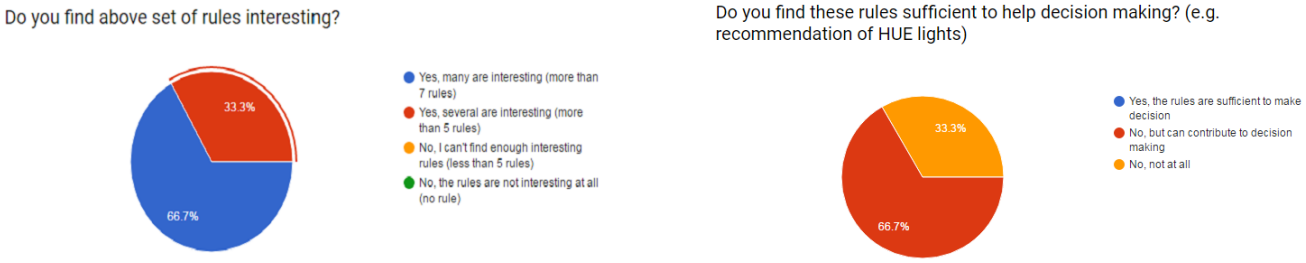
Figure 6 and Figure 7 shows the results of expert evaluation of selected association rules. In addition, full set of mined association rules are presented to domain experts as well. We want to know whether the ordering of association rules are reasonable, namely whether the more interesting rules are in the front of less interesting rules. Two domain experts give positive feedback and one give negative feedback.

From the discussion above, we know that domain experts find many generated association rules are interesting. None of them think the results are sufficient to make decision, however two of them think the results can contribute to decision making. For full set of association rules, two experts think the ordering is reasonable and another thinks the opposite.



Three domain experts evaluate the association rule mining result. For the first question, 2 of them think more than 7 rules (out of 10) are interesting to them, the other one thinks more than 5 rules are interesting. For the second question, 2 of them find these association rules not sufficient, however can contribute to decision making; 1 expert finds the rules are not helpful at all.

Figure 6: Domain experts evaluation of AR, first group



The second group of association rules have same expert evaluation as the first group of association rules have.

Figure 7: Domain experts evaluation of AR, second group

Table 21: Selected association rules

LHS	RHS
{'ColorI2'}	{'ColorS1'}
{'ColorB1'}	{'ColorI2'}
{'ColorI2'}	{'ColorB1'}
{'ColorS1'}	{'ColorI2'}
{'ColorB1'}	{'ColorS1'}
{'ColorS1'}	{'ColorB1'}
{'white1"ColorI2'}	{'ColorS1'}
{'Special1'}	{'ColorB1'}
{'Special1'}	{'ColorI2'}
{'Special1"ColorI2'}	{'ColorB1'}
{'white1"ColorS1'}	{'ColorI2'}
{'ColorB1"Special1'}	{'ColorI2'}
{'ColorB1"Special1'}	{'ColorS1'}
{'Ewhite2'}	{'Ewhite1'}
{'Special1"ColorS1"ColorI2'}	{'ColorB1'}
{'ColorB1"Special1"ColorI2'}	{'ColorS1'}
{'ColorB1"Special1"ColorS1'}	{'ColorI2'}
{'white1"ColorB1"Special1'}	{'ColorS1'}
{'Ewhite1 "ColorS1'}	{'Ewhite1'}
{'white1"ColorS1"white2'}	{'Ewhite1'}
{'Ewhite1 '}	{'white1"Ewhite1'}
{'white1"ColorB1"ColorI2'}	{'ColorS1'}
{'ColorB1"ColorS1'}	{'Special1'}
{'Ewhite1 "ColorI2'}	{'Ewhite1'}
{'white1"Special1"ColorS1'}	{'ColorB1'}
{'white1"Special1"ColorI2'}	{'ColorS1'}
{'white1"Special1"ColorI2'}	{'ColorB1'}
{'white1"Ewhite1"ColorB1'}	{'ColorS1'}
{'white1"ColorB1"Special1"ColorS1'}	{'ColorI2'}
{'white1"ColorB1"white2'}	{'ColorS1'}
{'white1"ColorB1"white2'}	{'Ewhite1'}
{'ColorB2"Special1'}	{'ColorI2'}
{'Ewhite1"white3'}	{'ColorS1'}
{'ColorB1"ColorS2'}	{'Ewhite1'}
{'ColorS2"ColorI2'}	{'ColorB1'}
{'ColorB1"ColorS2'}	{'ColorI2'}
{'ColorI2"white3'}	{'ColorB1'}
{'Special1"ColorI2'}	{'ColorB1"ColorS1'}
{'white1"ColorB1"ColorI2"ColorS1'}	{'Special1'}
{'Ewhite1"ColorB1"ColorI2'}	{'ColorS1'}
{'Ewhite1 "ColorB1'}	{'Ewhite1'}
{'Special1"ColorS1'}	{'ColorB1"ColorI2'}
{'Ewhite1 "white2'}	{'Ewhite1'}
{'Ewhite2"white1'}	{'Ewhite1'}
{'ColorA1'}	{'ColorS1'}
{'ColorA2'}	{'ColorB1'}
{'ColorA2'}	{'ColorI2'}
{'Ewhite1"ColorB1"white2'}	{'white1"ColorS1'}
{'white1"white3"white2'}	{'ColorS1'}
{'Ewhite1"Ewhite1 "ColorI2'}	{'ColorS1'}
{'white1"ColorS2"white2'}	{'ColorS1'}
{'white1"ColorB1"ColorS2'}	{'ColorI2'}
{'Ewhite1"ColorB1"white2'}	{'ColorS1'}
{'Special4'}	{'ColorI2"ColorS1'}
{'Special5'}	{'ColorS2'}
{'Special5'}	{'ColorS1'}
{'Special3'}	{'Special2'}
{'Special4'}	{'Ewhite1'}
{'Special3'}	{'ColorS1'}
{'Special4'}	{'ColorI2'}
{'ColorI2"ColorS1"white2'}	{'white1"Ewhite1'}
{'Ewhite1"ColorI2"white2'}	{'white1"ColorS1'}
{'white1"ColorB1"white2'}	{'Ewhite1"ColorS1'}
{'Ewhite1"Ewhite1 "ColorS1'}	{'ColorI2'}
{'white1"Ewhite1 "ColorS1'}	{'Ewhite1'}
{'ColorI2"ColorS1"white2'}	{'Ewhite1'}
{'Ewhite1"ColorI2"white2'}	{'ColorS1'}
{'Special1"white3'}	{'white1"ColorS1'}
{'Ewhite1 "white3'}	{'white2'}
{'Ewhite2"ColorI2'}	{'Ewhite1'}
{'ColorB1'}	{'ColorB1'}

In total 359 association rules are generated given $\sigma = 5, \varphi = 0.2$. We post process the rules to eliminate the influence of dominating products, eliminate uninteresting rules by using interest measures. Rules above satisfy: $Lift > 1$ when $IR < 0.5$, and $0.4 < maxconfidence < 0.6$ when $IR \geq 0.5$

3.4.2 Experiment of cross-level multilevel frequent pattern mining

Table 22: Number of AR with different abstraction

Middle level to abstract	white	ColorB	ColorI	ColorA	ColorS	Ewhite	without Abstraction
Number of AR excluding white1	248	275	279	270	253	269	270
Number of AR	248	359	366	357	331	352	359

As shown in Table 22, applying multilevel FP mining do change the number of ARs generated. Consider middle level *White*, which consists of *white1, white2, white3*. Both the number of ARs and the number of ARs excluding *white1* decrease dramatically. This happens because *white1, white2, white3* all have relatively high support compared with $\sigma(min) = 5$, which means for these three lights, the chance of passing $\sigma(min = 5)$ is high. Thus, abstracting these lights to a higher level actually reduces the variety of ARs and likely reduce the total number of ARs. On the contrary, abstraction of middle level *ColorI* increases the total number of ARs generated. This is because the support values of *ColorI3* and *ColorI1* (leaf level of *ColorI*) are not that high, When these two lights serve as a part of an AR, the support value of this AR may not high enough to pass $\sigma(min) = 5$. However, when grouping as *ColorI*, the support value of *ColorI* is high enough to have better chance to become a strong AR. This can be useful when the company wants to promote products that are barely bought by customers. By looking for higher level ARs of these products, relation with other products[10] can be found. Then recommendations based on this kind of relation are possible.

From the comparison above, we know new ARs can be discovered by cross-level AR mining. We show several discovered ARs here.

$$\{ColorB, ColorS1, Ewhite1\} \Rightarrow \{white1, white2\}$$

$$\{ColorB2, Special1, ColorS1\} \Rightarrow \{ColorI\}$$

3.4.3 Experiment of progressively deepening frequent pattern mining

Eliminate uninteresting FPs by removing not-qualified ancestors and transactions. To eliminate uninteresting rules in multilevel FP mining, one can choose to eliminate those rules after all the rules are generated (post analysis). An alternative approach is to remove possible uninteresting rules during the generating process. This approach can improve the efficiency of generating process and reduce the workload to analyze rules. In this experiment, we show the superiority of second approach, focusing on the benefit of reducing the analysis workload (the workload is reduced because the total number of FPs is less).

[10]Note that AR mining is about but not restricted with products. This is discussed in the area of Multidimensional pattern mining which we do not emphasize here

Table 23: number of frequent pattern generated from PDMFPM/normal FP

Abstraction Level	High	Middle	Leaf
Number of FP (PDMFPM)	3	11	54
Number of FP (Normal FP mining)	3	13	86

As mentioned above, two strategies are used here. The first strategy uses progressively deepening frequent pattern mining(PDMFPM) to mine FP, and same *is-a* relation is used as shown in Figure 2. The minimum support value for each level gradually decreases: $\sigma_{high} = 0.2, \sigma_{middle} = 0.05, \sigma_{leaf} = 0.005$[11]. Same support values are applied to second strategy: use FP mining without pruning possible uninteresting rules during generating process. Table 23 shows the number of FP mined by the two strategies. As can be seen, PDMFPM generates less Middle level FP and Leaf level FP compared with normal FP mining, this partly proves that PDMFPM can prune uninteresting FP during the mining process. As an example, a FP generated by normal FP mining in leaf level is shown below:

$$\{ColorB1, Ewhite1\}$$

However, we cannot find this FP in the results of leaf level PDMFPM results. Notice that in middle level results of PDMFPM, *Ewhite* do not appear in any FP. The antecedent of *Ewhite1*: *Ewhite* does not satisfy σ_{middle}, and has not been generated by PDMFPM. Therefore, according to Definition 3.1, descendants (uninteresting FP) of *Ewhite* are removed by PDMFPM.

Table 24: high level FP

PDMFPM
{'Ewhite'}
{'Color' 'Ewhite'}
{'Color'}

normal FP
{'Ewhite'}
{'Color' 'Ewhite'}
{'Color'}

Table 25: middle level FP

PDMFPM
{'ColorB'}
{'ColorI'}
{'ColorS"white'}
{'ColorS"ColorB'}
{'white"ColorB'}
{'ColorS"ColorI'}
{'white'}
{'colorB"ColorI'}
{'Special'}
{'ColorS'}
{'white"ColorI'}

normal FP
{'ColorI"white'}
{'ColorS"white'}
{'ColorS"ColorI'}
{'ColorB"ColorI'}
{'ColorS"ColorB'}
{'Special'}
{'ColorB"white'}
{'ColorB'}
{'ColorS'}
{'ColorS'}
{'white'}
{'Ewhite"white'}
{'ColorI'}

[11]Note that we use relative support here

Table 26: Leaf level FP

PDMFPM
{'ColorS2" ColorI1'}
{'ColorS1" white1" Special1'}
{'white3'}
{'white2" ColorS1'}
{'ColorI1" Special1'}
{'ColorB1" ColorB2'}
{'ColorI1" ColorS1" white1'}
{'ColorB1" white3'}
{'ColorS1" ColorB2'}
{'ColorB1'}
{'ColorB1" white1" Special1'}
{'white2'}
{'ColorB1" ColorS1'}
{'ColorI1" ColorB2'}
{'Special1'}
{'white2" white1'}
{'white2" Special1'}
{'ColorS2" Special1'}
{'white2" ColorS1" white1'}
{'ColorI1" ColorS1" Special1'}
{'ColorI1" ColorI1'}
{'ColorS2" ColorS1'}
{'ColorS1" white1'}
{'ColorS1" white3" white1'}
{'white3" white1'}
{'white1'}
{'ColorB1" Special1'}
{'ColorI1" white1'}
{'white2" white3'}
{'ColorB1" ColorI1" ColorS1'}
{'ColorS2" white1'}
{'ColorB2" white1'}
{'white1" Special1'}
{'white2" ColorI1'}
{'ColorB1" white1'}
{'ColorB1" ColorI1" Special1'}
{'ColorB1" ColorI1" white1'}
{'ColorI1" white3'}
{'white2" ColorB1'}
{'ColorI1'}
{'ColorI3'}
{'ColorS1" Special1'}
{'white2" ColorI1" white1'}
{'white2" ColorS2'}
{'ColorI1" ColorS1'}
{'ColorS1'}
{'ColorS2" ColorB1'}
{'ColorB1" ColorI1'}
{'ColorB1" ColorS1" white1'}
{'ColorS1" white3'}
{'ColorB1" ColorS1" Special1'}
{'ColorI1'}
{'ColorB2'}
{'ColorS2'}

