A Product Placement Framework to Improve Diversity in Retail Businesses

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science in Computer Science and Engineering By Research

by

Pooja Gaur
20173037
pooja.gaur@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500032, INDIA
April 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "A Product Placement Framework to Improve Diversity in Retail Businesses" by Pooja Gaur, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. P. Krishna Reddy

Dedicated to my family and friends

Acknowledgments

Firstly, I would like to express my gratitude towards my advisor Prof. P. Krishna Reddy. He introduced me to the research discipline and patiently guided me through the ups and downs of the journey. I also express my gratitude towards Prof. M. Kumara Swamy from CMR engineering and Prof. Anirban Mondal from Ashoka University who have put the effort in providing me continuous guidance and shaping up my research ideas.

I would like to thank my parents for supporting me in the direction of education. I would like to thank my elder brother and sister for motivating me to pursue research and supporting me in my graduate education.

I thank my friend Yogesh Sharma for supporting me by being an emotional anchor in multiple moments. I want to thank Srinivas sir from DSAC lab for guiding me on how to write the thesis. I would like to thank all my friends for their help and support throughout the time I spent in college.

At last, I would like to thank the almighty for the blessings and strength and countless people who have been important in my life.

Abstract

Retail stores have become a part of our daily lives and play a vital economic role in society. It has been reported that shelf-space allocation decisions in a retail store significantly impact the retailer's revenue. So, developing approaches for efficient product placement in available shelf space in retail stores is one of the key research issues. Several research efforts have been made to propose approaches for improving product placement in retail stores based on the knowledge of customer purchase history. Traditionally, several dynamic programming model based approaches were proposed to solve this problem. More recently, data mining approaches, like frequent pattern mining and utility pattern mining, have been employed to improve product placement based on the patterns extracted from customer purchase transactions.

In this thesis, we address the issue of product placement in a retail store by considering the diversity of products. Providing diverse options to customers can be useful in increasing the long term sustainability of retail stores. Other recommendation systems have also used increasing diversity to improve user interest. In the context of retail stores, a store that can cater to the needs of a diverse customer base (by providing diverse recommendations) has a higher chance to stay relevant for customers in the long term, thus, facilitating long term sustainability of retail businesses. We have proposed an approach for facilitating the placement of items in a diversified manner in a given retail store based on the concept hierarchy that exists among the items without compromising the revenue.

In the proposed approach, a concept hierarchy based approach is employed to determine the diversity value of the itemset quantitatively. The diversity of the given itemset captures the extent of its items belonging to multiple categories. By combining the notion of utility and diversity, we propose a framework to compute $diverse\ net\ revenue$ of the itemset. Given a set of transactions, price values of items, and concept hierarchy, we propose a methodology to build the Concept Hierarchy Utility Itemset Index (CHUI), which contains potential itemsets with high revenue and diversity. Next, we propose the approach to perform product placement based on the knowledge of itemsets in the CHUI. We conduct experiments on a real dataset, namely, Instacart retail dataset (containing 49,688 items and 1,31,208 transactions), to demonstrate the proposed approach's overall effectiveness.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Retail is the sale of goods and services from individuals or businesses to end users. Retailers are part of an integrated system called a supply chain. Retail stores, small or large, can be viewed as retailers who buy the products from manufacturers in bulk at a lower price and then sell the products to end customers at some higher value. Without retail stores, it would be very difficult for end users to have access to manufactured products. In this way, the retail stores act as the interface for end users to get access to the manufactured products. With higher and more diverse consumer demands, the rise of larger and larger retail stores has been observed.

In large retail stores, tens of thousands of items (the term, product and item is used interchangeably) need to be placed in the typically massive number of slots (of the shelves) of a given store. This poses several research challenges like scalable supply chain management [8], inventory management [64], stock-out management [64] and shelf space management [63, 64]. The presented work focuses on the issue of shelf space management. Existing research in [63, 64] indicates that product placement on retail shelf space affects consumer purchase patterns. Thus, efficient placement of items on the shelf space in retail stores can improve sales of the items as well as the revenue generated from sales. In this regard, several data mining approaches, like frequent pattern mining and utility mining, have been employed to get the products for placement from purchase history to improve the retail store's revenue.

In the existing utility mining based product placement approaches, efforts have been made to improve the retailer's revenue. In this thesis, we address the issue of product placement in a retail store by considering the diversity of products.

The remaining part of this chapter is organized as follows. In the next section, we present the background and overview of existing solutions. After explaining the research gap, we present an overview of the proposed approach. Finally, we list the contributions and provide an outline of the thesis organization.

Figure 1.1 Inside Supermarket

1.1 Background

Larger retail stores are able to cater to the diverse needs of any consumer and can act as a single point to get all retail needs met. This is the reason why over the past several decades, we have seen retail stores getting larger and larger. There are several popular medium-to-mega-sized retail stores that have relatively huge retail floor space. Some of such medium-sized retail stores are Walmart and supercenters. On the other hand, an example of mega stores includes Macy's Department Store at Herald Square (New York City, US) and Dubai Mall (Dubai). It is common for such mega-sized retail stores to have more than a million square feet of retail floor space.

Generally, medium-to-mega-sized retail stores offer a large variety of offerings to their customers based on product, brand, size, or price. Such large stores have numerous aisles, such that each aisle encompasses a wide range of products accommodated in the shelves' slots. A customer can find several varieties of chips, a multitude of dairy products, ready to eat food, clothes, appliances, to groceries, all within a single retail store. For example, Figure 1.1 shows a glimpse of what a customer sees inside a supermarket. Typically, a customer goes through various aisles while picking different items from shelves to get their desired products. In Figure 1.1, there are two aisles, and each aisle contains multiple shelves (a shelf is a horizontal row containing items). Out of these shelves, some shelves align with the user's eyes, some are above them, and some are below them. The shelves with higher visibility to the user (that align with their eye level) are called premium shelves. Each shelf is a horizontal arrangement

of the products. A slot refers to the space occupied by the product. So, products are arranged in slots of shelves or aisles. Large retail stores typically have a huge number of such aisles. This arrangement poses a complex yet enticing opportunity for retailers to organize the product in such a way as to improve customers' buying experience while aiming for high retail revenue (For example, increasing the reach of popularly purchased products, or having a high margin).

1.2 Overview of existing solutions

To deal with the problems that arise from the scale of retail stores, several research challenges have been posed. In this regard, existing works have focused on scalable supply chain management [8], inventory management [64], stock-out management [64], identification of frequent and/or high-utility itemsets [16] and strategic placement of items (products) as well as itemsets (set of products) for improving the revenue of retail stores [19].

In the area of product assortment and shelf space allocation in retail stores, several approaches using dynamic programming have been proposed in [28, 71]. Such methods depend on many parameters and are inadequate to deal with large scale data. The application of data mining approaches for deciding product placement presents a more efficient way. Frequent patterns are set of items that are frequently purchased together. Frequent pattern mining [4, 31, 55] is a pattern mining approach that analyzes customer purchase transactions to extract frequent patterns.

Frequent pattern mining is a commonly used technique in data mining, but it does not consider the price of items. This is an important factor, as the prices of items can greatly impact the revenue generated from a transaction. For example, while the pattern fish, caviar may be purchased less frequently than fish, chips, it could still result in higher revenue for a retail store due to the high price of caviar. To address this limitation, utility mining has been proposed as an alternative to frequent pattern mining, as it takes both frequency and price into account. In order to improve revenue for retailers, several itemset placement approaches have been proposed in the literature [16, 17, 18, 19, 53], which utilize specialized index structures (populated with customer purchase transaction data) to extract knowledge of high-revenue itemsets through utility mining.

Instead of solely focusing on placing high revenue itemsets, it can be useful to propose a variety of items to attract a diverse set of customers. Lately, many recommendation approaches have been proposed that do not solely focus on accuracy but also include new evaluation metrics such as diversity, novelty, and unexpectedness of the recommended choices to increase user satisfaction. In literature, efforts like [1, 39, 52] have been made to explore alternate recommendation approaches. The issue of improving the diversity in item placement for retail stores has been investigated in [25, 39]. In work [39] the author aimed at increasing diversity in retail items by mixing food items and other retail items. The work in [25] differentiated between two types of retailers, namely (a) limited-diversity retailers specializing in either food or non-food product categories and (b) extended-diversity retailers offering both food and non-food product categories. Another way to find diverse itemsets is to use the concept

hierarchy based diversity as presented in the work in [46]. Concept hierarchy maintains relationships between items. If we consider items as the leaf nodes and categories as inner nodes, the itemset which has items belonging to diverse categories is more diverse compared to itemset which has items belonging to similar categories.

1.3 Research gap

In the existing literature, data mining based techniques have been used multiple times to propose approaches to improve retail revenue based on customer purchase transaction history. Attracting a wider audience is another method of getting high revenue and maintaining long term sustainability. Stores can provide a wider diverse variety of items so that customers have more choice in terms of the items available to them for purchase. For example, a customer, who wishes to buy soft drinks, would generally prefer a wider and more diverse range of soft drinks (e.g., Coke, Coke-Light, Pepsi, Sprite, Limca, etc.) to choose from. As another example, a retail store could have different brands and varieties of soaps, shampoos, and tomato sauce. It is a well-known fact in the retail industry that customers belong to different market segments, e.g., Alice prefers a specific brand of tomato sauce or a specific brand of soft drink. In contrast, Bob may prefer another brand of tomato sauce and another brand of soft drink.

Given that each market segment is relatively large in practice, increasing the diversity in a retail store would generally attract customers from multiple market segments, thereby increasing the retail store's overall foot traffic. For example, if a retail store only stock Coke on its shelves, it would likely miss out on the customers whose preference could be Coke-Light or Pepsi. In fact, research efforts are being made to improve diversity for real-world retail companies by collecting data about sales, customer opinions, and views of senior managers as investigated in [25, 37, 61]. In essence, there is an opportunity to improve the product placement by incorporating the notion of diversity (of items) during the placement of itemsets. However, the idea of diversity in the context of retail stores has not been explored with existing approaches.

1.4 Overview of proposed approach

The notion of diverse frequent patterns has been proposed in [43, 44] to extract the knowledge of diverse items based on the customer purchase transactional dataset and a given concept hierarchy of the items. Furthermore, it has also been demonstrated that the knowledge of diverse frequent patterns can potentially be used to improve the performance of recommender systems. Observe that, while the works in [43, 44] address the diversity issue, they do not consider the context and issues associated with itemset placement in retail stores.

In this thesis, we have made an effort to improve the diversity of itemset placement in retail stores by proposing the notion of concept hierarchy based diverse revenue itemsets. In particular, based on

the notion of diverse revenue patterns, we have proposed a framework, which improves the diversity of itemset placement, while not compromising the revenue.

We name the proposed approach Diverse Revenue Itemset Placement ($DRIP$). The approach takes a transactional database, a concept hierarchy for the products, and the number of slots required to fill as inputs and outputs for the product placement. The $DRIP$ approach favors transactions with high revenue and high diversity. To this, we propose a new scoring metric named $diverse\_net\_revenue$ which combines diversity and revenue in a single metric. First, we create an index structure to store the top itemsets after parsing consumer purchase history. The index is a multi-level index with each level storing itemsets corresponding to a fixed size equivalent to level number, i.e., level 2 will contain all items with size 2 and so on. Within a level, the ordering is done based on $DRank$ and $revenue$. This helps in ordering itemsets from high diversity, high revenue to low diversity, low revenue. Once this index is constructed, the proposed scheme can be used to place itemsets in the available slots on the shelves to get the final placement.