In the leaf level FPs generated by PDMFPM, *Ewhite1, Ewhite2, Ewhite3, Ewhite5* are missed compared with leaf level FPs generated by normal FP. The reason is that their ancestor *Ewhite* is eliminated by PDMFPM when doing middle level mining, thus the descendants are removed as well.

normal FP
{'ColorS2" ColorI2'}
{'ColorB1" white3'}
{'white2" ColorS1'}
{'ColorI2" Special1'}
{'ColorB1" ColorB2'}
{'ColorS1" ColorB2'}
{'Special4'}
{'ColorS1" white3'}
{'Ewhite2" white1'}
{'ColorI2" ColorB2'}
{'white2" Special1'}
{'ColorS1" Ewhite1'}
{'Ewhite1" white1'}
{'ColorS2" ColorS1'}
{'Special2'}
{'ColorS1" white3" white1'}
{'Ewhite1'}
{'Special6" white1'}
{'Special1'}
{'white2" white3'}
{'Ewhite1" Ewhite2'}
{'ColorB2" white1'}
{'ColorI2" ColorS1" white1'}
{'ColorB1" Ewhite1'}
{'ColorB1" ColorI2" white1'}
{'ColorI2" white3'}
{'ColorI1'}
{'Ewhite1" white3'}
{'white2" Ewhite1" white1'}
{'ColorS1" Special1'}
{'ColorI2" Ewhite1'}
{'ColorI2" Ewhite2'}
{'Ewhite5" Ewhite1'}
{'ColorI2" ColorS1'}
{'ColorB1" ColorS1" white1'}
{'white3" white1'}
{'ColorI2'}
{'ColorB2'}
{'ColorS1" white1" Special1'}
{'white3'}
{'ColorB2" Ewhite1'}
{'ColorS1" Ewhite1" white1'}
{'white1" Special2'}
{'ColorB1'}
{'ColorI2" Ewhite1" white1'}
{'ColorB1" white1" Special1'}
{'white2'}
{'ColorB1" ColorS1'}
{'Ewhite5'}
{'white2" white1'}
{'Ewhite2'}
{'ColorS2" Special1'}
{'white2" ColorS1" white1'}
{'ColorS1" white1'}
{'Special5'}
{'ColorB1" Special1'}
{'white1'}
{'ColorI2" white1'}
{'ColorB1" ColorI2" ColorS1'}
{'ColorS2" white1'}
{'white1" Special1'}
{'white2" ColorI2'}
{'Special6'}
{'white2" ColorS1" Ewhite1'}
{'ColorB1" white1'}
{'ColorB1" ColorI2" Special1'}
{'white2" ColorB1'}
{'Ewhite1" Special1'}
{'Ewhite1" Ewhite2" white1'}
{'ColorS2" Ewhite1'}
{'ColorI3'}
{'ColorI2" ColorS1" Ewhite1'}
{'ColorS1" Ewhite2'}
{'white2" ColorI2" white1'}
{'ColorI2" ColorS1" Special1'}
{'white2" ColorS2'}
{'ColorI2" ColorI1'}
{'ColorS1'}
{'ColorS2" ColorB1'}

Our experiment result shows that compared with normal FP mining, PDMFPM effectively removes uninteresting FP without post analysis. Full set of generated rules are listed in Table 24, 25and 26.

Eliminate uninteresting FPs by gradually decreasing support value of each abstraction level. As explained in Section 4.1, using same support for multilevel FP mining has several disadvantages. Therefore, in this experiment, we use three strategies to validate the benefit of using gradually decreasing support value and show the drawbacks of using single support value for all levels. The first strategy uses PDMFPM with gradually decreasing support value: $\sigma_{high} = 0.2, \sigma_{middle} = 0.05, \sigma_{leaf} = 0.005$; the second strategy uses PDMFPM with single support value: $\sigma = 0.2$; the third strategy uses PDMFPM with single support value: $\sigma = 0.005$.

Table 27: number of FPs with different PDMFPM support value setting

Abstraction Level	High	Middle	Leaf
Number of FP (gradually decrease σ)	3	11	54
Number of FP (single σ_{high})	3	1	1
Number of FP (single σ_{leaf})	20	49	83

Table 27 shows our experiment results. With gradually decreasing σ, 3 high level FPs, 11 middle level FPs and 54 leaf level FPs are generated. However, when using single $\sigma = 0.2$, both middle level and leaf level have only 1 FP; when using single $\sigma = 0.005$, 20 high level FPs, 49 middle level FPs and 83 leaf level FPs are generated. From the numbers above, we know that different support value settings do effect FP generating.

One drawback of using single support value is that when setting the support value relatively high ($\sigma = \sigma_{high}$), low support patterns from leaf level are mostly ignored. For example, in Table 27, we notice that only 1 Middle level and 1 Leaf level FP are generated given $\sigma = \sigma_{high}$. Since FP in lower levels are not enough to analyze, only trivial patterns from high level can be found (e.g. $\{Color, Ewhite\}$).

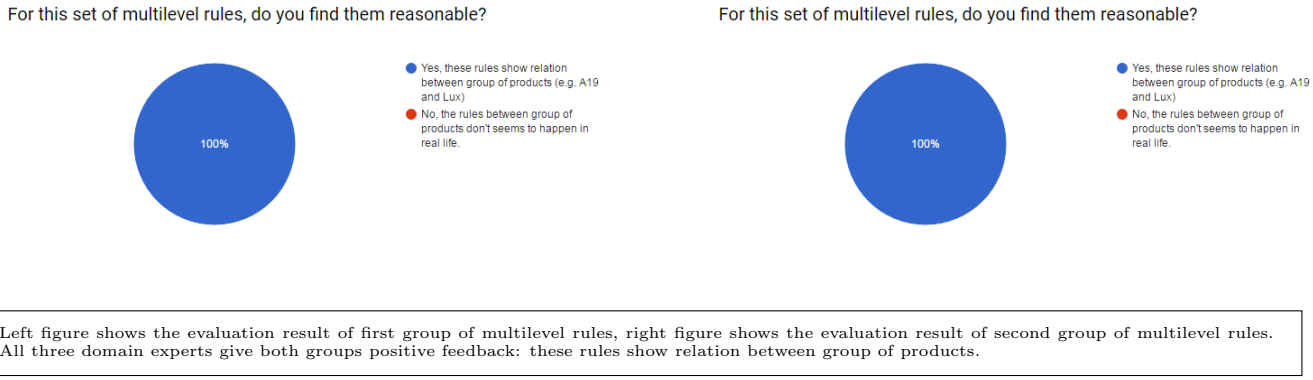
Another drawback of using single support value is that when setting the support value relatively low ($\sigma = \sigma_{low}$), many uninteresting FPs from higher levels are generated. For example, in Table 27, we notice that 20 High level FPs and 49 Middle level FPs are generated given $\sigma = \sigma_{leaf}$, which are far more than the corresponding results generated by gradually decreasing support value. Among these High level FPs, some are uninteresting to users. For example, $\{Color, White, Ewhite\}$ is uninteresting to users. The reason is obvious: relation among these three high level groups of lights cannot provide non-trivial information immediately to users as lower level FPs do. Further more, under this setting, more uninteresting rules are generated in High level, which result in uninteresting descendants generated at leaf level. For example: $\{white3, ColorS1, Ewhite1\}$ is generated because the antecedent of *white2*: *'White'* has not been removed at High level.

On the contrary, the results of PDMFPM using gradually decreasing σ show reasonable number

of rules generated in each level. Not that many FPs are generated in high level; enough amount of FPs are generated in middle and leaf levels for further analysis.

Our experiment shows that compared with single support value setting, gradually decreasing support value setting is superior, because it helps to shrink the size of high level FPs and to reserve enough FPs in middle level and leaf level.

Expert evaluation. Previous discussion in this section mainly focuses on how to effectively eliminate uninteresting patterns, and what the proper support threshold setting is. However, whether the generated multilevel association rules reflect relation between groups of products is not discussed. Therefore, we conduct domain expert evaluation to answer this question. Twenty multilevel association rules are selected from PDMFPM middle level rules and evenly divide into two groups for experts to evaluate.



Left figure shows the evaluation result of first group of multilevel rules, right figure shows the evaluation result of second group of multilevel rules. All three domain experts give both groups positive feedback: these rules show relation between group of products.

Figure 8: Domain experts evaluation of Multilevel AR. group 1 & group 2

Figure 8 shows the result of expert evaluation based on two selected groups of multilevel rules. Apart from this, full set of multilevel rules as shown in Table 25 are evaluated in terms of the ordering of rules. Since this is an optional task, two domain experts did the evaluation. One find the ordering is good, the other find the rules at beginning are not more interesting compared with the rules at end.

The answer of domain experts shows that multilevel association rule mining do reflect level-wise relation of products in real life, however ordering for multilevel ARs need to be improved.

3.5 Conclusion

To answer the first business question, we used association rule mining to find frequently purchased products combination. Basic association rule mining, progressively deepening multilevel associa-

tion rule mining, cross-level association rule mining were considered. How to select proper support and confidence thresholds was introduced first. We explained why the support and confidence thresholds are decided according to different contexts. Then we chose proper minimum support and minimum confidence mainly based on the number of ARs/FPs generated, and generated ARs/FPs by using the selected values. The generated ARs/FPs were studied thoroughly afterwards. We discussed the negative effects of dominating products and eliminated ARs influenced by dominating products. Different interest measures were discussed such as *kulc, max-confidence, imbalance ratio, cosine and lift, etc.*; proper measures were selected for our sample dataset. The selected measures: *max-confidence, lift, imbalance ratio* were used to filter the generated ARs. These generated ARs were presented to domain experts for evaluation. Both cross-level AR mining and PDMFPM were implemented. We conducted experiment of cross-level AR mining to show cross-level AR mining do discover 'invisible' ARs; in addition, the number of generated rules given different grouping strategies were discussed. Then we conducted two experiments of PDMFPM. First experiment aims to prove the superiority of PDMFPM compared with normal FP mining in terms of eliminating uninteresting rules. The results prove PDMFPM outperforms normal FP when eliminating uninteresting rules. Second experiment aims to prove using different support value for each level is a more reasonable approach for multilevel FP mining. Our results show that using a single minimum support may generate redundant high level FPs, or too few leaf level FPs, which proves that using different support value for multilevel FP mining is reasonable. Association rule mining and multilevel frequent pattern mining were applied on our sample dataset, which helped us to discover frequently purchased products combinations. The generated results were evaluated by domain experts. At least half of the association rules are said interesting by experts; some of the experts found association rules can contribute to decision making. As for multilevel FP results, all domain experts found the mined FP shows real life relation of products groups.

4 Are there any patterns reflecting purchase order of products?

Given the generated association rules in Section 3, what kind of products are purchased together frequently is clear. However, as mentioned in Section 1.1, when promoting products, we would like to know which products shall be promoted at first and which products shall be promoted later. Association rules cannot provide such information. Therefore, sequential pattern mining is introduced in this section, which aims to discover frequent purchased product sets in time ascending order. In short, frequent sequential patterns are patterns that:

1. appear frequently in the given dataset
2. are in ascending order in terms of purchase time

Such sequences help users to have a deep insight into the given dataset w.r.t. customer purchase order of certain products sets. In order to answer the second business question, we use sequential pattern mining to generate such frequent sequences. Further more, sequential pattern mining is used for predictive analysis. This way of using sequential pattern mining is demonstrated in previous research such as [Mobasher, *et al.*, 2002] and [Wright, *et al.*, 2015]. Since business sector is always interested in what the next purchase is, where next purchase implies a time based ordering, we would like to use mined sequential patterns for predictive analysis. In the following part of this section, the concept and definition of sequential pattern mining is introduced first, then we present how we use sequential pattern mining to predict the next purchase of customers based on their previous transactions (we see a plugin event as a purchase in a system); the result of multi-dimensional frequent pattern mining is integrated to our predictive analysis.

4.1 Sequential Pattern Mining

We first introduce the basic concept of sequential pattern mining. Let $I = (i_1, i_2, ..., i_m)$ be an itemset, where i_j is an item. Let $s, s = \langle s_1, s_2, ..., s_n \rangle$, be a sequence ordered temporally, where s_j is an itemset. All the transactions of a customer can be listed in a sequence with increasing time order, where each transaction is an itemset. We call such a sequence a *customer-sequence*.