To contrast this with other ordering measures, we have described Revenue Itemset Placement (RIP), which favors transaction ordering with high revenue and does not care about diversity. This scheme is based on the existing method and is treated as a baseline approach. We have also considered variations of the proposed approach (like ordering only by diversity) to observe the results of combining revenue and diversity in different ways. For performance evaluation, we have compared the revenue and diversity performance for the various approaches proposed on a real world dataset.

1.5 Contributions

The key contributions of the thesis are as follows.

1. We introduce the problem of concept hierarchy based diverse itemset placement in retail stores.
2. We present a framework and schemes for facilitating efficient retrieval of the diverse top-revenue itemsets based on a concept hierarchy.
3. We conduct a performance evaluation with a real dataset to demonstrate the overall effectiveness of our proposed schemes.

1.6 Thesis organization

We have organized the thesis in the following manner -

1. In Chapter 2, we discuss the related work about pattern mining in various domains. Next, we present the related work on concept hierarchy driven diverse pattern mining and how it has been used in various domains.

2. In Chapter 3, we discuss the background of our work in detail. Here, we explain the approach to building an itemset index and using concept hierarchy to get the diversity of an itemset.

3. In Chapter 4, we discuss the proposed revenue-based product placement framework to improve diversity in the retail business. We also present an illustrated example of the approach in action for a sample set of transactions.

4. In Chapter 5, we summarize the work presented in the thesis and provide a conclusion for the work.

Chapter 2

Related Work

In this chapter, we present the related work. First, we discuss the related work on shelf space management in retail stores. Next, we present the important pattern mining and utility mining approaches. After that, we discuss the pattern mining based product placement approaches in retail stores. Finally, we present the related work for diversity in pattern mining and concept hierarchy.

2.1 Shelf space management

Shelf space management is a method that deals with the allocation and organization of products on retail store shelves. Shelf space management can help increase sales and profit margins of inventory investment and enhance customer satisfaction by reducing out-of-stock occurrences [63]. Shelf space management consists of two important tasks (i) product assortment and (ii) shelf space allocation. Product assortment refers to the range of products that are offered by a retailer. The goal of product assortment is to provide customers with a diverse range of products to choose from and to ensure that the retailer is able to meet the needs and preferences of its target market. Shelf space allocation refers to the process of determining where and how products should be arranged on retail store shelves. It deals with the decision of allocating the available shelf space among the selected items that are included in product assortment [35]. This is an important aspect of retail operations, as the arrangement of products on shelves can have a significant impact on customer behavior and store profitability. In literature efforts like [6, 9, 10, 14, 15, 22, 23] have been made to improve the shelf space management.

In the early work [6], the author studied the relationship between product shelf space share and market share. They proposed a theoretical relationship between brand market share and the share of product display space, based on a profile of customer brand preferences. Further, they model stockout costs as a function of item demand. However, the lost sales due to stockouts were not incorporated into the demand or cost functions. Later, in the work [22], the author proposed a geometric programming model to solve the category space allocation problem with the goal of profit maximization. Their model incorporated direct and cross-space elasticity using a multiplicative model and modeled costs as a function of inventory investment and handling charges. Store size and size for each product category were included

as constraints to ensure that the number of products remained less than the available shelf space. The work in [23] extended the shelf space allocation model to include the effects of widely varying product growth potentials.

Further, in the work [14], the author focused on optimizing space allocation within a product category, using a model that incorporates both direct and cross-space effects and models cost as a function of sales per unit space. This work was coined as SHARP: shelf allocation for retailer's profit. In a later study, the work in [15] (SHARP II) extended the model by performing marginal analysis on a general theoretical model and proposed a search heuristic based on the convergence of the SHARP rule for each category. However, the use of marginal analysis becomes impractical with non-linear models, which limits the number of variables and functional forms that can be used.

To address the limitations of non-linear models, the work in [10] extended the SHARP II model by allowing simultaneous decisions on assortment selection and shelf space allocation. This work framed the placement problem as a constrained optimization problem, considering shelf space as a fixed resource and aiming to find the optimal allocation of space and assortment of stock keeping units to maximize the return on inventory. The model used simulated annealing to maximize the return on inventory, and included parameters such as shelf elasticities, search loyalty, and customer preference. However, the estimation of these parameters introduces uncertainty in the optimality of the results. However, in the work [9], the author demonstrated that even with variations in the judgemental estimates of these parameters by as much as 50 percent, the model outperformed traditional merchandising rules for product placement.

The use of space elasticity solutions for shelf space management often requires the estimation of a large number of parameters, which can be challenging for retailers who have access to vast amounts of data. Traditional methods may not be well suited to dealing with such large volumes of data, and can result in high costs and errors in the final model. In contrast, data mining techniques are well suited to extracting useful information from large datasets, and have been applied to the problem of determining product assortment and placement in retail stores. Hence, data mining approaches have become increasingly popular for this purpose.

2.2 Pattern mining based approaches for retail

In this section, we first provide an overview of pattern mining approaches. Then, we describe pattern mining-based product placement approaches for retail stores.

2.2.1 Pattern mining approaches

In recent years, retail stores have generated large amounts of customer purchase transaction data. By leveraging this data, there is a large potential to improve sales of retail stores by predicting customer choices and preferences. However, extracting useful information from such large datasets can be a

challenging task. In this section, we review the related work on pattern mining methods for extracting interesting patterns from large transactional databases. Specifically, we focus on two popular pattern mining techniques, namely *frequent pattern mining* and *utility mining*.

Frequent pattern mining is a well-known problem in the field of data mining, which involves extracting interesting and useful knowledge patterns from large transactional databases. In literature, frequent pattern mining is extensively applied to a variety of domains, including web data analysis [48], vehicle and communication data [21], biodata [29], and hardware monitoring in computer systems [70].

A pattern is a set of items belongs to a transactional database. A pattern is associated with two measures namely, *support* and *confidence*. The support of a pattern is defined as the number of transactions in the database that contain the pattern. A pattern is considered as a frequent pattern if its support is not less than a user-specified minimum support threshold. The confidence of a pattern $\{X,Y\}$ is a measure of how likely Y is purchased when X is purchased. The measure of support and confidence helps in the extraction of interesting association rules among the set of items purchased in retail stores. Among the many, the notion of frequency is well studied and most commonly used to extract interesting patterns from transactional databases [4, 5, 30, 32, 33].

In the work [4], the author proposed a level-wise *Apriori* algorithm for extracting association rules among the set of items. The *Apriori* is the first algorithm among pattern mining approaches. It generates candidate patterns by first identifying the frequent individual items in the dataset, and then extending those items to form larger and larger frequent patterns. The algorithm uses a bottom-up approach, starting with individual items and then combining them to form larger patterns. Once candidate items are generated, it has to scan through the database and uses a support metric to identify which patterns are frequent. Because of this scanning process, the algorithm has no choice but to scan the database as much as the maximum length among frequent patterns. This work employs downward closure property for efficient pruning of search space by avoiding the need to consider any superset of an infrequent subset of frequent itemsets. However, this approach requires multiple database scans for extracting frequent patterns. The UT-Miner [67] is a variant of *Apriori*, which is better suited to mining frequent patterns in sparse data, but it still suffers in performance due to its reliance on the *Apriori* method.

A more efficient *FP-growth* [33] algorithm was proposed to avoid multiple scans of the database for the generation of candidate patterns. The *FP-growth* addresses this issue by scanning the database only once and builds a hierarchical structure of the database, which is called *fp-tree*. The *fp-tree* structure helps to eliminate the candidate generation step. From the database, an *fp-tree* is first constructed, and then a conditional *fp-tree* is constructed based on the itemset counts in the path. The conditional *fp-tree* is mined to get the itemsets with a frequency greater than a given threshold value. The *fp-tree* avoids the need for candidate pattern generation, which allows it to achieve better performance than *Apriori* and UT-Miner in many cases.

Later, two important variants of the *FP-growth* algorithm have been proposed, namely, FP-growth-tiny [54] and IFP-growth [49]. FP-growth-tiny generates conditional fp-trees using conditional patterns without creating the conditional database. This allows it to quickly find frequent patterns in the data,

Table 2.1 Various methods for each category of UPM

UPM Category	**UPM Approaches**
Apriori-based	OOApriori, MEU, Two-Phase, FUM, DCG+
Tree-based	HYP tree, CTU-Mine, UP-Growth, CHUI-Mine
Projection-based	CTU-Pro, GPA, PB, PTA
New data format	HUI-Miner, D2HUP, FHM, EFIP, HUP-Miner

and making it more efficient than the original FP-growth algorithm. This approach improves the search space efficiency of FP-growth. On the other hand, IFP-growth, proposes an enhanced pruning method with a new tree structure called fp-tree+. This incorporates an address table to decrease the number of conditional fp-trees and improve mining speed. Overall, both FP-growth-tiny and IFP-growth are improvements on the original FP-growth algorithm, and give better performance and efficiency.

Another notion for pattern mining is *utility mining*. Utility mining deals with the concept of *utility* and *interestingness*. The utility is a measure of preference over a set of goods, and the *interestingness* measure is closely tied to a business value such as revenue. This addition of business value means that the downward property may not always hold, as the revenue of a superset may be larger or smaller than that of its subset, depending on the cost of the new items. This makes utility mining a more complex problem.

The existing approaches towards high-utility itemset mining can be broadly classified into four categories: *Apriori-based* approaches, *Tree-based* approaches, *Projection-based* approaches, *Data format based* approaches. Table 2.1 provides a summary of the various approaches within each category.

Several research efforts have been made in the area of utility mining [50, 59, 60]. These efforts aim to identify high-utility itemsets from transactional databases of customer transactions. In work [60], the author proposed the Utility Pattern Growth (UP-Growth) algorithm, which uses a data structure called the Utility Pattern Tree (UP-Tree) to track information about high-utility itemsets and employs pruning strategies for candidate itemset generation. Moreover, in work [50], the author proposed the HUI-Miner algorithm, which uses a data structure called the utility list to store utility and other heuristic information about itemsets. This enables it to avoid expensive candidate itemset generation and utility computations for many candidate itemsets. In the work[59], the author proposes an algorithm for mining closed high-utility itemsets, called CHUI-Miner, which can compute the utility of itemsets without generating candidates.

2.2.2 Pattern mining based product placement approaches for retail stores

The placement of items on retail store shelves has been shown to impact sales revenue. For example, in the work [13], the author demonstrates that retail store profits can be improved by efficiently using shelf space. Additionally, in the work [34], the author proposes a model and an algorithm for the

allocation of shelf space by considering the available space and the cost of the product. This method maximizes the total profit of a given retail store.