As introduced by Agrawal dan Srikant [1995], we say a sequence $s = \langle s_1, s_2, ..., s_n \rangle$ is contained in another sequence $S = \langle S_1, S_2, ..., S_n \rangle$ if there exist integers $i_1 < i_2 < ... < i_n$ such that $s_1 \subset S_{i_1}$, $s_2 \subset S_{i_2}$, ..., $s_n \subset S_{i_n}$. For example, sequence $\langle (a), (a, b), (e, g) \rangle$ is contained in sequence $\langle (c), (a, d), (a, b, c), (e, f, g) \rangle$, as $(a) \subset (a, d)$, $(a, b) \subset (a, b, c)$ and $(e, g) \subset (e, f, g)$. In contrast, sequence $(a), (b)$ is not contained in (a, b), because *a, b* are purchased in two transactions in first sequence. However in the second sequence, *a,b* are purchased in one transaction.

Similar to the support definition in frequent pattern mining, we say a *customer-sequence* supports a sequence s if s is contained in the *customer-sequence* of this customer. The absolute support of a

sequence is the total number of *customer-sequence* that supports this specific sequence. Therefore, the problem of sequential pattern mining is to find the sequences that satisfy the minimum *support* threshold from all sequences.

We extend the concept of association rule to sequential pattern mining, which we call sequence rule. We define a sequential rule as follows: for a specific sequence s, an sequential rule is an implication of the form $X \Rightarrow Y$, where $X \subset s$, $Y \subset s$, $X \cup Y = s, X \cap Y = \emptyset$. The definition of confidence value is similar to the definition in Equation 3. In this study, we restrict the definition of sequential rule as follows: for the RHS of a sequential Y, Y is the last itemset/transaction in a sequence. In Section 4.1.1, we explain this in detail.

4.1.1 Process of predictive analysis

In this subsection, we present our framework for predictive analysis. We first propose three objectives that interest us; then the implementation of this analysis is introduced step by step; finally we discuss how we evaluate the quality of prediction.
Three objectives are defined for this analysis.

1. Check the usefulness of sequential pattern mining for predicting customer purchase behavior of connected lighting system products.

2. How the completeness of a customer purchase history affects predicting ability.

3. How the length of transaction record influences prediction ability.

We propose these three objectives for the following reasons: Objective 1 is proposed because we want to know whether sequential pattern mining is a proper predictive method for purchase behavior. Objective 2 is proposed because purchase records of customers, being used as training set, are not always complete, since information may lost at any steps during data collection phase. Therefore, to identify the degree of influence that lost record brings, we propose objective 2. Objective 3 is proposed because customers have different lengths of purchase sequence, we want to know whether longer purchase sequence (history) can better reflect purchase behavior.

We design three approaches to investigate the objectives above following the idea of [Wright, *et al.*, 2015] to set up the experiment. For all approaches, 10-fold cross validation has been used to evaluate the quality of prediction, which also divides the dataset into a training set and a test set. This helps to assess how the results of this approach generalize to the whole transaction dataset. As introduced in section 4.1, the restriction of definition for RHS, we divide each generated frequent sequence to an antecedent-consequent pair, in which the antecedent is the previous transaction records (a list of product sets) of a customer and the consequent is the last transaction record of a customer. For ease to write, we say that the antecedent of a sequence is the LHS and the consequent of a sequence is the RHS.

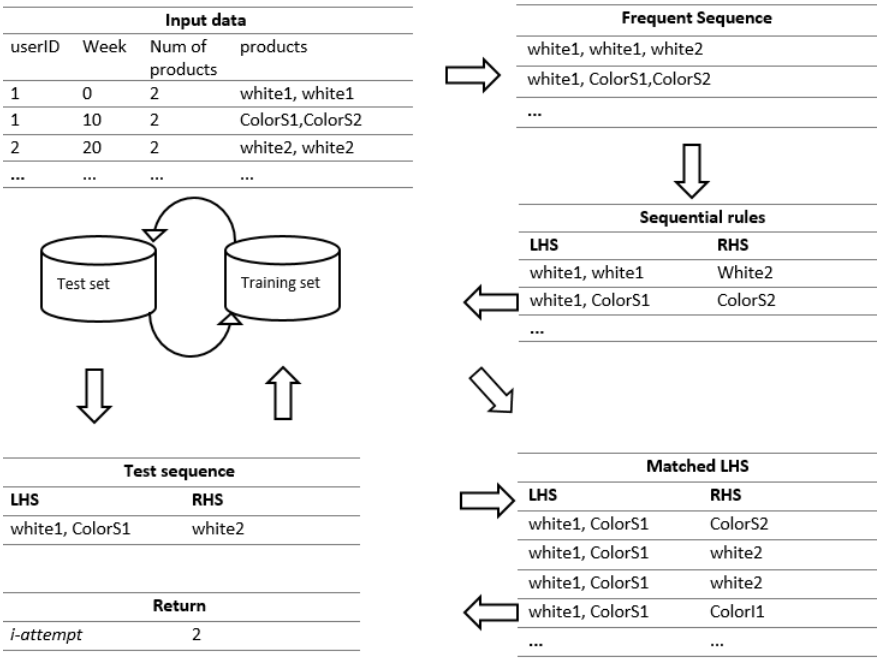


Figure 9: Experiment flow of sequential pattern mining for predictive analysis

Figure 9 shows the basic flow of our experiment. For the first objective, the approach is described as follows:

1. For each sequence in test set, we match the sequences in training set whose LHS is same as the LHS of the test sequence and extract matched sequences.

2. The weighted support-confidence value is calculated for all matched sequences. Then sort the sequences according to weighted support-confidence value.

3. Traverse the extracted matched sequences, find those matched sequences whose RHS is same as the RHS of the test sequence. If RHS match happens at the first ordered sequence, we call it *1-attempt*. Similarly, if RHS match happens at the i^{th} ordered sequence, we call it $i - attempt$.

4. To evaluate how successful the prediction is, number of each *i-attempt* is summed. Then the percentage of successful prediction given *i-attempt* is calculated.

For the second objective, the approach is described as follows:

1. Do the first 3 steps in the previous approach.

2. For each test sequence, We see a match of RHS is successful if this match happens within 3-attempts. Calculate the total number of successful matches; calculate percentage of successful prediction.

3. Delete the first item from test sequence's LHS, resulting in a length reduced LHS. Do the first 2 steps again.

4. Finish this process after 4 iterations.

For the third objective, the approach is described as follows:

1. Do the first 2 steps in approach 1.

2. When doing the third step in approach 1, *i-attempt* as well as the LHS length of test sequence is returned.

3. For every LHS length returned, successful rate of prediction is calculated.

We illustrate the experiment setting in Section 4.4 and discuss the result in Section 4.5.

4.2 Literature Review on Sequential Pattern Mining

Sequential pattern mining is another important branch in pattern mining research territory, which looks for ordered items/events in a dataset. This technique is first introduced in [Agrawal dan Srikant, 1995], aiming to solve the problem of finding all sequential patterns from the given transaction database. Many other sequential pattern algorithms are proposed later on, for example GSP [Srikant dan Agrawal, 1996], [Han, *et al.*, 2001] and SPADE [Zaki, 2001]. These algorithms mainly focus on improving the efficiency of the mining process. To expand the usage of sequential pattern mining, other algorithms are introduced such as multi-dimensional sequential pattern mining [Pinto, *et al.*, 2001], constraint based sequential patter mining [Zaki, 2000], etc.

Based on the algorithms above, sequential pattern mining is applied to different real life applications. For example, transaction data analysis [Srikant dan Agrawal, 1996], DNA sequence analysis, stock market analysis , banking chrun analsis [Chiang, *et al.*, 2003], etc. Wright, *et al.* [2015] introduce a predictive analysis for diabetes therapy based on sequential pattern mining. They see each patient's medication records as a sequence and try to predict the next medication by using the previous medication sequence. The result of their study shows that sequential pattern mining is fairly effective in next medication prediction. Since our product transaction data is organized in a similar way as the medication record sequence described above, we want to verify whether sequential pattern mining based predictive analysis is an effective method for our problem. The intuitive idea of this study is that if the transaction history of two customers are the same, the future purchase is likely the same.

4.3 Data wrangling

For ease of understanding and according to the terminology in sequential pattern mining, we use *customer* to represent *connected lighting system*, *transactions* to represent *products plug in*; we do not mention every detail of the dataset here, only the useful features are presented.

In our dataset, every transaction of each customer is a record. This record contains customer IDs and timestamps of the transaction. Given these information, we can organize our data of each customer as a *customer-sequence, s*, where for each s_j in s, s_j represents j^{th} transaction of a customer along with the timestamp. We store the newly generated data into a new dataset and use this dataset as the input of predictive analysis. This process of data is detailed in the following. In the pre-processed dataset, each user's record is stored in a nested list where each sub-list is a user's change-record. From that dataset, *user ID, week number* and *product* information are extracted. As shown in Table 28, the extracted information is organized as follows: each line in this dataset is a change-record, we also call it a transaction. The first column is the userID, which indicates the user that the transaction belongs. The second column is the week, during which the transaction happens. The third column is the number n of products involved in this transaction, and it is followed by n the fourth column, which includes n products.

Table 28: Example of input data for predictive analysis

userID	week	num of products	products
1	0	2	white1, white1
1	10	2	ColorS1, ColorS2
2	20	2	white2, white2
3	4	3	ColorI12, ColorB1, white2

4.4 Experimental setting

Based on the techniques introduced above, in this section, the process of this predictive analysis is explained. There are four major steps in all three experiments: frequent sequence mining, construct matched LHS dataframe, match RHS, evaluation of results. For each step, certain experiment settings are applied. We introduce the experiment setting of all experiments together, since the major part of these experiments are the same. For the experiment 1 and 3, only the evaluation methods are different; for experiment 2, an additional step is described.

Generate frequent sequence. We use the R package *arulesSequences* [12] developed by Christian Buchta and Michael Hahsler. This package provides the implementation of algorithm *cSPADE* introduced in [Zaki, 2000]. *cSPADE* is an improved version of algorithm *SPADE*, which not only generate ordered sequences but also consider a variety of syntactic constraints [Zaki, 2001]. The algorithm is briefly explained in Appendix A.2. This algorithm is chosen because several constraints brought by it can be very useful to solve our problem, which is introduced Appendix A.2

[12]https://cran.r-project.org/web/packages/arulesSequences/index.html

and in this subsection. Given the input data as described in Section 4.3, and the following setting: $\sigma_{min} = 1e^{-4}, \varphi_{min} = 0.2, maxsize = 1$, we generate the frequent sequences to be analyzed. In total 1656 frequent sequences are generated, 1061 left after filtered by φ_{min}. The setting: $maxsize = 1$ is one of the constraints we mentioned before. This constraint limits the maximum number of products in an itemset, which is 1 in this case. Formally, let $, s = \langle s_1, s_2, ..., s_n \rangle$ be a sequence ordered temporally, where $s_j, 1 \leq j \leq n$ is an itemset. By applying this constraint, the maximum length of s_j is 1. In another word, s_j can only be an item. This is applied because we assume that the multiple-item itemsets may increase the complexity of the frequent sequences that contain them, thus the number of matched LHS may decrease.

Construct matched LHS dataframe. After the frequent sequences are generated, for each sequence, both LHS and RHS are extracted. Note that, as explained before, RHS is the last itemset of the sequence. Since we restrict the number of items in itemset, RHS is actually the last item of the sequence. For extracted LHS, the duplicate products are removed. For example: a LHS $\langle \{white1\}, \{white1\}, \{ColorS1\} \rangle$ is transformed to $\langle \{white1\}, \{ColorS1\} \rangle$. We do this because we are only interested in the relation between different items for now. Then we divide the dataframe into 10 folds, with even number. The first fold is used as the test set and the rest are the training set. For each test sequential rule in the test fold, its LHS is compared with the LHS of sequential rules in training set. When LHS matches, the sequential rules in training set are added to the *matched LHS dataframe* of the test sequential rule. In addition, the *matched LHS dataframe* is ordered by a weighted support-confidence value of each rule.

Match RHS. Given *matched LHS dataframe* of a specific sequential rule, we are able to check if RHS of this test sequential rule appears in this dataframe. The process is simple, the *matched LHS dataframe* is traversed to see if any RHS in this dataframe matches the RHS of test sequential rule. As introduced in Section 4.1.1, the *i-attempt* is returned for further analysis.

Gradually decrease the length of transaction record. For experiment two, an additional step is required to check how losing history transaction records may affect prediction. As briefly described in Section 4.1.1, we do previous steps the same as in experiment 1 three times, except that each time an item is eliminated from the LHS of sequential rules in *matched LHS dataframe*. Then the *matched LHS dataframe* after elimination is used for RHS matching, and return the *i-attempt*.