In retail stores, a variety of products compete for fixed shelve space. A variety of space elasticity models were proposed to solve this problem. In work [64], the author proposed a non-linear integer programming-based space allocation model to optimize space allocation in retail stores to generate revenue. However, in the work [63], the author proposed a method to solve the problem by relating it to the knapsack problem and allocating shelf and space according to the item profit. However, this method is non-linear in nature and uses a large number of variables, which reduces efficacy. Moreover, in work [12], the author utilized association rules to address the item placement problem in retail stores using frequent patterns for item selection and demonstrated their effectiveness. In another work [20], the author presented a data mining-based approach using multi-level association models to solve the problem. Furthermore, in work [26], the author investigated the use of mixed-integer programming for retail assortment planning and shelf space allocation to maximize retailer revenue.

A framework has been presented in [16, 19] to determine the top-revenue itemsets for filling a required number of retail-store slots, given that items can physically vary in terms of size. A specialized index structure, designated as the $STUI$ index, has been proposed in [16, 19] for facilitating quick retrieval of the top-revenue itemsets. Furthermore, an approach has been proposed in [17] for placing the itemsets in the premium slots of large retail stores to achieve diversification and revenue maximization. Notably, the notion of diversification in work [17] is based on reducing the placement of duplicate items. This facilitates improving long-term retail business sustainability by attempting to minimize the adverse impact of macro-environmental risk factors, which may dramatically reduce the sales of some of the retail store items. As such, the emphasis of the work in [17] is orthogonal to the focus of this work. Moreover, in work [17], the author proposed the kUI (k-Utility Itemset) index for quickly retrieving top-utility itemsets of different sizes. Additionally, in the work [18], the author addresses the problem of itemset placement by considering slots of varied premiumness.

2.3 Diversity in patterns and concept hierarchy

2.3.1 Related work on diversity in patterns

A pattern is a set of items. Consider two patterns bread, butter and beer, diaper, extracted from a supermarket's transactional database. Furthermore, consider that both patterns have equal support. A pattern is interesting if it can provide useful information or behavior about the data. In the example of bread, butter and beer, diaper, we would consider the latter to be more interesting because it includes items from different categories (drinks and baby items), while the former includes items that belong to the same category (food items). In other words, we say that the pattern beer, diaper is more diverse (has more diversity value) than bread, butter. This demonstrates the potential value of distinguishing between patterns with items belonging to many categories and patterns with items belonging to a few

categories. The existing pattern extraction approaches (frequent, sequential, periodic, etc.) fail to make such a distinction.

In the work [40], the author proposes using diverse association rules and diversity-based measures to evaluate the interestingness of data sets. To demonstrate their approach, the study compares two datasets from the UCI machine learning library (the adult [7] and breast cancer datasets [62]) and extracts the top 15 association rules from each. They use measures such as variance and Shannon entropy to compute the interestingness of these rules and find that data sets with more diverse rules are more interesting. This work highlights the potential value of incorporating diversity into the evaluation of data sets.

Moreover, in work [56], the author makes an effort to improve the performance of recommender systems by exploiting the notion of diversity in Case-Based Reasoning (CBR) approaches. In general, CBR systems solve new problems by using solutions to previously solved problems stored in a case-base. The effectiveness of a CBR system depends on its ability to select the right case for the target problem, which is often done using similarity-based measures. However, these measures may not always find new, relevant cases. The notion of diversity is exploited as an alternative to frequency measure. In the work [56], the author suggests diversity computation by altering the existing similarity measures. They defined diversity for a set of items as the average dissimilarity between all pairs of items. Improving the diversity characteristics of a fixed-size recommendation sacrifices similarity. A strategy was proposed in [11], that balances the trade-off between similarity and diversity in recommendation sets, allowing for diverse recommendations that are still similar to the target query.

In another work [69], the author proposes a method for improving user satisfaction in recommender systems through the use of diversity. Traditionally, the recommender system focused on optimizing accuracy using metrics like precision and recall. These metrics improve accuracy without considering the element of surprise in the recommendations. For this, in the word [69], the author proposed an approach focusing on the importance of diversity by trying to improve user satisfaction rather than accuracy. The approach returns a list of recommendations that better cater to a user's full range of interests through the selection of lists with low inter-list similarity. The experiments conducted in this work demonstrated that real users prefer more diversified results.

2.3.2 Related work on concept hierarchy

A concept hierarchy is a structure that organizes a set of concepts in a hierarchical manner or partial order manner. It is a way of representing and organizing knowledge patterns within a specific domain. In a concept hierarchy, concepts are arranged in a tree-like structure, with more general concepts at the top and more specific concepts at the bottom of the tree. The concept hierarchies express knowledge in concise, high-level terms and facilitate the mining of knowledge at multiple levels of abstraction. They are essential components for data warehousing, as they are used to form dimensions in multidimensional databases [36, 68].

In literature, concept hierarchies have been used in many ways, including taxonomy [24], ontology [47], and open-web directories [38]. Taxonomy [24] is the study of organizing and classifying different

types of organisms. This includes looking at their characteristics, such as how they look and behave, their genetics, and their biochemistry. So far, taxonomists have identified about 1.78 million species of plants, animals, and microorganisms, but it is believed that there may be up to 30 million species in total. We do not yet fully understand all of these species. Ontology [47] is the formal naming and definition of the types, properties, and interrelationships of entities that fundamentally exist within a particular domain of discourse. An open-web directory [38], (also known as a link directory) is a directory on the World Wide Web that categorizes links to other websites. It is used to organize the data collected from the World Wide Web into categories.

An approach for computing the diversity of a given pattern (itemset) using concept hierarchies is proposed in [46]. In this approach, items in a given domain are grouped into categories and the relationship between the items and their categories is represented in a tree-like structure. In this structure, items occupying the leaf nodes and the intermediate nodes represent the categories and the root node of the concept hierarchy is a virtual node. The merging behavior of the items in a pattern can be used to determine the pattern diversity. The patterns where items in the leaf nodes are merged at a higher depth of the tree (lower distance from the leaf) are less diverse compared to patterns where items in the leaf node are merged at lower depth of the tree (higher distance from the leaf).

2.4 How the proposed approach is different

From this literature study, we understood that frequent pattern mining and utility mining have been applied in self space management. In retail stores, it is important to consider the varied preferences of customers in order to enhance the satisfaction of the customer. Moreover, existing works utilized the concept hierarchies to increase diversity in recommendation systems [44, 45, 69]. Recall that data mining has lot of potential in the extraction of varied preferences like knowledge from market basket transactional data and can improve the revenue of the retail stores. In the literature, shelf space management has been well studied in a traditional manner and does not utilize data mining techniques. However, these approaches are not feasible when dealing with large amounts of customer purchase data. This gives us an opportunity to leverage the data mining techniques in shelf space management for extracting diverse frequent patterns from transactional data. Multiple methods have been proposed for item placement to improve revenue in retail stores. In this work, we explore the notion of concept hierarchy and extract diverse patterns from transactional data. We use this to propose an improved itemset placement scheme which helps in improving the itemset diversity while maintaining the revenue of retail stores.

To the best of our knowledge, there have been no prior approaches that use the notion of concept hierarchy to increase diversity in product placement in retail stores. In this thesis, we propose a method for extracting diverse patterns from large transactional databases and placing them in limited retail shelf slots. We compare our approach to various combinations and conduct experiments on real-world datasets to demonstrate that our approach optimizes retailer revenue while improving product diversity.

Chapter 3

Background

In this chapter, we explain the various concepts that have been used in the proposed approach. In section 3.1, we discuss the utility mining problem and the popular solutions used for it. In section 3.2, we dive into one top-k utility itemset mining approach and discuss how it works. In section 3.3, we discuss the existing placement approach in retail store which give weightage to the revenue of an itemset while placing it. We describe this approach and set it as a baseline used in comparisons during experiments. In section 3.4, we dive into a method to quantitatively describe the diversity of itemset using concept hierarchy.

3.1 Utility mining

Pattern mining is a field of data mining that involves extracting valuable information from large datasets. In recent decades, there have been numerous approaches for interesting pattern mining have been extensively studied. Some of the proposed approaches are frequent pattern mining (FPM) [3, 33], frequent episode mining (FEM) [1, 2], association rule mining (ARM) [4, 5], and sequential pattern mining (SPM) [27]. While these algorithms utilize various interestingness measures, such as frequency and co-occurrence, they often do not allow for the discovery of patterns that are oriented toward utility.

The utility is a measure of preferences over a set of goods. It represents the satisfaction that a consumer derives from those goods. As a subjective measure, a utility can be defined as the usefulness of an itemset, as described in [65, 66]. However, traditional approaches to pattern mining that rely on frequency or other metrics may not always identify patterns that are truly useful to the user. To address this limitation, utility-oriented pattern mining (UPM) has emerged as an important task in the field of data mining, focusing on identifying patterns that are truly useful to the user.

Definition 1 (High Utility Itemset, HUI [58, 65])**.** *The utility of an item i_j appearing in a transaction T_q is denoted as $u(i_j, T_q)$ and defined as $u(i_j, T_q) = q(i_j, T_q) \times pr(i_j)$. The utility of an itemset X in T_q is defined as $u(X, T_q) = \sum_{i_j \in X \wedge X \subseteq T_q} u(i_j, T_q)$. The total utility of X in a database D is $u(X) = \sum_{X \subseteq T_q \wedge T_q \in D} u(X, T_q)$. An itemset is said to be a High-Utility Itemset (HUI) if its total utility*

Table 3.1 Commonly used symbols and their expression

Symbol	Description
I	An unordered set of m distinct items $I = \{i_1, i_2, i_3\}$
k-itemset	An itemset with *k* number of items in itself
X	An itemset with *k* distinct items
$q(i_j, T_q)$	The purchase quantity of an item i_j in transaction T_q
$u(i_j, T_q)$	Utility of item i_j in transaction T_q
$u(X, T_q)$	The utility of an itemset X in transaction T_q
$minutil$	A predefined minimum high utility threshold
$TWU(X)$	The transaction-weighted utilization of an itemset
$HTWUI$	The high transaction-weighted utility itemsets
$TWDC$	The transaction-weighted downward closure property
HUI	A high-utility itemset

in a database is no less than the user-specified minimum utility threshold (such that $u(X) \geq minutil$); otherwise, it is called a low-utility itemset.

Definition 2 (Utility Oriented Pattern Mining, UPM)**.** *UPM is a new mining framework that utilizes the utility theory and various mining techniques (e.g., data structure, pruning strategy, upper bound) to discover interesting patterns (e.g., High-Utility Itemset Mining (HUI), High-Utility Association Rule Mining (HUAR), High-Utility Sequential Pattern Mining (HUSP), High-Utility Sequential Rule Mining (HUSR), High-Utility Episode Mining (HUE)), and these derived patterns can lead to utility maximization and high benefit in business or other tasks.*

In this section, we take a closer look at the well known *Two-Phase* [51] method which is an Apriori-based method to mine high-utility itemset.

The Two-Phase algorithm, proposed in [51], introduced the concept of Transaction-Weighted Downward Closure ($TWDC$) property to address the challenge of the downward closure property of support measure not holding for utility measures, which are neither monotone nor anti-monotone. The $TWDC$ property states that if an itemset X is not a High Transaction-Weighted Utility Itemset ($HTWUI$), then no superset of X can be a HUI. The algorithm discovers HUIs in two phases. In phase 1, it uses an TWU upper bound to prune the search space and finds all itemsets X with $TWU(X) \geq minutil$. It starts by scanning the database once to get all 1-itemset $HTWUI_1$, and then generates (k+1)-level candidate itemsets (of length k+1) from length-k candidates $HTWUI_k$ (where $k > 0$). It examines the TWU

Table 3.2 Utility of items

Item	A	B	C	D	E
Utility	10	5	6	9	3

Table 3.3 Transactional database

	A	**B**	**C**	**D**	**E**
T1	2	3	0	0	0
T2	0	4	3	9	0
T3	1	2	1	4	3
T4	6	4	2	1	2
T5	0	2	1	6	5
T6	4	3	0	0	1
T7	1	5	0	0	6
T8	3	0	3	3	2
T9	0	0	1	1	0
T10	0	2	0	2	3

values of candidates by scanning the database once for each iteration, and terminates when no more candidates can be generated. In phase 2, it scans the database again to calculate the exact utility of each candidate in the set $HTWUI_k$ and filters the high utility itemsets from the high transaction-weighted utilization itemsets found in Phase 1.

Definition 3 (Transaction Utility [51])**.** *The transaction utility of transaction T_q, denoted as $tu(T_1)$, is the sum of the utilities of all the items in T_q: $tu(T_q) = \sum_{i_p \in T_q} u(i_p, T_q)$*

Definition 4 (Transaction-Weighted Utilization [51])**.** *The transaction-weighted utilization of an itemset X, denoted as $twu(X)$, is the sum of the transaction utilities of all the transactions containing X :* $twu(AD) = tu(T_4) + tu(T_8) = 14 + 57 = 71$.

Definition 5 (High Transaction-Weighted Utilization Itemset [51])**.** *For a given itemset X, X is a high transaction-weighted utilization itemset it $twu(X) \geq \epsilon$, where ϵ is the user specified threshold.*

To provide further insight into the functioning of the Two-Phase algorithm, we will use an example. Suppose the items in our database are A, B, C, D, E. Table 3.2 gives the per-item utilities for these items. These utilities can be profit/cost etc and are assigned to the items by the user. Table 3.3 gives the

Table 3.4 Transaction utilities

TID	TU	TID	TU
T1	35	*T6*	58
T2	119	*T7*	53
T3	71	*T8*	81
T4	107	*T9*	15
T5	85	*T10*	37

Table 3.5 Transaction weighted utilization after phase 1

Itemset	TWU	Itemset	TWU
A	405	B	565
C	478	D	515
E	492	A,B	324
A,C	259	A,D	274
A,E	370	B,C	382
B,D	419	B,E	492
C,D	478	C,E	344
D,E	381	A,B,E	289
A,C,D	259	A,C,E	259
A,D,E	259	B,C,D	382
B,C,E	263	B,D,E	263
C,D,E	263	A,C,D,E	259
B,C,D,E	263	-	-

Table 3.6 Final high utility itemsets after phase 2

TID	TU	TID	TU
D	234	A,B	225
B,D	268	C,D	282
A,B,E	226	A,C,D	208
B,C,D	282	A,C,D,E	229

transactional database. Each row represents a transaction T_i, where each column gives the number of times that item is present in the transaction.

Given Table 3.2 and Table 3.3, we can calculate the transaction utility (TU) for each transaction in the database. TU is calculated as per the definition 3. Transaction utilities are presented in Table 3.4.

To perform pattern mining, we need a user specified threshold called $minutil$. Itemsets with utility greater than or equal to $minutil$ are high-utility itemsets and the itemsets with utility less than $minutil$ are low-utility itemsets. Consider the $minutil$ for our example as 200.

To start the phase 1, we calculate the transaction weighted utilization (TWU) for one-itemsets. We calculate the TWU for items based on definition 4. The TWUs for one itemsets are $\{A : 405|6, B : 565|8, C : 478|6, D : 515|7, E : 492|7\}$ where each entry is $item : twu|frequency$. Now, for each iteration, we see if the TWU of the itemset from the list is greater than $minutil$ and extend them by size 1 to create a list of candidates. We continue this process till we get no candidates. Finally, we have a list of itemsets with TWU greater than $minutil$. In phase 2, the algorithm does a scan of the candidate list to get the high utility itemsets. Table 3.5 shows the itemsets generated by phase 1. Table 3.6 shows the itemsets generated by phase 2.

3.2 k-Utility Itemset index

Identifying High Utility Itemsets (HUIs) involves setting a user-defined threshold value for $minutil$. However, in practical situations, it can be challenging for users to determine the appropriate $minutil$ value that will yield the desired number of patterns. Additionally, some $HUIM$s may not perform well in cases where there are negative utilities, and extracting itemsets from a large dataset can be time-consuming. To address these issues, various algorithms have been proposed for finding the top-k high utility itemsets from a given transactional database. One such approach is to use a data structure called a (k-$utilityitems$) to efficiently retrieve the top utility itemsets.

The k-Utility Items (kUI) index [18] is a multi-level index that organizes a large number of itemsets by size. Each level of the index corresponds to a specific size of an itemset. The kUI index is divided into N levels, with each level containing a fixed number of itemsets, denoted by λ. At the k^{th} level of the index, the kUI stores the $top - \lambda$ ordered itemsets of size k. The kUI index is designed to optimize the process of finding potential itemsets of a given size from a large dataset.

3.2.1 Description of kUI index

Consider a set T representing a transactional database wherein transactions are made from a finite set of m items, Λ. Each item $i \in \Lambda$ is associated with a price ρ_i and a frequency of sales σ_i. Consider the $revenue$ as a utility measure for this case. We formally define the notion of $net - revenue$ as NR.

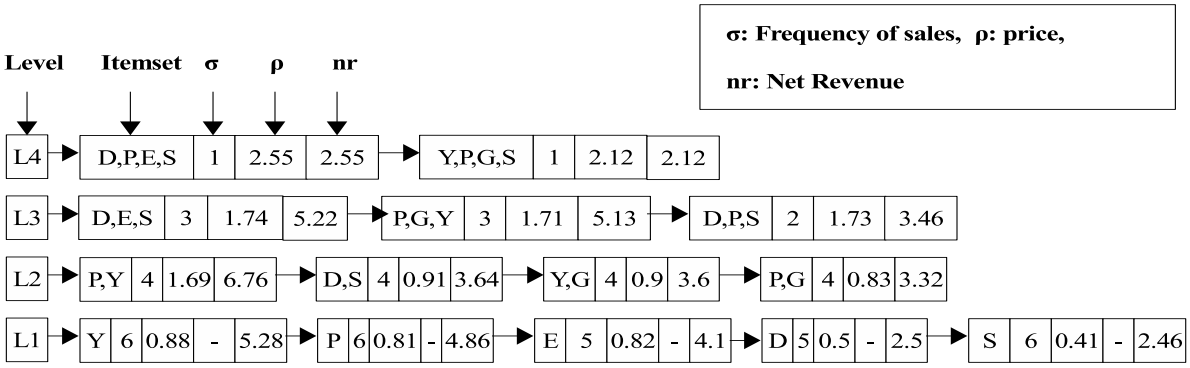

Figure 3.1 Example of the kUI index

Definition 6 (Net Revenue)**.** *Net Revenue of a specific item k is the product of price of the item ρ_k and its frequency of sales σ_k i.e.,*

$$NR_k = (\rho_k * \sigma_k)$$

Level n of the kUI indicates that the itemsets of size n are present at this level. This helps in the fast retrieval of $top-\lambda$ itemsets of any size. Each level is a linked list wherein each element has the structure of $\{itemset, \sigma, \rho, NR\}$, where $itemset$ is a set of items, σ refers to the frequency of item sales, ρ refers to the item prices and NR refers to the net revenue of the itemset and is computed as $\sigma \times \rho$.

Figure 3.1 shows an illustrative example of the kUI index. Each node in the linked list at the k^{th} level contains a tuple of $\langle itemset, \sigma, \rho, nr \rangle$, where ρ is the price of the given itemset $itemset$, σ is the frequency of sales of the itemset and nr is the net revenue, which is computed as $(\rho \times \sigma)$. The itemsets at each level of the kUI index are sorted in descending order based on net revenue to facilitate quick retrieval of the top-revenue itemsets.

3.3 Itemset placement in retail store

In this section, we will examine an existing method for using the kUI index to place itemsets based on revenue, which we will refer to as the Revenue-based Itemset Placement approach (RIP). This approach will serve as a baseline for our proposed method.

In the work [18], the author proposed an approach for itemset placement in retail stores. This approach considers that along with having shelves, the retail stores have a concept of *premiumness*, and given a choice to place a high revenue itemset on a shelf, it is beneficial to place the itemset in a high premium slot.

In our case, we will simplify this approach by assuming that the retail store has multiple shelves, but the premiumness of each shelf is not considered. We will refer to this simplified baseline approach as the Revenue-based Itemset Placement (RIP) method.

The Revenue-based Itemset Placement (RIP) approach consists of two main components. The first part involves parsing a transactional dataset and storing the top utility itemsets in an index for efficient retrieval. The second part involves systematically placing these itemsets in the available slots.

For the first component of the RIP approach, we use a kUI index as the data structure to store the itemsets of different sizes, ordered by their net revenue. For the second component, we follow the following procedure: while there are empty slots available, select an itemset from the top of the different levels of the kUI index that maximizes the revenue gained by the retailer per slot. This process is repeated until all itemsets have been placed.

3.4 Concept hierarchy derived diversity

A Concept Hierarchy (CH) is a structure that organizes a set of concepts in a hierarchical manner. It is a way of representing and organizing knowledge within a specific domain. In a concept hierarchy, concepts are arranged in a tree-like structure, with more general concepts at the top and more specific concepts at the bottom. There are some other names for concept hierarchy in the literature, such as taxonomy, is-a hierarchy [57], and structure attribute [42].

Concept hierarchies organize data or concepts in hierarchical or partial order forms, allowing for the expression of knowledge in concise, high-level terms and facilitating the mining of knowledge at multiple levels of abstraction. Concept hierarchies can be useful in a variety of contexts, including information retrieval, natural language processing, and education. They can also be used to help build and organize thesauri, taxonomies, and other types of classification systems. These hierarchies are also used to form dimensions in multidimensional databases, making them important components for data warehousing [36, 68]. For example, in a concept hierarchy, ”tea” and ”coffee” may be child nodes with ”beverage” as their parent, connected by the relation ”belongs to.”

The notion of diversity in a pattern refers to the extent to which the items in the pattern belong to multiple categories. This diversity can be measured by examining the merging behavior of the items in the pattern, or how they fit into a concept hierarchy. A pattern with low diversity would have items that are mapped to the same or few categories in the CH, while a pattern with high diversity would have items that are mapped to multiple categories and take longer to merge into higher-level categories as they are traversed from the root of the CH. By analyzing the merging behavior of the items in a pattern, we can determine its diversity value.