4.5 Results

As described in Section 4.1.1, three experiments are designed to verify our objectives. In this section, we present experiment result and discuss the result according to our objectives.

4.5.1 Experiment results: example of generated sequences

We present a set of example sequential patterns here. The setting to generate these sequences are stated in section 4.4, **Generate frequent sequence**.

Table 29: Example Sequential Patterns

sequential pattern	support (relative)
{ColorS1}-{white1}	0.020
{Ewhite1}-{Ewhite2}	0.019
{white1}-{ColorI2}	0.019
{ColorS1}-{ColorS1}	0.017
{Ewhite1}-{ColorS1}	0.016
{white2}-{Ewhite1}	0.015
{white2}-{ColorS1}	0.012
{white1}-{white1}-{ColorS1}	0.012
{Ewhite1}-{white3}	0.012
{white1}-{ColorS1}-{ColorS1}	0.012
{ColorB1}-{white1}	0.011
{Ewhite1-white1}-{white1}	0.010
{white1}-{ColorS1}-{white1}	0.010
{Ewhite1}-{ColorS2}	0.009
{white1}-{ColorB2}	0.009
{ColorS1}-{white3}	0.009

In total 1656 frequent sequences are generated from our dataset. As state in Section 4.4, this set of sequences contains only 1-item itemsets for later predictive analysis. Although this is a subset of all frequent sequences, it is enough to answer our second business question: Are there any patterns reflecting purchase order of products?

4.5.2 Experiment results: usefulness check of sequential pattern mining for predictive analysis

For experiment 1, we want to know whether sequential pattern mining is useful when dealing with customer purchase behavior problem, especially this connected lighting system products. This can be done by calculating the percentage of successful prediction given different *i-attempt* threshold. As mentioned before, the *i-attempt* is recorded for each sequential rule from the test set.[13] Given different *i-attempt* threshold (e.g. 1, 2, 3, 4, 5), we calculate how many of the *i-attempt* are made within each of these thresholds. Then divided by the total number of sequential rules in test set. This value is the percentage of successful prediction with different *i-attempt* threshold.

[13]For the cases that no matched LHS is found and the case no matched RHS is found, FALSE is returned.

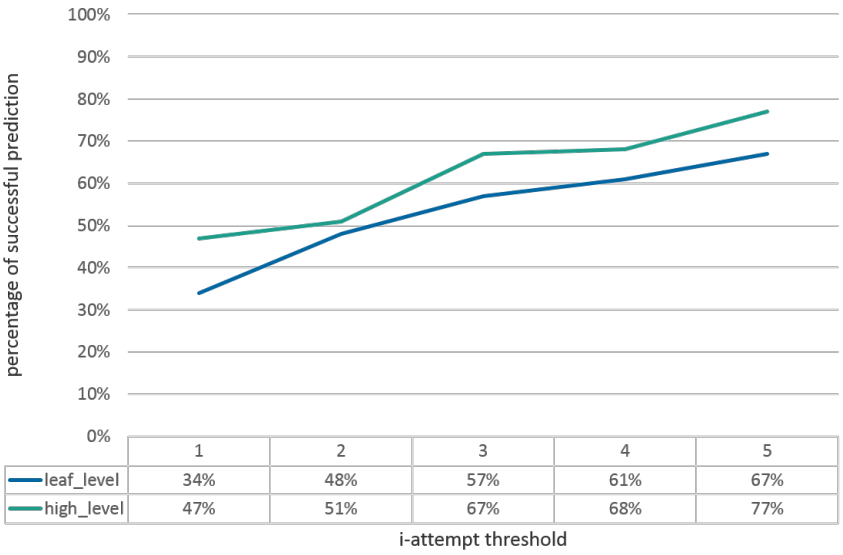


Figure 10: Experiment 1 result: percentage of successful predictions

Figure 10 shows the percentage of successful prediction given 5 different thresholds: 1, 2, 3, 4, 5. As mentioned before, we apply this experiment in two datasets: the original dataset and the dataset, in which items have been abstracted to a high level. Only the original dataset's result will be discussed, namely the *leaf_level* (green line) in this graph. As shown in the graph, when the *i-attempt* threshold is 1, 34% of predictions are successful. The successful rate increases steadily along with the increase of $i - attempt$ threshold. When *i-attempt* reaches 5, the successful rate stops at 67%. This means that given a customer transaction record $T = (t_1, t_2, ..., t_n)$, if we are able to find a customer whose transaction records (exclude last transaction) is the same as T. We have 34% chance to correctly predict the next purchase of this customer by only using the first transaction record, in which the last transaction is our prediction. If this threshold is increased to 5, the successful rate is also increased to 67%. In addition, by using 10-folds validation, we find these results quite stable with a float range no more than 3%. In addition, when adding multilevel pattern mining, the successful rate of prediction has a good amount of increase at each threshold, although not significant. This might because the varieties of groups are less at higher level, thus the range of potential prediction is not as wide as the leaf level.

From the results above, we find the successful prediction rate is a bit low when the *i-attempt* threshold is low. However, when the *i-attempt* threshold is high enough, the result is quite reliable. This implies that when making recommendation only based on the result of sequential pattern mining, suggesting more possible products to users may increase the effect of recommendations.

4.5.3 Experiment results: influence of purchase record integrity and purchase record length

As explained in Section 4.4, the experiment is conducted to support objective 2 and 3. Figure 11 and Figure 12 show the result of leaf level experiment and high level experiment respectively. We mainly explain the result in Figure 11.

The left, middle, and right bar in each group of bars represent the LHS with 2 items, 3 items and 4 items respectively. First group of bars represent the prediction result when the LHS have not been modified. Second group of bars represent the result after the first item is eliminated. Similarly, the third group and fourth group of bars represents the results after first two and first three items are removed. Note that in the third and fourth bar groups, the result of 2-item LHS is not shown in third bar group and the result of 2-item and 3-item are not shown in the fourth bar groups. This is because after first two items are removed, length of 2-item LHS is zero, thus can not match test 2-item LHS and no result can be generated.

In general, the successful prediction rate is higher in the first and second bar groups compared with third and fourth bar groups. For example, for LHS with 4 items, when LHS are complete (no record is deleted) 77% of predictions are successful; when the first item is removed, still 77% of predictions are successful. However, when the first two items are removed, successful rate decreases to 61%. The successful rate is even lower when first three items are removed, only 15%. Similar to 4-item LHS, the successful rate of 3-item LHS also decrease along with the shrinking of training LHS, from 61% to 18%. For 2-item LHS, the result also meets a decrease from 42% to 36%. Our experiment shows the best result can be obtained when the training LHS are complete; the percentage of successful rate decrease along with the removal of history records. This answers our second objective: the completeness of a customer purchase history do affect the predicting ability of sequential pattern mining.

From the result above, it is clear that the completeness of history record do affect the ability of prediction. We infer the main reason as follows: when the history record (training set LHS) is complete (e.g. the length of LHS is n, the total number of length n LHS is m), the test LHS can find enough history record to compare and match. However, when the history record is incomplete (e.g. first 2 items are deleted, the length of LHS is $n-2$ and the number of LHS is $m-j$, where $j \geq 0$), the number of matched LHS may decrease. Thus success rate will fall. Therefore, when using this approach to solve predictive problems in our connected lighint system dataset, we suggest using complete history record. However, this solution may bring another problem: too long a test LHS sequence can hardly match any same training LHS. Because customers buy products differently, and for any two long records, the chance of match is very low. Two solutions are proposed to solve this problem: we can calculate the similarity between test LHS and the training LHS and find the most similar training LHS for RHS matching; another solution is to use a shorter but more informative pattern, which is introduced in Section 5.

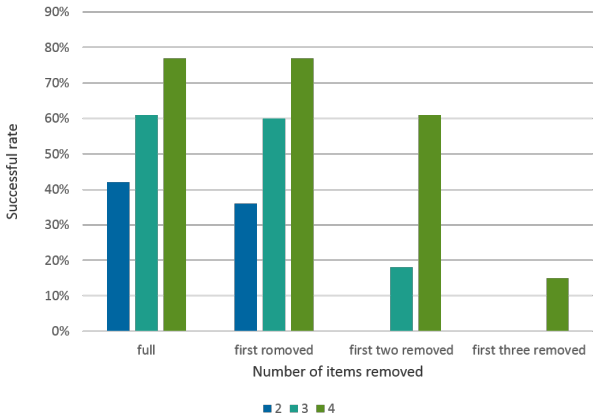


Figure 11: success rate of leaf level

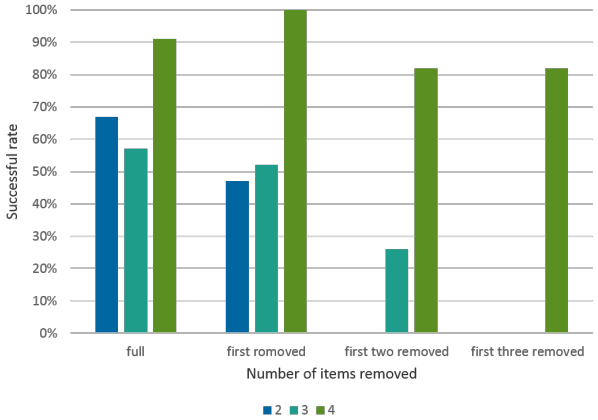


Figure 12: success rate of high level

From another perspective, we investigate how the length of LHS [14] affects prediction successful rate. In Figure 11, for 2-item LHS, 42% predictions are successful; for 3-item LHS, 61% predictions are successful; for 4-item LHS, 77% predictions are successful. Similar trend is shown in the result of second bar groups, where the successful rates are 36%, 60% and 77% for 2-items LHS, 3-items LHS and 4-items LHS respectively. In Figure 12, although the first bar groups (complete LHS), shows few fluctuations that 2-item LHS performs better than 3-items LHS, the second bar groups still shows increasing success rate trend from less items LHS to more items LHS (2-item LHS to 4-item LHS). These results shows that in our experiment, longer LHS can bring better prediction results. Since we only choose sequential patterns length from 3 to 5, longer LHS is not validated to this observation.

According to the discussion above, it is clear that the successful rate of prediction is related to the length of sequential pattern. Long sequential patterns have bigger chance to precisely predict the next purchase compared with the shorter patterns. Therefore, when predicting a customer's next purchase, enough history records shall be used to increase the prediction successful rate. However, this shares similar problem with the previous discovery: too long a LHS can hardly match other LHS. To solve this problem, a proper LHS length n shall be found, where *n-item* LHS perform best among all the different length LHS. This will be part of our future work as explained in Section 6.

4.6 Conclusion

In this study, we introduced sequential pattern mining to answer the business question: Are there any patterns reflecting purchase order of products. We first introduced the motivation of investigating this business question, and explained how sequential pattern mining suits this question. Then,

[14] Length of training LHS and testing LHS are the same

sequential pattern mining was introduced in detail, where extension and restriction of original definition were introduced as well. Three predictive analysis objectives were proposed as extensions of the original business question. Approaches to realize these objectives were introduced respectively. Afterwards, implementation of each approach was illustrated. By using frequent sequential pattern mining, we conducted the predictive experiments, in which we conducted the first experiment to verify objective 1, and second experiment to verify objective 2, 3. The results of first experiment shows that sequential pattern mining can be used for predictive analysis, however the prediction quality varies according to the *i-attempt* threshold. We also suggested the possibility to collaborate sequential pattern mining with other data mining techniques for further use. The result of second experiment shows that the loss of history records does affect prediction quality. In addition, the length of LHS also affects the prediction successful rate, where long sequences are more likely to output successful prediction. In the end, we discussed the drawback of sequential patterns. Firstly, it only reserves the order information, and users cannot get more temporal information from sequential patterns. Secondly, customer sequences might be very long, which might negatively influence the successful rate of prediction. The result sequences shown in Section 4.5.1 answer the second business question. The predictive analysis extends the business questions in terms of potential to make recommendations.