As an example, the concept hierarchy in Figure 3.2 is constructed using a sample of products from the Instacart Market Basket Analysis dataset [41]. In this hierarchy, the pattern Carrot, Lettuce has low diversity because the items in the pattern quickly merge into higher-level categories, while the pattern

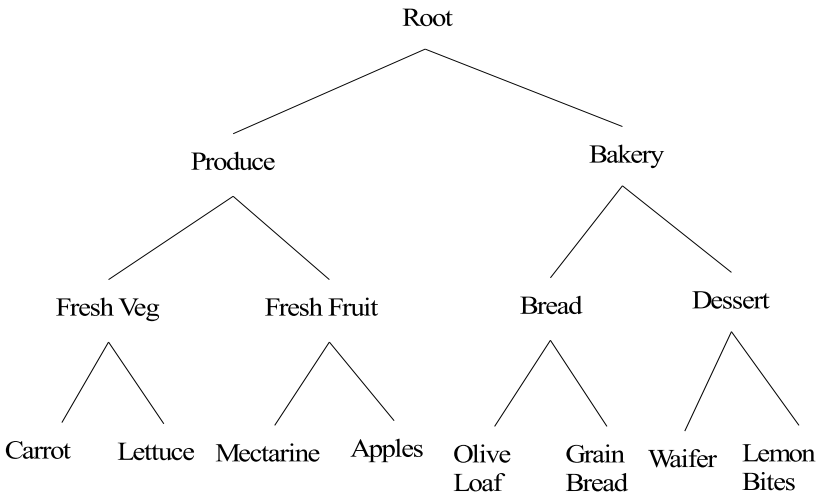

Figure 3.2 A sample concept hierarchy from instacart market basket analysis dataset

Carrot, Grain Bread has the highest diversity because the items take longer to merge and eventually all merge at the root of the hierarchy.

Given a pattern and a concept hierarchy [43, 44, 46], the diversity is computed using the Diversity Rank ($DRank$) metric. Given a set of items and the concept hierarchy C, the formula to compute the $DRank$ of a pattern Y is as follows

$$DRank(Y) = \frac{|\Pi(Y/C)| - (|Y| + h - 1)}{(h-1)(|Y| - 1)} \tag{3.1}$$

In the formula mentioned above, $|\Pi(Y/C)|$ represents the number of items in the projection of concept hierarchy C for pattern Y. The projection is a sub-tree of the concept hierarchy that includes only the paths from the root to the nodes of the items in Y. $|Y|$ is the number of items in pattern Y, and h is the height of the concept hierarchy, which is the maximum length of the root to leaf node path.

To understand how to measure diversity using concept hierarchies, consider the example in Figure 3.3, where the concept hierarchy is a tree and $P_1 = X, Y$ and $P_2 = Y, Z$ are the patterns for which we want to calculate the diversity rank ($DRank$). The projection of P_1 contains the elements $R, C1, X, Y$, and the projection of P_2 contains the elements $R, C1, Y, C2, Z$. The height of the concept hierarchy is 2. Using these values, we can calculate the $DRank$ for both P_1 and P_2 as follows (note that the root node (R) is a dummy node and is not included in the calculation):

$$DRank(P_1) = \frac{(3 - (2 + 2 - 1))}{(1 * (1))} = 0$$

$$DRank(P_2) = \frac{(4 - (2 + 2 - 1))}{(1 * (1))} = 1$$

Based on the calculations above, it can be seen that the diversity rank of P_2 is greater than the diversity rank of P_1. This can also be observed visually: the items in P_1 merge quickly into higher-level

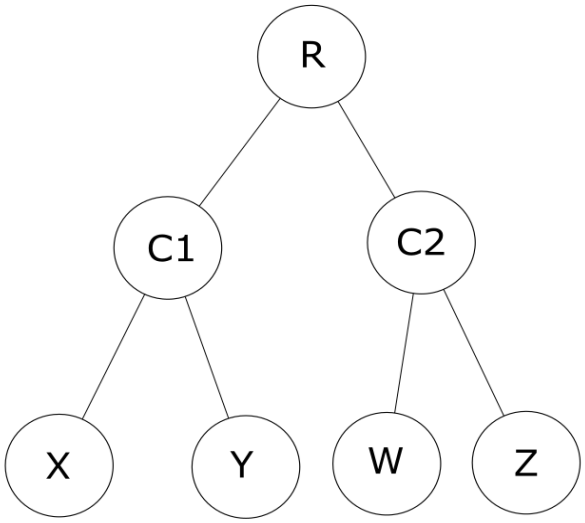

Figure 3.3 Example concept hierarchy

categories, resulting in low diversity, while the items in P_2 take longer to merge, eventually all merging at the root, resulting in higher diversity.

3.5 Summary

In this chapter, we introduced the concepts that are used in our proposed approach for diverse pattern mining in retail. We first took a look at the general idea of utility mining and how an apriori based method is used to perform utility pattern mining. Next, we discussed a variation of utility pattern mining and top-k utility pattern mining in particular. We discussed the need for top-k utility mining and how the kUI index works. We use these two concepts to implement a diversity and utility based, top-k utility mining method.

Next, we discussed a baseline approach for pattern mining and placement in a retail store. This approach is based on prioritizing the revenue of the itemsets to increase revenue for the retail stores. We use this approach to compare our proposed approach. Finally, we introduced the concept of concept hierarchy and how it can be used to measure the diversity of patterns using the diversity rank ($DRank$). Our proposed approach uses these concepts to prioritize patterns based on both diversity and utility for placement in retail stores.

Chapter 4

A Revenue-based Product Placement Framework to Improve Diversity in Retail Businesses

In this chapter, we present the proposed approach in detail and demonstrate the benefit of the proposed approach toward improving diversity and revenue in a retail business. The core approach delivered in our work is Diversity and Revenue Based Itemset Placement ($DRIP$). We first explain the components of this approach in depth, and then we present the other implemented approaches as variations of the primary method.

In section 4.1, we discuss the basic idea of the proposed approach. In section 4.2, we explain the proposed approach to improve diversity and revenue. In section 4.3, we provide the illustrative example to explain the proposed approach. In section 4.4, we explain the other implemented approaches for comparison. We provide the experimental results in section 4.5 and give a summary and conclusion to the chapter in the last section, 4.6.

4.1 Basic idea

As discussed in the introduction (chapter 1), the placement of products on the shelves of retail stores can considerably impact the revenue generated from sales. The transactional data of retail customers provide rich information about customer purchase patterns. The transactional data is a collection of patterns (or itemsets) where each transaction is the set of items bought by a customer during a single purchase. There has been significant interest in using this data to optimize product placement in order to increase revenue for large retail stores.

Given a transactional database, we presented an approach to identify high-revenue itemsets in section 3.3. We designate this approach as Revenue-based Itemset Placement (RIP) approach and use it as the baseline approach in the experiments. The RIP approach uses a specialized index to rank and place high-revenue itemsets on the shelves in order to maximize retailer revenue. The RIP approach is effective at identifying high-revenue itemsets, but it does not consider the diversity of items in the itemset placement process. To address this limitation, in section 3.4, we have discussed an approach to

compute the Diversity Rank ($DRank$) of an itemset which considers the relationships between different items in the retail store as represented by a concept hierarchy. By incorporating the $DRank$ of an itemset into the RIP approach, we aim to identify a diverse set of high-revenue itemsets for placement in retail stores. The goal of this approach is to improve the diversity of itemset placement in retail stores while maintaining overall revenue.

The proposed approach aims to develop a methodology that identifies potential high-revenue itemsets with relatively high $DRank$ values. The basic idea is to propose a new ranking mechanism to rank itemsets based on both revenue and diversity. Hence, given an itemset X, we propose a new measure called $diverse\_net\_revenue$ (dnr). Specifically, we define $dnr(X)$ as per definition 7 for an itemset X. The goal of this approach is to identify itemsets that are both diverse and high-revenue, which can be useful for optimizing placement in retail stores.

Definition 7 (Diverse Net Revenue)**.** *Diverse Net Revenue of an itemset X is the product of net revenue of the itemsets and its $drank$.*

$$dnr(X) = drank(X) \times nr(X)$$

Based on the notion of $diverse\_net\_revenue$, we propose an approach, which we shall henceforth refer to as the Diverse Revenue Itemset Placement ($DRIP$) approach. The $DRIP$ approach includes the following components:

1. a methodology to identify top itemsets with high dnr value and build an index called Concept Hierarchy based Diversity and Revenue Itemsets ($CDRI$) index.
2. a methodology to place the itemsets by exploiting the $CDRI$ index, given the number of slots as input.

Observe that the proposed $DRIP$ approach prioritizes the itemsets with high net revenue and high $DRank$ values. Consequently, it may miss some top-revenue itemsets with low $DRank$ values. As a result, even though $DRIP$ improves both revenue and diversity, it may not outperform the RIP approach on the net revenue aspect. However, it provides the extra benefit of giving users more choices, addressing a wider audience, and increasing long-term sustainability while maintaining high retail revenue.

We also propose another approach called Hybrid Itemset Placement (HIP) approach, which considers the top-revenue itemsets as in the RIP approach and top itemsets with high revenue and high $DRank$ values as in the $DRIP$ approach. In this implementation, we provide a configurable hyper-parameter called $RD\text{-}ratio$, which allows for customizing the amount of diversity we want to introduce in the placement. Another approach that we consider is the Diverse Itemsets Placement (DIP) approach, which focuses solely on maximizing diversity in the placement of itemsets. These approaches are used to compare the effects of different combinations of combining high revenue and diversity in retail placement.

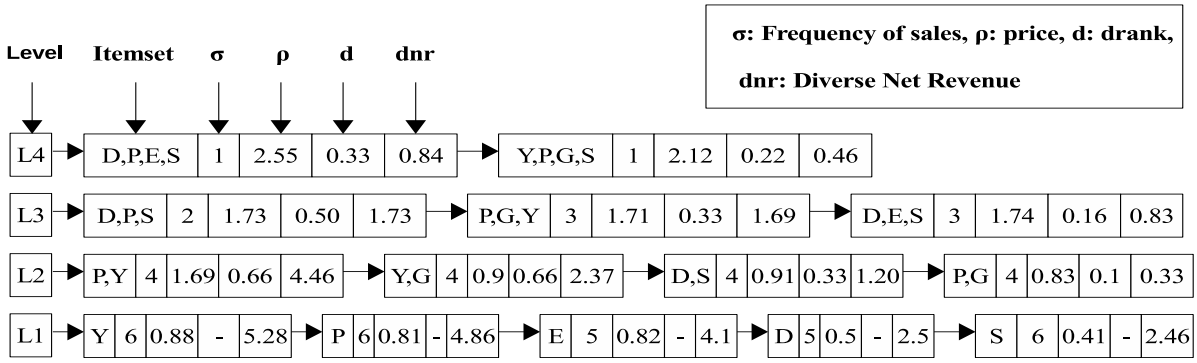

Figure 4.1 Illustrative example of the CDRI index

4.2 Proposed approach

This section explains the proposed Diverse Revenue Itemset Placement approach. The first subsection presents the novel approach of building an index that prioritizes itemsets based on both revenue and diversity. The second subsection explains the placement scheme that utilizes this index to populate the shelves in retail stores.

4.2.1 Building of the CDRI index

The Concept Hierarchy based Diversity and Revenue Itemsets index ($CDRI$), is a proposed index that ranks itemsets based on both diversity and revenue using a concept hierarchy. Concept hierarchy is a taxonomy of items and categories that can be used to define diversity for itemsets objectively.

$CDRI$ is a multi-level index containing N levels. At each level of the $CDRI$ index, λ itemsets are stored. The k^{th} level of the $CDRI$ index corresponds to itemsets of size k. This size-based arrangement optimizes the task of fetching diverse patterns of any desired size ordered by its $diverse\_net\_revenue$.

Each level of the $CDRI$ corresponds to a hash bucket. The data is stored as a linked list of nodes at each level. The node is a data structure containing the required information for an itemset. Each node in the linked list at the k^{th} level contains a record with the following fields: $< itemset, \sigma, \rho, DRank, dnr >$. Here, $itemset$ is the pattern of interest, ρ is the price of the given itemset, σ is the frequency of sales of the itemset, $DRank$ is the objective value indicating the diversity of itemset, and dnr is $diverse\_net\_revenue$ for the itemset.

Table 4.1 Utility of items

Item	P	Y	G	S	D	E	M	A
Utility (price)	10	5	6	9	3	1	1	1

Table 4.2 Transactional database

Transactions
P,Y,G
G,S,M,A
D,E,S,P
A,D,P,S,G
P,Y,S,G
M,A,D,E,S
D,E,G,Y
P,M,Y,G
P,A,E,Y
Y,S,D,E

The $CDRI$ index is built level by level, starting with level 1. Level 1 of the $CDRI$ index is formed by inserting a record for each item from the transactional dataset with $support$ greater than a user-defined threshold $support\_threshold$. Thus, it specifies the items that will be used for the construction of subsequent itemsets in other levels. For any level k, we follow the following process: (1) generate candidate itemsets by concatenating the itemsets of the $k-1$ level with the items of level 1, (2) remove duplicates and exclude itemsets with size different from k, (3) compute the support of each candidate itemset by scanning the transactional database, For each itemset X, the price is calculated. The price of the itemset is equal to the sum of the prices of the items in it, and $DRank(X)$ is computed using the formula in equation 3.1. (4) Next, we compute the $diverse\_net\_revenue$ (dnr) of each itemset using the formula in equation 7, (5) sort the itemsets according to their dnr, and (6) filter the top-λ itemsets at the k^{th} level of the $CDRI$ index.

Consider items and corresponding prices presented in Table 4.1. Suppose the concept hierarchy is present for those itemsets as shown in Figure 4.2. Table 4.2 shows the example of customer purchase history. Figure 4.1 depicts an example of the $CDRI$ index as used in the proposed approach. Each node in the linked list at k^{th} level contains a tuple of ($itemset$, σ, ρ, $drank$, dnr).

Algorithm 1 contains the pseudo-code to build the $CDRI$ index. The algorithm essentially has two parts, i.e., building level 1 of the index and building level 2 to max_level of the index. All items (i.e.,

Algorithm 1: Building of CDRI index

Input: T: Transactional database,
ρ_i: price for each item i,
C: Concept hierarchy,
max_level: Number of levels to create in $CDRI$,
λ: Max number of entries in each level of $CDRI$,
st: support threshold

Output: $CDRI$ /* CDRI index */

Initialize $CDRI$ with max_level empty lists;

/* Building level 1 of CDRI */
Scan T and compute support $s(X)$ for each item X in T and store it in array A;
Sort A based on the support ;
Remove all items from A which have the support less than the st ;
Select top-λ items from A and for each item X, insert $\langle X, s(X), \rho(X)\rangle$ in $CDRI[1][\lambda]$ in the descending order of the $support \times price$;

/* Building level 2..max_level of CDRI */
for $lev = 2$ *to* $lev = max\_level$ **do**
    Create a combination of itemsets in lev - 1 with the items in level 1 and generate all itemsets of length lev and store in A;
    Scan T and compute support $s(X)$ for each itemset X in A;
    Sort A based on the support;
    remove all items which have support less than the st;
    Compute price ρ_i for each itemset X in A by combining the price values of the items in X;
    Compute $DRank(X)$ for all X in A;
    Sort all itemsets in A in descending order based on $diverse\_net\_revenue$ (dnr) values of itemsets. i.e., $dnr(X) = DRank(X) \times Support(X) \times \rho(X)$;
    Select λ itemsets from A in the descending order of dnr value of the itemsets and for each itemset X in A, insert $\langle X, s(X), \rho(X), DRank(X), dnr(X)\rangle$ in $CDRI[lev][\lambda]$ in the descending order of dnr value of the itemsets in A;
end
return $CDRI$;

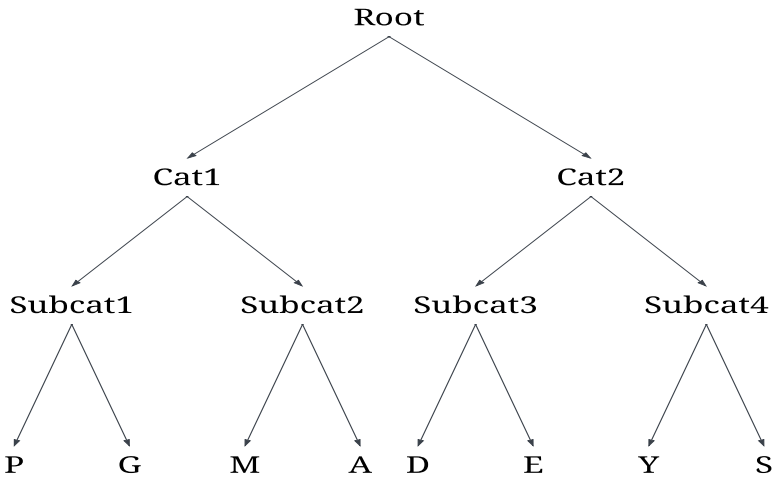

Figure 4.2 Sample concept hierarchy

one-itemsets) are picked from the transactional dataset T, placed in a temporary list A, and sorted based on $support$ (Line 1-4). Then, items in A with frequency less than $support\_threshold$ are removed (Line 5). The top-λ items are picked from A and placed in level 1 of the $CDRI$ index (Line 6). For each level, lev in 2 to max_level, all possible candidate itemsets of size lev are generated using itemsets placed in level 1 and level $lev - 1$ and placed in list A (Line 9). Items in A with a support less than $support\_threshold$ are removed (Line 10-12). Now, the price for each itemset in A is calculated as the sum of the prices of items it has (Line 13), and $DRank$ is calculated for each itemset in A using the concept hierarchy (Line 14) with the method discussed earlier. We have a $diverse\_net\_revenue$ now. Items in A are sorted based on decreasing order of $diverse\_net\_revenue$ (Line 15). Finally, top-λ itemsets are picked and placed to form one level of the $CDRI$ index (Line 16). This process is repeated for each level to form a complete $CDRI$ index (Line 18).

4.2.2 Itemset placement scheme in DRIP

After the completion of the former step, we have an index for efficiently retrieving itemsets of any given size in the order of decreasing dnr ($diverse\_net\_revenue$). We want to leverage this data structure to place itemsets on the shelves of retail stores.

In our proposed approach, we consider that we know the number of slots (in the shelves) to fill for a given retail store. Each slot represents the space occupied by one item, and we have limited the scope of the problem by assuming that each item occupies one slot of space, i.e., their size is equivalent. So, given the number of slots, the proposed placement scheme ($DRIP$) places itemsets.

It should be noted that the $CDRI$ index contains itemsets of different sizes. We only consider itemsets with a size greater than 1 for placement, as demonstrated in our experiments. Given itemsets of different sizes and net revenue values, our approach must decide which size of itemset to pick in order

to maximize the revenue contribution per slot. To this end, we define the concept of $revenue\_per\_slot$ in equation (8), which allows us to evaluate the performance of different size itemsets in terms of their revenue contribution per slot occupied.

Definition 8 (Revenue Per Slot)**.** *Consider an itemset X of size k which consumes k slots. $revenue\_per\_slot$ refers to the per slot revenue contribution of the itemset.*

$$net\ revenue\ per\ slot\ for\ X = \frac{price(X) \times support(X)}{size(X)}$$

In simple terms, the placement approach is to keep picking the itemsets with maximum $revenue\_per\_slot$ and placing it in the available slots till no slot remains unfilled.

Algorithm 2: DRIP placement algorithm

Input: $CDRI$ index,
Ts: Total number of slots

Output: $slots$ (Placement of itemsets in slots)

1 Initialise slots[Ts] to NULL ;
2 $CAS \leftarrow Ts$ /* CAS indicates the number of currently available slots */;
3 **while** $CAS > 0$ **do**
4 Starting from level=2, select one itemset with top dnr from each level of $CDRI$ index and place it in A;
5 Choose the itemset X from A which has the highest $Revenue\_per\_slot$;
6 Remove X from $CDRI$ index;
7 If $(CAS - |X|) < 0$ then break;
8 Place X in slots[Ts];
9 CAS = CAS - $|X|$;
10 **end**
11 **return** $slots$;

Algorithm 2 depicts the steps for the proposed $DRIP$ scheme. The algorithm stops when CAS (currently available slots) becomes 0. The CAS is initialized as the number of slots to fill (Line 2). While there are slots left to fill, get the top node of each level of the $CDRI$ index and place it in a temporary list A (Line 4). Now, the best node (node with highest $revenue\_per\_slot$) is picked from A and also removed from $CDRI$ as it is chosen to be placed (Line 5-6). Then, append the itemset of this node in the slots and reduce the number of slots currently available as we have placed an itemset (Line 7-9). When the loop ends, the itemsets have been placed in the $slots$ and the algorithm returns $slots$ (Line 11).