5 Are there any patterns reflecting time span between purchase behavior?

In Section 3 and Section 4, frequent pattern mining, association rule and sequential pattern mining are introduced thoroughly. Frequent pattern mining helps to find itemsets that appear frequently in a given dataset. Based on the discovered frequent patterns, we are able to generate association rules which satisfy φ_{min}. By applying statistical interest measures, we are able to find rules that are interesting and reasonable. However, generated association rules barely provide temporal information. Popular AR algorithms such as Apriori [Agrawal, *et al.*, 1993], TreeProjection [Agarwal, *et al.*, 2001], FP-tree [Han, *et al.*, 2000] only extract information of items in transaction records, thus temporal information is ignored. In our study, since we concatenate every transaction of each customer as a whole record, the products in our frequent patterns can happen at any time within the analyzed year. Therefore, we cannot say the items in frequent patterns we found are purchased in same transaction. Some multidimensional frequent pattern mining algorithms and proposals [Kamber, *et al.*, 1997], [Intan, 2007] do consider additional information other than items. For example, Kamber, *et al.* [1997] introduce *metarule* to find multidimensional AR that can compile with certain metarules; Intan [2007] uses fuzzy labels to generate meaningful multidimensional AR. However, since most of these multidimensional algorithms treat temporal information as one of the dimensions, only time point is considered. As an example:

$$\forall x \in person, purchased(x, white1) \wedge month(x, September) \Rightarrow possess(x, ColorS1)$$

The rule above states that for a person x, if she purchased *white1* in *September*, she is likely to have *ColorS1* in her system. This is a typical multidimensional association rule which provides time information. However, it only concerns the time of purchase.

To integrate other temporal information in pattern mining, we introduce sequential pattern mining for predictive use. The items in a sequence are organized in time ascending order, we assume customers who share a same history transaction sequence will behave similarly in the future. Based on this assumption, we conduct our predictive analysis. Although using sequential pattern mining for predictive analysis shows reasonable result, the shortcoming is obvious. 1. For long test sequence, it is hard to match a same sequence from history record. 2. Only order of products is reserved, other temporal information is lost. According to the discussion above, we need a pattern mining technique which generates such patterns that:

1. the items order is reserved ;
2. the sequence should be shorter and informative ;
3. other temporal information is integrated.

For example, we would like to see a pattern which shows a certain combination of products; preferably the products are organized in a kind of order; this pattern satisfies such kind of constraints

that all products in this pattern shall happen within a certain time window.

Given the requirements above, we find frequent episodes mining especially suitable for our problem. The result of frequent episodes mining provides information of products purchase order, time window of this pattern, the time gap between consecutive purchase in a pattern, etc. These information are crucial supplements of association rule mining and sequential pattern mining. For example, one of the issue we would like to know is what products are frequently bought after a certain starter product within a certain time window. We find it hard to answer this issue by using association rule mining or by using sequential pattern mining, because both techniques cannot provide full information (time window, purchase order, pattern frequency). However, a frequent episode is able to answer such a question, since the mined episode happens in a certain time window, satisfying user set σ_{min}, can have specific order of purchase. Therefore, we introduce frequent episodes mining to answer this question.

5.1 Frequent Episodes Mining

In frequent episodes mining, the event sequence is used as input data. For each event in the event sequence, the time of occurrence is attached to this event. An example of an event sequence is:

$$(A, 2), (B, 3), (C, 6), (C, 7), (A, 7), (B, 7), (E, 9), (F, 14), (C, 14), (A, 15)$$

In the example above, *A, B, C, E, F* are event types (in our case, the event type is a purchase of certain light); the number associated with a event is the time when that event occurs. The problem here is to find the frequent episodes in the event sequence. Mannila, *et al.* [1995] define an episode as: partially ordered sets of events. We use directed acyclic graphs to show three type of episodes:

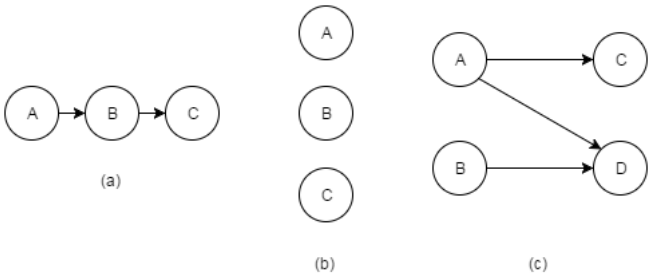


Figure 13: three type of episodes

Episode (a) is a serial episode, in which each event *A, B* and *C* occurs in time ascending order and happens within a certain time window[15]. Episode (b) is a parallel episode, in which *A, B* and

[15]Events not necessarily happen one next to other

C can occur in any order and happen within a certain time window. In episode (c), A and B can occur in any order, C and D the same. However, D can only happen after A and B, C can only happen after A. Therefore, possible event sequences satisfy (c) can be:

$$A, B, C, D$$

$$A, C, B, D$$

In this study, we want to find such event sequences that satisfy certain preset time window and following the definition of serial episode as mentioned above. We give definitions of *event sequence, episode, frequent episode* as follows:

Event Sequence: given a set of event types $\mathcal{E}$, an *event pair* is (e, t), where $e \in \mathcal{E}$ and t is the associated time of e, we call $S = (e_1, t_1), (e_2, t_2), ..., (e_n, t_n)$ an *event sequence*, if $e_i \in \mathcal{E}$ and $T_s \leq t_i < T^s$ for all $i = 1, ..., n$, where T_s, T^s are starting time and ending time respectively. In addition, for all $i = 1, ..., n-1$, $t_i \leq t_{i+1}$.

Episode An episode $(V, \leq, g)$ is a set of nodes V, $\leq$ a partial order on V, and a mapping $g : V \to \mathcal{E}$ which maps each node in the episode to an event type. We say an episode serial if $\leq$ is a total order; we say an episode parallel if $\leq$ is trivial. An episode can also be nether serial nor parallel. Given an episode $(V, \leq, g)$ and an event sequence $S = (e_1, t_1), (e_2, t_2), ..., (e_n, t_n)$, an occurrence of this episode on S is a map $h : V \to \{1, ..., n\}$ where $g(v) = \mathcal{E}_{h(v)}$ for $\forall v \in V$, for $\forall v, w \in V$ with $v \leq w$ we have $t_{h(v) < t_{h(w)}}$. In addition, we say an episode $\mathcal{B} = (V_\mathcal{B}, <_\mathcal{B}, g_\mathcal{B})$ is a sub-episode of episode $\mathcal{A} = (V_\mathcal{A}, <_\mathcal{A}, g_\mathcal{A})$, if all event types of $\mathcal{B}$ are also in $\mathcal{A}$; the order of events in $\mathcal{B}$ is the same in $\mathcal{A}$.

Frequent episode We say an episode is frequent if the number of the occurrences of this episode exceeds a certain minimum threshold in a given dataset. In our study, we discover non-overlap frequent episode [Laxman, *et al.*, 2005] from a sample dataset. Two occurrences of an episode is called non-overlap if there is no event in one appears in between the other. For an episode, the total number of appearances of non-overlap occurrences is the frequency of this episode. In our study, we find episodes, of which frequency exceeds the minimum threshold which is set by user, in addition, the episodes satisfy certain temporal constraints (e.g. time window).

5.2 Literature Review on Frequent Episode Mining

Frequent episodes mining is a subfield of temporal data mining, which concerns mining hidden relations between data with temporal dependencies [Antunes dan Oliveira, 2001]. Two important branches in this field are sequential pattern mining and frequent episodes mining. In Section 4, we have extensively discussed sequential pattern mining, which is used for predictive analysis in our work. Here, we introduce frequent episodes mining, which not only reserves order information, but also other temporal information. Temporal pattern mining is first introduced by Mannila, *et al.*

[1995]. The study aims to find episodes that show behavior and actions of the users and corresponding systems. Two areas that they focus in the study are telecommunications network monitoring and empirical user interface study, because the log from these systems provide a huge amount of data with temporal information. In [Mannila, *et al.*, 1995], an Apriori-like algorithm *WINEPI* is introduced to find frequent serial/parallel episodes in a given dataset. Many other studies extend the frequent episodes mining in various directions. Harms, *et al.* [2002] introduce *MOWCATL*, which find frequent patterns that precede the occurrence of certain patterns in other sequences with user-specified time lag in between. The algorithm finds such patterns in both single sequence and multiple sequences. An example of such a rule can be described as: for customers in a given dataset, if light A and light B are purchased within a week, then after maximum one month's time, light C and light D will be purchased within a week. In [Huang dan Chang, 2008], another two algorithms *MINEPI+* and *EMMA* are introduced. These two algorithms enable episodes mining on complex sequence (intervals can be hours, days, weeks, etc.). Another work by Tatti dan Cule [2012] extends the concept of closed frequent pattern to closed frequent episodes by introducing the concept of strict episodes. Based on frequent episodes mining algorithms, many applications use this techniques to solve real life problems. In [Ang, *et al.*, 2013], frequent episodes mining is used to proactively predict the occurrence of drifts in peers in distributed classification problems. In [Leemans dan van der Aalst, 2014], the author integrates frequent episodes mining in process mining to discover patterns in an event log. In [Clifton dan Gengo, 2000], frequent episodes mining is used in *Intrusion Detection* to identify false alarms from those systems.

5.3 Data wrangling

Based on the input data for sequential pattern mining described in Section 4.3, we modify the dataset to make it suitable for frequent episodes mining and the business problem. As shown in Table 28, each transaction's products information is recorded in column 3 and 4: *num of products* and *products* respectively. What we need to do is to extract start kit from every week-0 transaction record and rename it as the corresponding start kit. For example, the first row of Table 28, two products *white1, white1* are purchased at week 0. We look up the start kit list and find this is an start kit *white1, white1*, thus naming and grouping them as *white*1_*start*. This process is applied to every record in this dataset. The example of after process dataset is shown in Table 30.

Table 30: Example of input data for frequent episodes mining

userID	week	num of products	products
1	0	2	*white1_start*
1	10	2	ColorS1, ColorS2
2	0	2	*white2_start*
3	4	3	ColorI2, ColorB1, white2

As can be seen in row 0 and row 2, the starter products are shown in column *products* (*white1_start*, *white2_start*). This format of data is used as the input data for frequent episodes mining.

5.4 Experiment setting

5.4.1 Objective

In this section, we introduce our objective of this experiment: find *after-start* purchase. People who have connected ligthing systems always have a starter kit, which can include different kinds of products. The product combination in starter kits can be very different, ranging from various products. However, starter kits are not user customized, only limited number of combinations are provided. We assume that customers may buy different starter kit for some reasons (e.g. price, usage). These starter kits can be very basic ones, for example, starter kits containing only *white2*. However, can also contain products with more functionalities. After having used for a period of time, the customer may want to buy another product that is compatible to the connected lighting system. Our study aims to find the products to be bought given a specific time window. Formally, given all event sequences $\mathcal{S} = s_1, s_2, ..., s_n$, $\forall s_j \in \mathcal{S}$, the first event is one kind of start kit $k_j \in \mathcal{K}$. $\mathcal{K} = k_1, k_2, ..., k_m$ is the set of starter kit. We want to find a set of frequent serial episodes such that for all episodes in this set, the first event is one kind of starter kit; the span of all episodes in this set satisfy a user specific time window.

As an example, given event sequences below, a user specific time window 3, minimum frequency threshold $\sigma = 2$:

$$(k_1, 1), (e_2, 4), (e_3, 5), (e_1, 8)$$
$$(k_{10}, 4), (e_6, 5), (e_2, 5), (e_1, 10)$$
$$(k_1, 2), (e_3, 2), (e_1, 2), (e_4, 2), (e_2, 4)$$

We say (k_1, e_2) is a frequent serial episode, since the number of occurrence of this episode is 2, which exceeds minimum frequency threshold, and k_1 happens before e_2 in every corresponding event sequence. Another episode (e_3, e_1) is frequent serial episode as well. However, as mentioned before, only episodes whose first event is a starter kit are considered. Therefore, (e_3, e_1) is not further considered.

5.4.2 Approach

In this section, we introduce the approach to find after-start purchase. As explained before, frequent episodes mining can help us to find episodes which satisfy σ_{min} and happen within a certain time window. These generated episodes are very close to what we expect from our sample dataset. Therefore, we abstract our problem to the problem of frequent episodes mining. The approach is simple and straightforward as follows:

1. Transform customer records to customer event sequences.
2. Given user specific time window and σ_{min}, mine frequent serial episodes from the customer event sequences.

3. Exclude mined frequent episodes without a starter kit.

4. Return rest episodes.

Since frequent episodes mining is unsupervised , we ask domain experts to evaluate whether these episodes are interesting and useful for company decision making. The result of experiment is detailed in Section 5.5.

5.4.3 Experiment implementation

We use TD-Miner [16] in this experiment to help generating frequent episodes, which is provided by Debprakash Patnaik. This is a handy software to mine frequent episodes, in which the number of products in episodes, time window and other variable can be set by user.

Two major steps are conducted in this experiment. First, we transform the previous transaction dataset to a starter kit dataset as described in Section 5.3. Second, we use the generated starter kit database as the input of frequent episodes mining. Given a preset minimum frequency σ_{min}, a pre-defined time window and the number of products in episodes, qualified frequent episodes are generated. With these frequent episodes, we can answer the business question proposed before. Since this is a rather unsupervised approach, we mainly rely on the expert evaluation of mined episodes to justify whether this approach is useful.