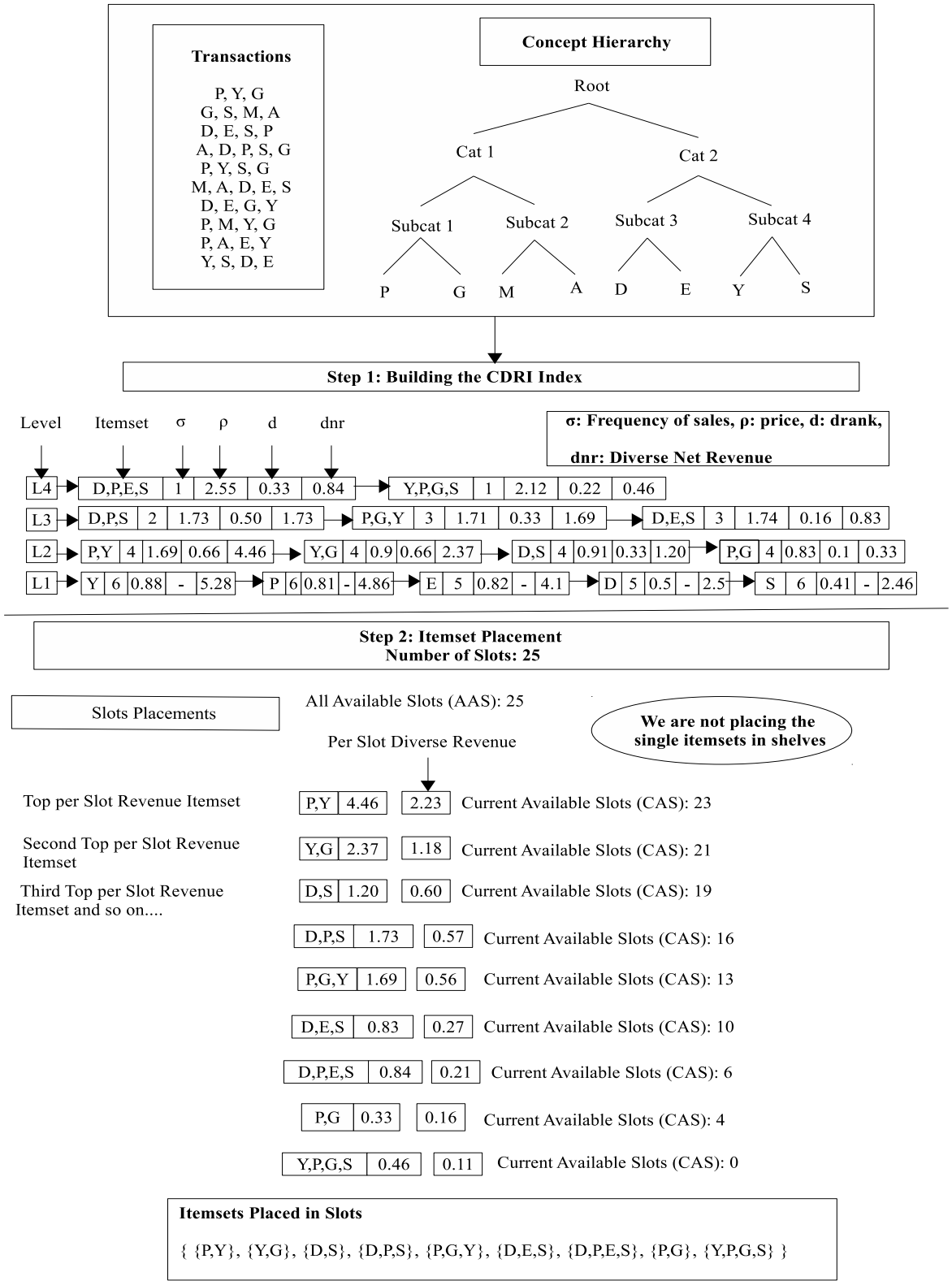

Figure 4.3 Illustrative example of the DRIP approach

4.3 Illustrative example of DRIP approach

To better understand how the algorithm works, we present an example of using the $DRIP$ approach as follows (Figure 4.3 shows the whole approach graphically).

Transactional data and concept hierarchy for the items are prerequisites for the approach. For simplicity, consider that we have transactional data for a certain retail store with 10 transactions consisting of 8 items. These are presented as input in Figure 4.3. Also, consider that we need to fill 25 slots in the retail stores by using itemsets from the transactions. We explain the placement of the itemsets in given slots using the $DRIP$ approach.

In the $DRIP$ approach, the first step is to build a $CDRI$ index, by using the proposed Algorithm 1. The constructed $CDRI$ index can also be located in the figure related to illustrative example. Each node in the $CDRI$ has $<$ itemset, frequency of itemset, price of itemset, drank of itemset, diverse net revenue$>$. Consider that the total number of levels was pre-decided to be 4. Notice that each level consists of itemsets of that size i.e., level k will have itemsets of size k. Also, notice that within a level, the itemsets are ordered using $diverse\_net\_revenue$.

The second step is to use the placement scheme for picking items from the $CDRI$ index and placing them in the slots. First, we compare $revenue\_per\_slot$ of top-nodes from each level of $CDRI$ index i.e., $\{L2 : 2.23, L3 : 0.57, L4 : 0.21\}$. Since the top revenue for level 2 is the highest, we place the itemset from level 2 in the slots first and remove it from $CDRI$ index. The updated per slot revenues for comparison becomes $\{L2 : 1.18, L3 : 0.57, L4 : 0.21\}$, and we pick the node from level 2 as it has the highest per slot revenue. This process is continued till all slots are filled. The step-wise placement is shown in step 2 in Figure 4.3.

4.4 Approaches implemented

In this section, we provide the various approaches as the variation of the proposed approach that we have implemented in order to observe the effectiveness of the proposed approach in an experimental setup.

4.4.1 Implementation details of the RIP approach

The Revenue based Itemset Placement (RIP) approach is a benchmark for comparing the effectiveness of the proposed approach. This approach described in more detail in section 3.3, orders itemsets based on their net revenue. This approach is motivated by existing work in [18], in the sense that motivation is similar but the placement scheme is updated. The RIP approach serves as a reference point for comparing the proposed and other approaches in terms of their ability to favor net revenue in ordering itemsets.

Construction of itemset index: The itemset index used in this case is called the Revenue Itemsets (RI) index. This is also a multi level index but only considers net revenue ($price \times frequency$) while ordering the itemsets.

Placement of itemsets:. Given a number of slots, the placement scheme keeps picking itemsets from RI index based on high *revenue per itemset* and placing them in the slots. This process is continued until all slots are filled.

Since the priority is given to net revenue here, this is expected to benefit the retailer revenue the most while not considering diversification of the itemset.

4.4.2 Implementation details of the DIP approach

The diversity based Itemset Placement (DIP) is the second approach and is based on the motivation of favoring only diverse items. The itemsets with higher $DRank$ are given higher priority while ordering elements.

Construction of itemset index: The itemset index used in this case is called the DI index (Diverse Itemsets index). This is also a multi level index where only diversity ($DRank$) is considered while ordering the itemsets.

Placement of itemsets: Once ordering is done, the rest is just the placement (as discussed in Algorithm 2).

This approach is expected to be least favorable for retailer revenue as it solely focuses on increasing diversity.

4.4.3 Implementation details of the DRIP approach

The third experiment implementation is Diverse Revenue based Itemset Placement $DRIP$ approach. This is the middle ground between the preceding two approaches.

Construction of itemset index:

Itemsets that have a higher $diverse\_net\_revenue$ (as defined in definition 7) are given priority while ordering. The building of $CDRI$ index pseudocode for this approach is given in Algorithm 1.

Placement of itemsets: Once the ordering is done, the placement method is the same as in Algorithm 2. Hence, the proposed approach $DRIP$ is a combination of building $CDRI$ index step and the placement step. Since, the focus here is on both $DRank$ and $revenue$, it is expected to improve both retailer revenue and diversity of itemsets to improve long term user satisfaction.

4.4.4 Implementation details of the HIP approach

Additionally, we observe the performance of combining the proposed approach and net revenue approach. Instead of building a single $CDRI$ index, this approach takes two $CDRI$ indices (one built in RIP approach and one built in $DRIP$ approach).

Construction of itemset index: Given the transactional database over a set of items, concept hierarchy over a set of items, and price details of items, we build both RI index and $CDRI$ index. The process to build $CDRI$ index is explained in the preceding section. The process to build RI index is similar to the process of building $CDRI$ index, except, the itemsets are indexed based on net revenue (as in kUI index, refer section 3.2).

Placement of itemsets: The HIP scheme uses a combination of RI index and $CDRI$ index to get itemsets for placement. During placement, it maintains vectors to top pointers and their $revenue\_per\_slot$ for both RI and $CDRI$. We introduce a variable called $RD\text{-}ratio$, which represents the ratio of the portion of slots to be filled from RI and $CDRI$ index. For example, if $RD\text{-}ratio$ is 0.3, 30% of given slots will be filled by itemsets from $CDRI$ and 70% of given slots will be filled by itemsets from RI.

This approach gives priority to both retailer revenue and diversity in the placement generation. One advantage observed here is that it gives control to the retailer to vary $RD\text{-}ratio$ in order to set a balance between retailer revenue and long term sustainability.

4.5 Performance evaluation

In this section, we present the experimental setup followed by the results of the experiments conducted.

4.5.1 Experimental setup

The experiments for this work were conducted on a 64-bit core i7-3537U CPU with 8GB of RAM running Ubuntu 18.04. The proposed diverse pattern mining approaches were implemented in Python 3.7.

The dataset used for the experiments was the Instacart Market Basket Analysis dataset [41], which contains over 3 million grocery orders from approximately 200,000 instacart users, with 49,688 items and 131,208 transactions. The average transaction length was 10.55. The dataset was divided into a training set and a test set in a ratio of 80:20, and the index for the proposed approaches (RIP, $DRIP$, HIP, and DIP) was built from the training set.

For the experiments, we used the concept hierarchy provided by the dataset, which organizes the items based on their department and category information. The dataset did not include price information for the items, so we generated prices for each item by randomly selecting a price bracket from the range [0.01, 1.0] and assigning a random price within that range to the item. The range [0.01, 1.0] was divided into six price brackets: [(0.01, 0.16), (0.17, 0.33), (0.34, 0.50), (0.51, 0.67), (0.67, 0.83), (0.84, 1.0)]. For each item, we randomly selected one of the price brackets and then assigned a random price within that range to the item. The price of an itemset was calculated as the sum of the prices of the items it contained.

Table 4.3 Parameters of performance evaluation

Parameters	***Default***	***Variations***
Total number of slots (Ts)	10*(10^3)	(1, 2, 4, 8, 12,16,20)*(10^3)
$RD\text{-}ratio$	0.3	0.0, 0.2, 0.4, 0.6, 0.8, 1.0
Top frequent itemsets	-	1,2,4,6,8,10,12,14,16,18

To evaluate the performance of the diverse pattern mining approaches, we followed a placement approach, as described in section 4.2. We simulated a real-life retail scenario by iterating over the transactions in the test set and calculating the revenue from the subsets of items that were placed in the given slots. We included the revenue of the corresponding subset towards TR, simulating a scenario where retailers place items on shelves and observe revenue only if the customer buys one of those items.

Table 4.3 summarizes the parameters used in the evaluation. The *Total number of slots (Ts)* refers to the number of slots allocated on the shelf. In the Hybrid Itemset Placement (HIP) approach, we fill a portion of slots with high net revenue itemsets from the RI index and the remaining portion with diverse revenue itemsets from the $CDRI$ index. We use the $RD\text{-}ratio$, which is the ratio of the number of slots to be filled from the RI index and the number of slots to be filled from the $CDRI$ index, to conduct experiments on the HIP approach. $Top\ frequent\ itemsets$ refers to the itemsets whose frequency of sales was observed with different approaches.

Performance metrics: The performance metrics for the experiments are Total revenue (TR) and $DRank$. Total revenue (TR) is the total retailer revenue for the test set. To calculate TR, we find the maximal set of items placed on the shelf for each transaction in the test set and add the revenue of that itemset to TR. $DRank$ is the mean $DRank$ of all the placed itemsets purchased by customers.

The experiments are performed on all four approaches: Revenue based Itemset Placement (RIP) approach, Diverse Revenue based Itemset Placement ($DRIP$) approach, Hybrid Itemset Placement (HIP) approach, and Diversity based Itemset Placement (DIP) approach. The RIP approach places high revenue itemsets in the given slots, using the RI (Revenue Itemsets) index. The $DRIP$ approach places itemsets using the $CDRI$ index. The HIP approach fills the portion of high net revenue itemsets from the RI index and the remaining portion with itemsets from $CDRI$ index. The DIP approach places only high $DRank$ itemsets using DI (Diverse Itemsets) index. For each of RI, $CDRI$, DI indexes, we fix the number of levels as 4 and the number of itemsets in each level as 5000.