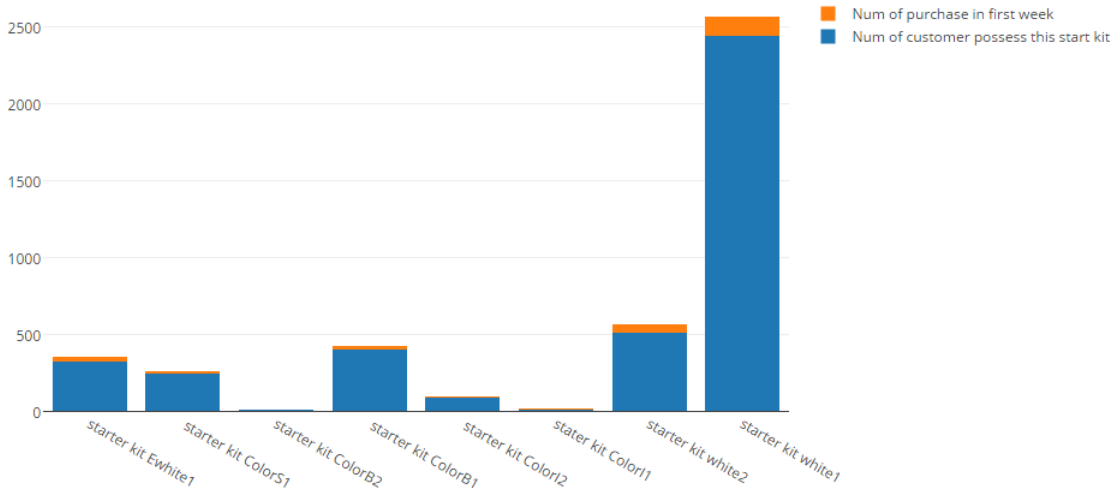
5.5 Results

In this section, descriptive results are shown to present the nature of our sample dataset in terms of frequent episodes.

5.5.1 Nature of starter kit dataset

There are 27 kinds of fixed combination starter kits, and even more kinds of free combination starter kits. However, only eight kinds of starter kits exist in our sample dataset. They are *starter kit Ewhite1, starter kit ColorS1, starter kit ColorB1, starter kit ColorB2, starter kit ColorI1, stater kit ColorI2, starter kit white2, starter kit white1.* Among these eight starter kits, not surprisingly *starter kit white1* appears most (2440 times), which means more than half of all customers in this dataset chose *starter kit white1* to begin. In contrast, only 6 customers chose *stater kit ColorB2* to begin. For other starter kits, 318 customers chose *starter kit Ewhite1*, 244 customers choose *starter kit Color1*, 402 customers choose *starter kit ColorB1*, 86 customers choose *starter kit ColorI2*, 10 choose *starter kit ColorI1* and 512 customers choose *starter kit white2*. Figure 14 and Figure 15 present more descriptive analysis results w.r.t. customer purchase history.

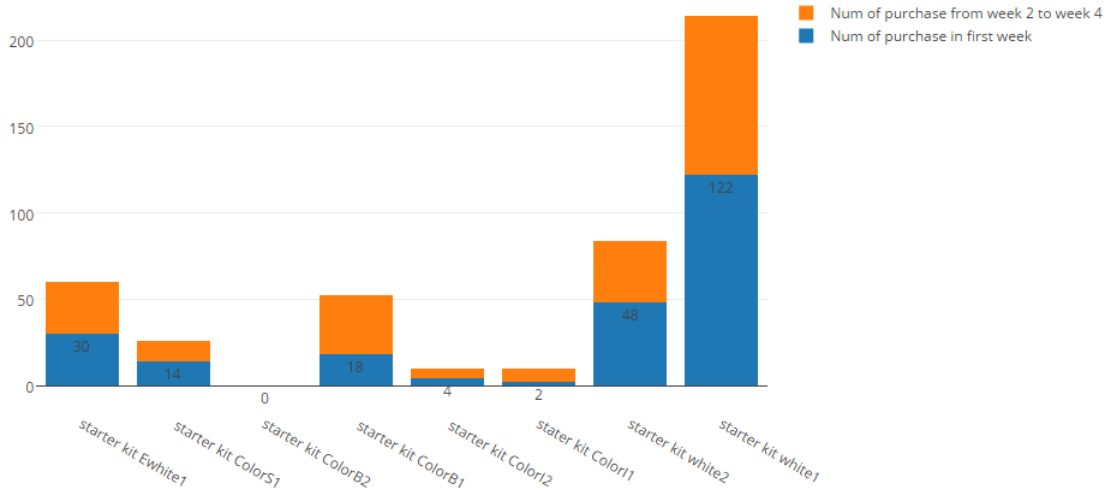
[16]https://github.com/patnaikd/tdminer



starter kit	num of starter kit	num of purchase in week 1
starter kit Ewhite1	318	30/9.4%
starter kit ColorS1	244	14/5.7%
starter kit ColorB2	6	0/0
starter kit ColorB1	402	18/4.5%
starter kit ColorI2	86	4/4.7%
statrer kit ColorI1	10	2/20%
starter kit white2	512	48/9.4%
starter kit white1	2440	122/5%

For each bar in this plot, the blue (bottom) part represents total number of certain starter kit in sample dataset; the orange (top) part represents number of customers (possess certain starter kit) who have bought new products in first week after purchased starter kit. As can be seen, Percentage of customers who purchased new products ranges from 4.5% to 9.4%, which means most customers chose not to purchase new products immediately after they purchased starter kit, only few chose the opposite. 20 percent of customers who have bought *starter kit ColorI1* immediately purchased new products, however the cardinality is too little. Therefore, we conclude that the majority customers do not purchase new products in the first week after they engaged.

Figure 14: Number of each starter kit compared with number of purchase in week 1



starter kit	purchase in week 1	purchase within week 2-4
starter kit Ewhite1	30	30/0
starter kit ColorS1	14	12/-14.3%
starter kit ColorB2	0	0/0
starter kit ColorB1	18	34/89%
starter kit ColorI2	4	6/50%
starter kit ColorI1	2	8/300%
starter kit white2	48	36/-25%
starter kit white1	122	92/-24.6%

For each bar in this plot, the blue(bottom) part represent the number of customers (possess certain starter kit) who have bought new products in the first week after engaged; the orange(top) part represent the number of customers (possess certain starter kit) who have bought new products within week 2 to 4. Number of customers (possess *starter kit white1*) who choose to purchase new products reduced 24.6%, even if the given time window 3 weeks is three times compared with 1 week. However, customers who have Color starter kits such as: *starter kit ColorB1, starter kit ColorI2, starter kit ColorI1* behave differently. For example, number of customers(possess *starter kit ColorB1*) who have bought new products from week 2 to week 4 increased 89% compared with the corresponding number in week 1. Although not validated, this plot shows that customers who have *Color starter kit* (*ColorB2, ColorB1, ColorI2, ColorI1* belongs to color category) are more likely to continue purchase compared with customers who have *white1 starter kit, white2 starter kit.*

Figure 15: Number of purchase in week 1 compared with number of purchase within week 2-4

5.5.2 Generated frequent episodes

The descriptive results above implies mainly two issues: 1. Few customers purchase new products right after they first engaged. 2. Different starter kits do affect further purchase behavior of customers. In this subsection, the results of frequent episodes mining are shown in Table 31. These generated episodes are presented to domain experts for evaluation.

Table 31: Mined frequent episodes

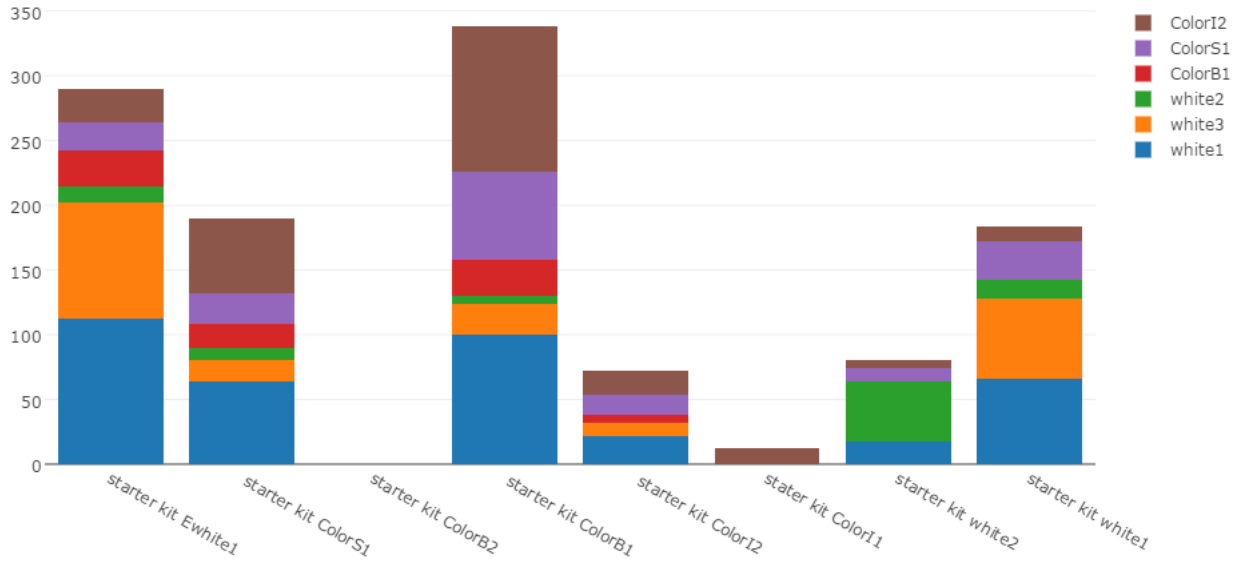
Episodes	Support
starter kit ColorB1-ColorI2	112
starter kit Ewhite1-white1	106
starter kit ColorB1-white1	100
starter kit Ewhite1-Ewhite1	78
starter kit ColorB1-ColorS1	68
starter kit ColorS1-white1	64
starter kit ColorS1-ColorI2	58
starter kit white1-white1	48
starter kit white1-Ewhite1	38
starter kit white2-white2	34
starter kit white1-ColorS1	30
starter kit ColorB1-ColorB1	28
starter kit Ewhite1-ColorB1	28
starter kit Ewhite1-ColorB2	28
starter kit ColorS1-ColorB2	26
starter kit Ewhite1-ColorI2	26
starter kit white1-Ewhite1-Ewhite1	24
starter kit ColorB1-Ewhite1	24
starter kit ColorS1-ColorS1	24
starter kit ColorI2-white1	22
starter kit ColorB1-ColorB2	22
starter kit Ewhite1-ColorS1	22
starter kit white1-white1-white1	18
starter kit white2-white1	18
starter kit ColorI2-ColorI2	18
starter kit ColorS1-ColorB1	18
starter kit ColorI2-ColorS1	16
starter kit ColorS1-Ewhite1	16
starter kit white1-white2	14

Episodes	Support
starter kit white1-ColorS1-ColorS1	12
starter kit white2-white2-white2	12
starter kit Ewhite1-Ewhite1-Ewhite1	12
starter kit white1-ColorI2	12
starter kit ColorB1-ColorI1	12
starter kit ColorB1-ColorI2 0	12
starter kit Ewhite1-white2	12
starter kit white2-ColorS1	10
starter kit ColorI2-Ewhite1	10
starter kit ColorS1-white2	10
starter kit Ewhite1-Ewhite5	10
starter kit white1-ColorB2	8
starter kit ColorS1-Special2	8
starter kit ColorS1-ColorA1	8
starter kit Ewhite1-ColorI2 0	8
starter kit white1-white2-white2	6
starter kit white2-Special2-Special2	6
starter kit white2-white2-white1	6
starter kit Ewhite1-white1-white1	6
starter kit white2-Special2	6
starter kit white2-ColorI2	6
starter kit ColorI1-ColorI2	6
starter kit ColorI2-ColorB1	6
starter kit ColorB1-white2	6
starter kit ColorB1-ColorA1	6
starter kit ColorB1-Ewhite5	6
starter kit ColorS1-ColorI2	6
starter kit Ewhite1-ColorI1	6

In total, 57 frequent episodes are mined from our sample dataset, where 7 starter kits appear. These frequent episodes satisfy minimum support threshold 5; all the episodes span within 4 weeks; in addition, every episode contains one starter kit. *starter kit ColorB1* appears 11 times in this set, which is the highest. whereas *starter kit Monet* appears only once. The most frequent episode is $starterkitColorB1, ColorI2$, presenting 112 times.

5.5.3 Analysis of frequent episodes

Based on the generated frequent episodes shown above, a descriptive study is conducted to find how starter kit may influence later purchase behavior. We choose 6 popular products to analyze, which are *ColorI2, ColorS1, ColorB1, white2, white3, white1.* All eight starter kits that appear in our dataset are considered. For each of these starter kit, total number of each product purchased after this starter kit is calculated. Note that, all these purchases happen from week two to week four, which implies the purchases happen very close to the engagement of connected lighting systems. Figure 16 shows the result of analysis based on generated frequent episodes.



In total, six lights are analyzed in this descriptive study, in which three *Color* lights and three *white* lights are considered; eight starter kit are analyzed, where one *Ewhite* starter kit, five *Color* starter kit and two *white* starter kit are considered. Note that, *Ewhite* lights have similar functionalities as *white* lights have.

As can be seen in the graph above, customers who have bought starter kit *Ewhite1* made around 200 purchases of *white2, white3, white1*; customers who have bought starter kit *white2* made more than 80% purchases of *white2, white1*; customers who have bought starter kit *white1* made around 70% purchases of *white1, white2, white3*. However, percentage of purchased *white* lights never exceed 50% in systems start with *Color* starter kits. This result implies that people who have bought *white, Ewhite* starter kits tend to buy *white* lights compared with *Color* lights within the first 4 weeks.

In contrast, customers who have bought starter kit *ColorS1* made around fifty purchases of *ColorI2*; customers who have bought starter kit *ColorB1* made more than a hundred purchases of *ColorI2* and around sixty purchases of *ColorS1*; customers who have bought starter kit *ColorI1* made almost all purchases of *ColorI2*. This result implies that customers who have bought *Color* starter kits tend to buy more *Color* lights from week two to week four. In addition, *ColorI2* is the most popular product among all the products we examined.

Result of starter kit *ColorB2* is lost because no products listed above are purchased by customers who have this starter kit.

Figure 16: Number of each kind of products purchased for each group of customers(with same starter kit)

5.5.4 Expert evaluation

Twenty frequent episodes are selected from Table 31, and evenly divided to two groups. The evaluation of first group of frequent episodes is shown in Figure 17. The evaluation of the second group is shown in Figure 18.

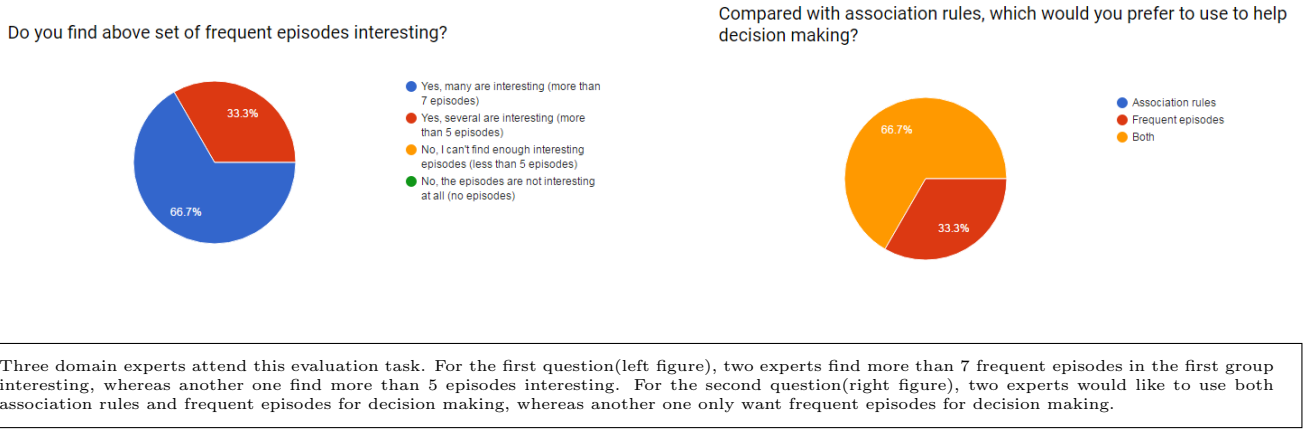


Figure 17: Domain experts evaluation of frequent episodes group 1

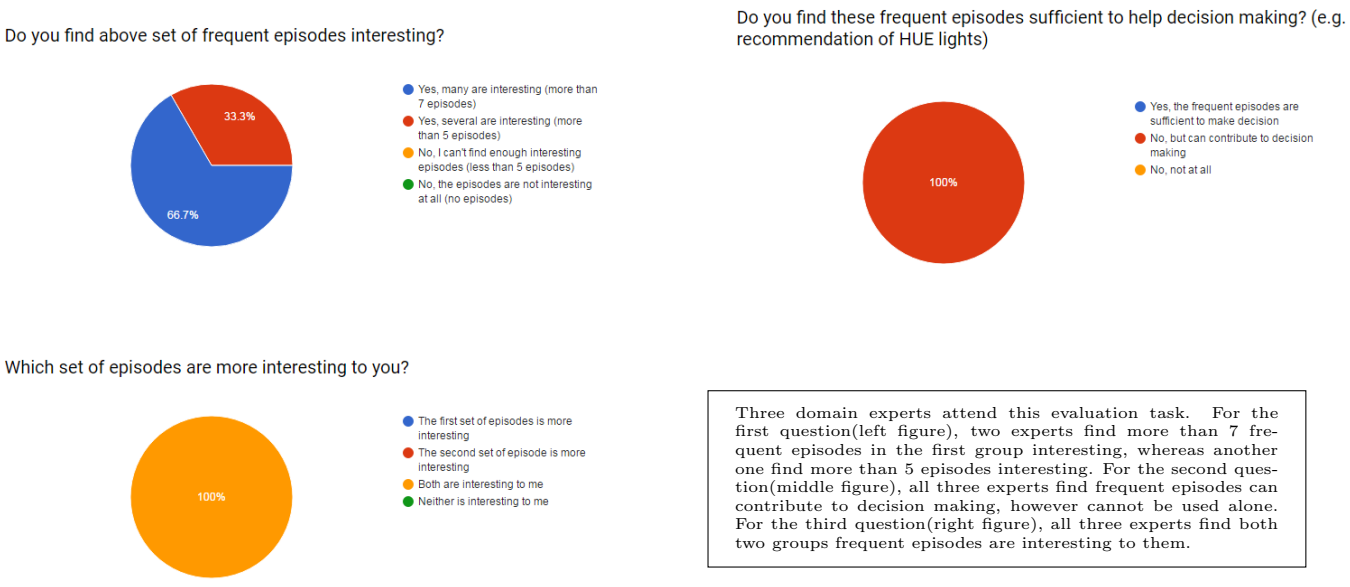


Figure 18: Domain experts evaluation of frequent episodes group 2

Apart from the evaluation above, we also ask whether the high support frequent episodes appear more often compared with low support episodes in real life customer purchase behavior. All three experts give positive answer to this question.

The evaluation results show that many of mined frequent episodes are interesting; business sector is more likely to use frequent episodes for decision making instead of association rules, however they still find frequent episodes cannot support decision making alone; the support of frequent episodes do reflect real life purchase behavior.

5.6 Conclusion

In this study, we used frequent episodes mining to answer the business problem: *What kind of products are frequently purchased in order, given a specific time window?*. The limitation of association rule mining and sequential pattern mining was stated at the beginning to explain the motivation of introducing episodes mining. Furthermore, the superiority of frequent episodes mining when dealing with temporal information was detailed. Then we discussed popular algorithms to generate frequent episodes and some related applications. Next, we proposed the objectives to discover based on the business question; the approach and implementation of the experiment were discussed as well. We transform the original sample dataset to a starter kit dataset for frequent episodes mining. From the analysis of transformed starter kit dataset, two main observations are: 1. Few customers purchase new products in the first week they engaged. 2. Customers with different starter kits behave differently in future purchase in terms of the willing to buy new products. The generated episodes satisfy $\sigma = 5$ and span within 4 weeks; every episodes starts with a certain starter kit. Since the frequent episodes are descriptive, these episodes were presented to domain experts for evaluation. The evaluation results show that domain experts are satisfied with the result of frequent episodes mining in 3 aspects: 1. Many of the frequent episodes are interesting to them. 2. Compared with association rules, they are more willing to use frequent episodes for decision making 3. Domain experts find frequent episodes interesting mainly because these episodes provide additional temporal information.

6 Conclusion

In this study, three business questions of interest were answered. We introduced association rules to answer the first question: *What kind of products are frequently purchased together?* Sequential pattern mining is introduced to answer the second question: *What is the purchase order of frequently purchased products*? Frequent episodes mining is introduced to answer the third question: *What kind of products are frequently purchased in order given a specific time window*? Since association rule mining and frequent episodes mining are descriptive data mining techniques, we presented the generated result of both techniques to domain experts for evaluation. Although sequential pattern mining is descriptive as well, it is used and evaluated predictively.

To answer the first business problem, three sub-techniques were introduced to answer the problem from different perspectives. They are basic association rule mining, progressively deepening multilevel association rule mining and cross-level association rule mining. The first technique generates basic association rules, the second technique generates level-wise association rules and the third technique generates cross-level association rules. To filter the generated rules, statistical measures were applied to eliminate uninteresting rules. In addition, the result of progressively deepening multilevel association rule mining shows that this technique effectively eliminates uninteresting rules. We also showed that in multilevel pattern mining, different support values for different levels are more reasonable compared with single support value. As for future extension, although association rule mining is a descriptive data mining technique, it is possible to integrate it as a feature to other popular data mining techniques (e.g. classification).

To answer the second business problem, two sub-experiments were conducted by using sequential pattern mining. In the first experiment, we aim to verify whether sequential pattern mining can be used predictively in this context. In the second experiment, we aim to verify whether losing history records influences the sequential pattern mining's ability of predictive analysis; this experiment also verifies whether the length of customer purchase record influences the predicting ability of this technique. The result of experiment 1 proves the ability of sequential pattern mining to do predictive analysis. The result of experiment 2 shows that losing history record does influence the predicting ability and the length of customer purchase record also influences the successful rate of prediction. However, our results were generated from the modified sample dataset, in which only 1-product transaction records are considered and the duplicated products are eliminated. This modification reduces the complexity of sequences and shortens some sequences. In other words, the modification reserves part of the sequences' information however some are ignored. As a future work, we would like to apply same experiment on original sequences for predictive use, which should be more convincing compared with current results.

To answer the third business problem, frequent episodes mining was introduced. The dataset was modified for a more specific objective: mining frequent episodes including starter kit. Two main observations from the modified dataset are: 1. Few customers purchase new products in the first

week after engagement. 2. Starter kits do influence the future purchase behavior of customers. Afterwards, frequent episodes were generated given specific constraints. As for future work, apart from the descriptive use, we also want to frequent episodes predictively by integrating with other data mining techniques, which is a supplement in terms of temporal information.

The results of expert evaluation also prove that the business questions are well answered. The result answer of first business question shows that experts find most of the association rules interesting, and the association rules can at least contribute to decision making. This evaluation implies that the mined association rules can help to design product bundles. The result answer of third business question shows that experts find most mined frequent episodes interesting. They also shows interest to use frequent episodes for decision making use. This evaluation implies that the mined frequent episodes can help to recommend products within certain time windows. In addition, some domain experts find the results of this study give a confirmation to what she/he is working on; others suggest temporal reasoning shall be emphasized. They also provided certain situations where they would like to use these results, for example, specific bundle offering and specific product communication adjustments, understanding consumer journey and product recommendation.

As mentioned before, we mainly used association rule mining and frequent episodes mining as descriptive tools to answer business questions, which rely on domain experts to do post evaluation. However, this requires basic knowledge of data mining and pattern mining, which restricts the range of potential users of our results. Therefore, in the future, we would like to improve the frame-work to enable users make decision directly from the output results. This can be achieved by integrating other supervised data mining techniques, and present informative corresponding visualization. As for sequential pattern mining, we did the predictive analysis only on simplified dataset. Therefore, future work should also concentrate on using real customer purchase sequences to do predictive analysis.

From business perspective, we answered business questions w.r.t. products bundle, order of products promotion and time window on products promotion. However, more business questions can be answered. For example, apart from knowing time window of certain set of products, business sector may also want to know the maximum/minimum purchase interval between consecutive products. In addition, if we are given usage records of systems at hour granularity, customer usage behavior patterns can be mined. This type of pattern helps us to know customers' behavior in terms of how they use products, which can be used to improve the functionalities of the products.

A Algorithms

A.1 fast frequent pattern mining algorithm FP-growth

FP-growth is a fast frequent pattern mining algorithm proposed in [Han, *et al.*, 2000] and improved in [Han, *et al.*, 2004]. Different from Apriori like algorithms [Agrawal, *et al.*, 1993], [Borgelt, 2003], this algorithm does not have candidate generation phase, which needs large memory space for huge amount of candidate sets. The author proved that this approach outperforms other frequent pattern mining approaches [Agrawal, *et al.*, 1993],[Agarwal, *et al.*, 2001] at that time. Győrödi, *et al.* [2004] and Kumar dan Rukmani [2010] also show the superiority of FP-growth in their application and research respectively. Although efficiency is crucial to the scalability of applications, however, since our sample dataset is relatively small and the result of different algorithm is same (given same minimum support and minimum confidence), we do not emphasize the importance of efficiency in our study, thus putting more effort on applying the algorithm to solve our problem.

In this chapter, we describe the algorithm in detail. In general, FP-growth consists of two phases: FP-tree generation and frequent pattern mining from conditional database. Here, the conditional database is a kind of projected database each associated with a frequent item. The definition of FP-tree by Han, *et al.* [2000] is shown below:

FP-tree

1. FP-tree have one root node: 'null', a set of item prefix sub-trees as children of root, a frequent-item-header table.

2. Each node in the item prefix sub-tree have three fields: item *item-name, count, node-link*. *Item-name* is the item represented by the node, *count* is the number of transactions represented by the portion of the pat reaching this node, and *node-link* is the link to the next node in carrying the same *item-name*, or null if there is none.

3. Each entry in the *frequent-item header table* consists of two fields: *item-name, head of node-link*. The *head of node-link* points to the first node in FP-tree carrying the entry's *item-name*

Given the definition of FP-tree above, the algorithm used in *Orange3-Association* introduced by Han, *et al.* [2004] to construct FP-tree is shown below:

Algorithm to construct FP-tree

Input: transaction database: D; Minimum support: σ
Output: FP-tree

1. Scan D once. Find all the frequent items satisfy σ, sort them in support descending order and store in L.

2. Create the root of FP-tree (T) as 'null'. Do the following for each transaction *Trans* in D:

   Sort the frequent items in *Trans* according to the order in L. Let the sorted frequent item list in Trans be$[p|P]$, where p is the first element and P is the remaining list. Call function insertTree($[p|P]$): If T has a child N such that *N.item-name =p.item-name* , then increment *N's* count by 1; else create a new node N, and set the count as 1, its parent link be linked to T, node-link to the nodes with same *item-name* via the node-link structure. If P is not empty, call insertTree($[P, N]$) recursively.

The generated FP-tree is further used for frequent pattern mining, described as follows:

Algorithm to mine frequent patterns

Input: a FP-tree, transaction database D, minimum support σ
Output The complete set of frequent patterns.

call FP-growth(*FP-tree,null*)
Procedure FP-growth(*Tree, A*)
{
    if *Tree* contains a single path P **then**
        for each combination (B) of the nodes in the path P **do**
            generate pattern $B \cup A$ with *support = minimum support of nodes in B*
    else for each a_i in the header of the *Tree* **do**
    {
        generate pattern $B = a_i \cup A$ with *support* = a_i.*support*;
        construct B's conditional database and B's conditional *FP-tree: TreeB*;
        if $TreeB \neq \emptyset$
            call FP-growth(*TreeB, B*);
    }
}

A.2 cSPADE

As already introduced in section 4.1, many algorithm for frequent sequential mining are proposed to improve the efficiency of sequential pattern mining [Agrawal dan Srikant, 1995],[Srikant dan Agrawal, 1996],[Han, *et al.*, 2001],[Zaki, 2001]; some other algorithms provide solutions to sequential pattern mining where constraints are in need [Garofalakis, *et al.*, 1999],[Zaki, 2000],[Pei, *et al.*, 2007].

In this study, we use R package *arulesSequences* provided in [Hahsler, *et al.*, 2011], which implement the algorithm *cSPADE*. This sequential pattern mining algorithm provides various constraints

which facilitate our experiment. Although efficiency is not our main concern, *cSPADE* out perform *GSP* from a factor of 2 to more than an order of magnitude [Zaki, 2001]. The main reason of this superiority is that *cSPADE* use a efficient way to do support counting. Compared with the conventional *horizontal* layout, *cSPADE* use a *vertical* database layout, where each item is associated with its *idlist* (containing sequence id and event id). Based on this, by performing temporal join on subsequences, the support count can be determined. However, this approach generates large amount of intermediate *idlists*, which are too heavy for main-memory when the dataset is extremely large. Therefore, Zaki [2001] proposed *suffix-based equivalence classes* to split the large search space into small ones, which can be processed independently. This means each *suffix-based equivalence classes* has all information in need to generate frequent sequences that have same suffix. This approach allows processing each *suffix class*[17] independently, thus less intermediate sequences are generated and requirement of memory is lower. The algorithm itself is shown as follows:

SPADE(min_sup):
$\mathcal{P} = \{\text{parent classes } P_i\}$
for each parent class P_i **do** Enumerate-Frequent(P_i)

Enumerate-Frequent(S):
    for all sequences $A_i \in S$ **do**
        $T_i = \emptyset$
        for all sequences $A_j \in S$, with $j > i$ **do**
            $R = A_i \cup A_j$
            $\mathcal{L}(R) = \text{Temporal-Join}(\mathcal{L}(A_i), \mathcal{L}(A_j))$
            if$(\sigma(R) \geq min_{sup})$ **then** $T = T \cup \{R\}$ print R;
        Enumerate-Frequent(T)
    delete S

In this algorithm, $\mathcal{L}$ represents the *idlist*. For example, $\mathcal{L}(X)$ is the *idlist* of item X. As explained in Section 3.1, $\sigma(R)$ is the support value of sequence rule R. For detailed illustration of *SPADE*, please refer to [Zaki, 2001].

Moreover, we use the constraint *maxsize* to limit the size of each event/transaction. Integrating this constraint in *SPADE* is straightforward. In the algorithm above, we only need to check whether $maxsize(R) \leq max_w$, where max_w is the constraint threshold for maximum number of items in an event/transaction. Since we haven't use other constraints incorporated in *cSPADE* (e.g. length of sequence, time gap between transaction, maximum time window between any two transaction, etc.), how to integrate these constraints is not detailed here.

[17]Suffix of a sequence is the sub-sequence excluding certain items on the left side. e.g. $A \rightarrow CD \rightarrow E$ and $B \rightarrow CD \rightarrow E$ share a same suffix $CD \rightarrow E$

References

Agarwal, R. C., Aggarwal, C. C. and Prasad, V. (2001). "A tree projection algorithm for generation of frequent item sets." *Journal of parallel and Distributed Computing* **61**. 350–371

Agrawal, R., Imieliński, T. and Swami, A. (1993). "Mining association rules between sets of items in large databases." *ACM SIGMOD Record* **22**. 207–216

Agrawal, R. and Srikant, R. (1995). "Mining sequential patterns." IEEE. 3–14

Aloysius, G. and Binu, D. (2013). "An approach to products placement in supermarkets using prefixspan algorithm." *Journal of King Saud University-Computer and Information Sciences* **25**. 77–87

Ang, H. H., Gopalkrishnan, V., Zliobaite, I., Pechenizkiy, M. and Hoi, S. C. (2013). "Predictive handling of asynchronous concept drifts in distributed environments." *IEEE Transactions on Knowledge and Data Engineering* **25**. 2343–2355

Antunes, C. M. and Oliveira, A. L. (2001). "Temporal data mining: An overview.". 13

Bennett, J. and Lanning, S. (2007). "The netflix prize.". 35

Borgelt, C. (2003). "Efficient implementations of apriori and eclat."

Cao, H., Jiang, D., Pei, J., He, Q., Liao, Z., Chen, E. and Li, H. (2008). "Context-aware query suggestion by mining click-through and session data." ACM. 875–883

Chiang, D.-A., Wang, Y.-F., Lee, S.-L. and Lin, C.-J. (2003). "Goal-oriented sequential pattern for network banking churn analysis." *Expert Systems with Applications* **25**. 293–302

Clifton, C. and Gengo, G. (2000). "Developing custom intrusion detection filters using data mining." IEEE. 440–443

Garofalakis, M. N., Rastogi, R. and Shim, K. (1999). "Spirit: Sequential pattern mining with regular expression constraints.". 7–10

Győrödi, C., Győrödi, R. and Holban, S. (2004). "A comparative study of association rules mining algorithms."

Hahsler, M., Chelluboina, S., Hornik, K. and Buchta, C. (2011). "The arules r-package ecosystem: analyzing interesting patterns from large transaction data sets." *Journal of Machine Learning Research* **12**. 2021–2025

Han, J. and Fu, Y. (1999). "Mining multiple-level association rules in large databases." *Knowledge and Data Engineering, IEEE Transactions on* **11**. 798–805

Han, J., Kamber, M. and Pei, J. (2011). "Data mining: concepts and techniques." Elsevier

Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U. and Hsu, M. (2001). "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth.". 215–224

Han, J., Pei, J. and Yin, Y. (2000). "Mining frequent patterns without candidate generation." ACM. 1–12

Han, J., Pei, J., Yin, Y. and Mao, R. (2004). "Mining frequent patterns without candidate generation: A frequent-pattern tree approach." *Data mining and knowledge discovery* **8**. 53–87

Harms, S. K., Deogun, J. and Tadesse, T. (2002). "Discovering sequential association rules with constraints and time lags in multiple sequences." Springer. 432–441

Huang, K.-Y. and Chang, C.-H. (2008). "Efficient mining of frequent episodes from complex sequences." *Information Systems* **33**. 96–114

Intan, R. (2007). "A proposal of fuzzy multidimensional association rules." *Jurnal Informatika* **7**. pp–85

Kamber, M., Han, J. and Chiang, J. (1997). "Metarule-guided mining of multi-dimensional association rules using data cubes..". 207

Kumar, B. S. and Rukmani, K. (2010). "Implementation of web usage mining using apriori and fp growth algorithms." *Int. J. of Advanced Networking and Applications* **1**. 400–404

Laxman, S., Sastry, P. and Unnikrishnan, K. (2005). "Discovering frequent episodes and learning hidden markov models: A formal connection." *IEEE Transactions on Knowledge and Data Engineering* **17**. 1505–1517

Leemans, M. and van der Aalst, W. M. (2014). "Discovery of frequent episodes in event logs." Springer. 1–31

Linden, G., Smith, B. and York, J. (2003). "Amazon. com recommendations: Item-to-item collaborative filtering." *Internet Computing, IEEE* **7**. 76–80

Mannila, H., Toivonen, H. and Verkamo, A. I. (1995). "Discovering frequent episodes in sequences extended abstract."

Mobasher, B., Dai, H., Luo, T. and Nakagawa, M. (2001). "Effective personalization based on association rule discovery from web usage data." ACM. 9–15

Mobasher, B., Dai, H., Luo, T. and Nakagawa, M. (2002). "Using sequential and non-sequential patterns in predictive web usage mining tasks." IEEE. 669–672

Omiecinski, E. R. (2003). "Alternative interest measures for mining associations in databases." *IEEE Transactions on Knowledge and Data Engineering* **15**. 57–69

Pei, J., Han, J. and Lakshmanan, L. V. (2001). "Mining frequent itemsets with convertible constraints." IEEE. 433–442

Pei, J., Han, J. and Wang, W. (2007). "Constraint-based sequential pattern mining: the pattern-growth methods." *Journal of Intelligent Information Systems* **28**. 133–160

Pinto, H., Han, J., Pei, J., Wang, K., Chen, Q. and Dayal, U. (2001). "Multi-dimensional sequential pattern mining." ACM. 81–88

Sandvig, J. J., Mobasher, B. and Burke, R. (2007). "Robustness of collaborative recommendation based on association rule mining." ACM. 105–112

Srikant, R. and Agrawal, R. (1996). "Mining sequential patterns: Generalizations and performance improvements." Springer

Tan, P.-N., Kumar, V. and Srivastava, J. (2002). "Selecting the right interestingness measure for association patterns." ACM. 32–41

Tan, P.-N., Kumar, V. and Srivastava, J. (2004). "Selecting the right objective measure for association analysis." *Information Systems* **29**. 293–313

Tatti, N. and Cule, B. (2012). "Mining closed strict episodes." *Data Mining and Knowledge Discovery* **25**. 34–66

Wong, K. W., Zhou, S., Yang, Q. and Yeung, J. M. S. (2005). "Mining customer value: From association rules to direct marketing." *Data Mining and Knowledge Discovery* **11**. 57–79

Wright, A. P., Wright, A. T., McCoy, A. B. and Sittig, D. F. (2015). "The use of sequential pattern mining to predict next prescribed medications." *Journal of biomedical informatics* **53**. 73–80

Wu, T., Chen, Y. and Han, J. (2007). "Association mining in large databases: A re-examination of its measures." Springer. 621–628

Wu, T., Chen, Y. and Han, J. (2010). "Re-examination of interestingness measures in pattern mining: a unified framework." *Data Mining and Knowledge Discovery* **21**. 371–397

Zaki, M. J. (2000). "Sequence mining in categorical domains: incorporating constraints." ACM. 422–429

Zaki, M. J. (2001). "Spade: An efficient algorithm for mining frequent sequences." *Machine learning* **42**. 31–60