4.5.2 Effect of variation in the total number of slots (Ts)

The proposed approach aims to improve the diverse revenue of the itemsets placed on the shelf while minimizing any compromise to the revenue generated, with a focus on long-term sustainability and user

satisfaction. In this section, we present the results of experiments performed by varying the total number of slots allocated on the shelf.

4.5.2.1 Effect of variations in the number of slots on total revenue

Figure 4.4 shows the variation of the total revenue for the Revenue Itemset Placement (RIP), Diverse Revenue Itemset Placement ($DRIP$), Hybrid Itemset Placement (HIP), and Diversity Itemset Placement (DIP) approaches. It can be seen that the revenue increases for all approaches with the number of slots, as more products are available for purchase and an increased number of customers can find the products in the retail store. Among the approaches, it is observed that the total revenue of the RIP approach dominates the other approaches. This is due to the fact that only high net revenue itemsets are placed in the RIP approach.

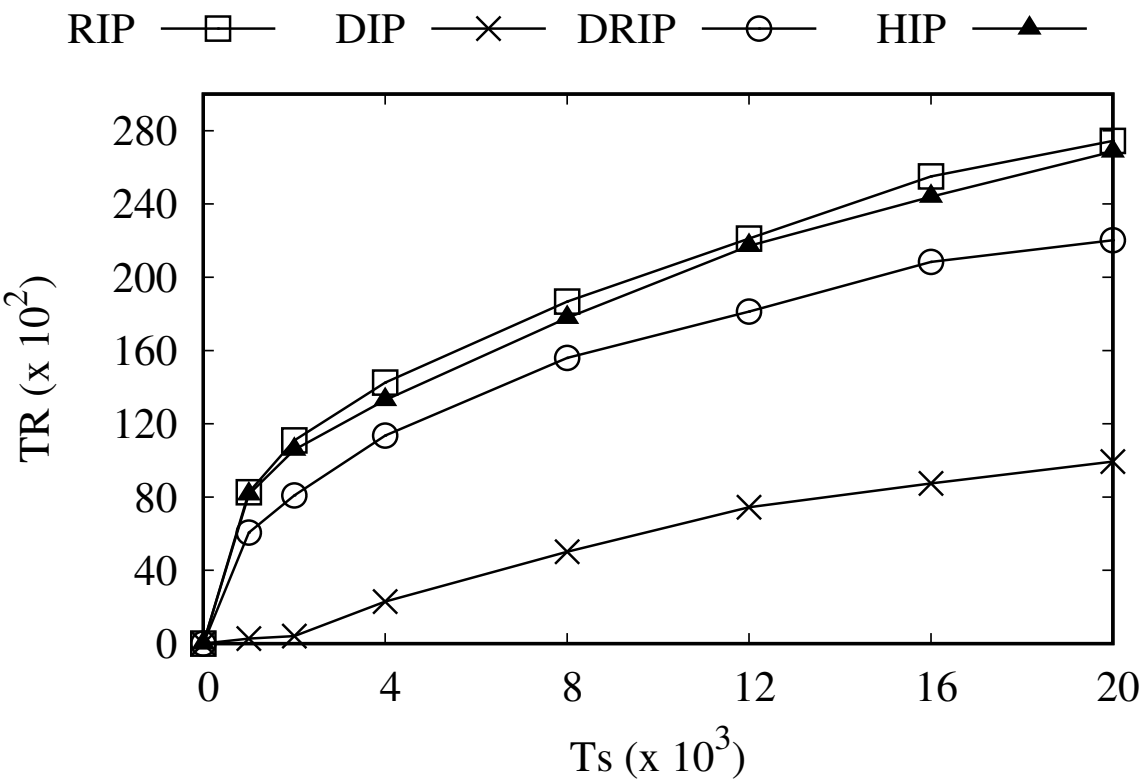

Figure 4.4 Effect of variations in the number of slots on total revenue

As expected, the performance of the DIP approach is the lowest among the four approaches, because the DIP approach places only high $DRank$ itemsets, ignoring the net revenue value of the itemsets. The total revenue of the DIP approach is not zero because the placed itemsets still have some revenue. The performance of the $DRIP$ approach is less than the RIP approach, but significantly more than the DIP approach. This is because the $DRIP$ approach places itemsets with both high net revenue

and high diversity. The revenue of the $DRIP$ approach is less than the RIP approach because high net revenue itemsets with low $DRank$ values are not placed in the slots. The Figure also shows the performance of the HIP approach, which was obtained using an RD-ratio of 0.3. This ratio places 70% of itemsets with the top net revenue value from the RIP approach and 30% of itemsets with both top net revenue and $DRank$ values from the $DRIP$ approach. The HIP approach performs better than the $DRIP$ approach because it places more of the top revenue itemsets, and is closer to the RIP approach in terms of total revenue.

4.5.2.2 Effect of variations in the number of slots on DRank

Figure 4.5 shows the variation of the $DRank$ values for the Revenue Itemset Placement (RIP), Diverse Revenue Itemset Placement ($DRIP$), Hybrid Itemset Placement (HIP), and Diversity Itemset Placement (DIP) approaches. It can be seen that, except for the DIP approach, the average $DRank$ value increases initially and becomes stable for the RIP, $DRIP$, and HIP approaches. For the DIP approach, the average $DRank$ value remains constant with the number of slots. Among the approaches, as expected, the average $DRank$ value is the highest for the DIP approach. The DIP approach also maintains a stable $DRank$ value with the increasing number of slots. This is because as the number of slots increases, the DIP approach places an increased number of high $DRank$ itemsets, resulting in the average $DRank$ not changing significantly.

As expected, it is observed that the $DRank$ performance of the Revenue Itemset Placement (RIP) approach is the lowest among the four approaches, because the RIP approach places only high net revenue itemsets, ignoring the $DRank$ value of the itemsets. The $DRank$ value of the RIP approach is not zero because it still places some itemsets with high $DRank$ values. The performance of the proposed Diverse Revenue Itemset Placement ($DRIP$) approach is observed to be less than the Diversity Itemset Placement (DIP) approach, but significantly more than the RIP approach. This is because the $DRIP$ approach contains itemsets with both high $DRank$ values and high net revenue values. The $DRank$ value of the $DRIP$ approach is less than the DIP approach because high $DRank$ itemsets with low net revenue values are not placed in the slots. The figure also shows the performance of the Hybrid Itemset Placement (HIP) approach with an $RD\text{-}ratio$ of 0.3, which places 70% of itemsets with top net revenue values from the RIP approach and 30% of itemsets with both top net revenue and high $DRank$ values from the $DRIP$ approach. Since itemsets with top $DRank$ values are excluded, the $DRank$ value of the HIP approach is less than the $DRIP$ approach.

From the analysis of these two metrics, it can be seen that the $DRIP$ approach performs better when considering the task of simultaneously increasing $DRank$ and total revenue. It can also be observed that the HIP approach, if tuned well, can perform closer to the RIP approach in terms of total revenue while still providing the advantage of introducing diversity.

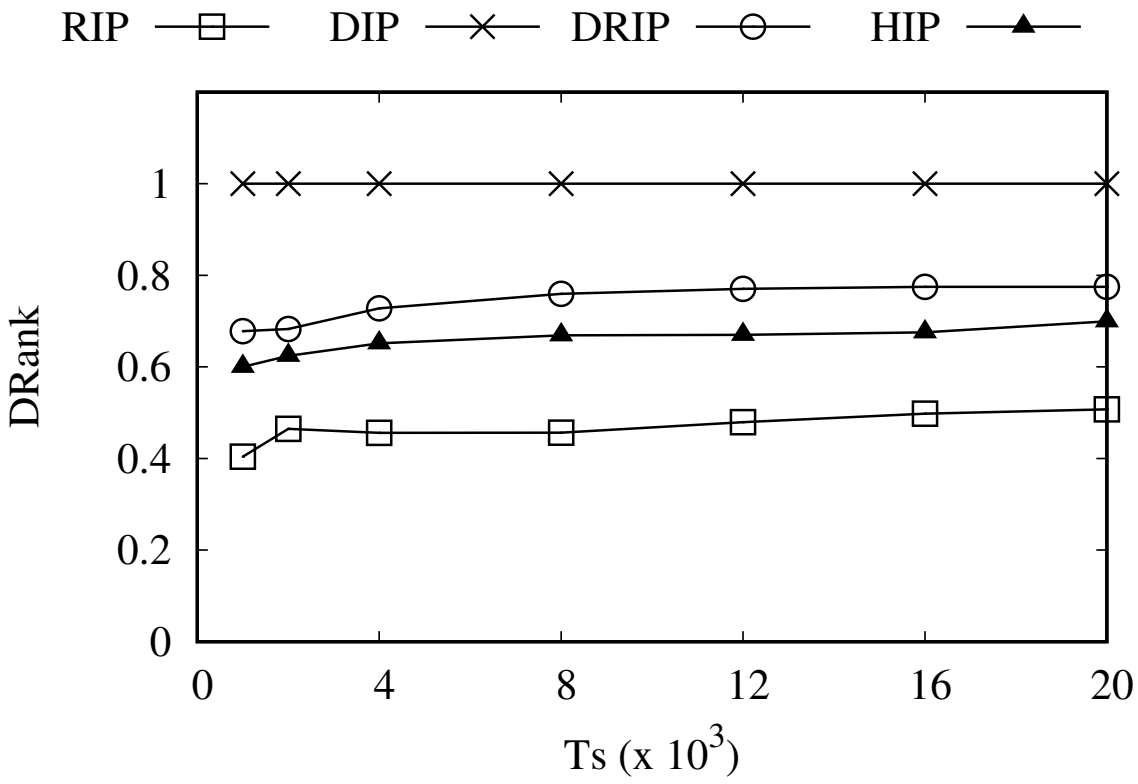

Figure 4.5 Effect of variations in the number of slots on DRank

4.5.3 Effect of variation in $RD\text{-}ratio$

For the Revenue Itemset Placement (RIP), Diversity Itemset Placement (DIP), Diverse Revenue Itemset Placement ($DRIP$), and Hybrid Itemset Placement (HIP) approaches, Figure 4.6 shows the variations of total revenue and Figure 4.7 shows the variations of $DRank$ value as the $RD\text{-}ratio$ is varied. In this experiment, the number of slots is fixed at 12000. It is notable that the $RD\text{-}ratio$ does not affect the RIP, $DRIP$, and DIP approaches because the number of slots is fixed.

4.5.3.1 Effect of variations in $RD\text{-}ratio$ on total revenue

Figure 4.6 shows the results of total revenue as the $RD\text{-}ratio$ is varied. It can be observed that the HIP approach exhibits similar performance to the RIP approach at $RD\text{-}ratio = 0$, and it is similar to the $DRIP$ approach at $RD\text{-}ratio = 1$. As the $RD\text{-}ratio$ is varied from 0 to 0.4, the total revenue of HIP is equal to that of RIP, indicating that the top-revenue itemsets are contributing significantly to the total revenue. As the $RD\text{-}ratio$ is increased further, the performance of HIP starts to decline and becomes equal to that of $DRIP$ at $RD\text{-}ratio = 1$. This indicates that after $RD\text{-}ratio = 0.4$, the HIP

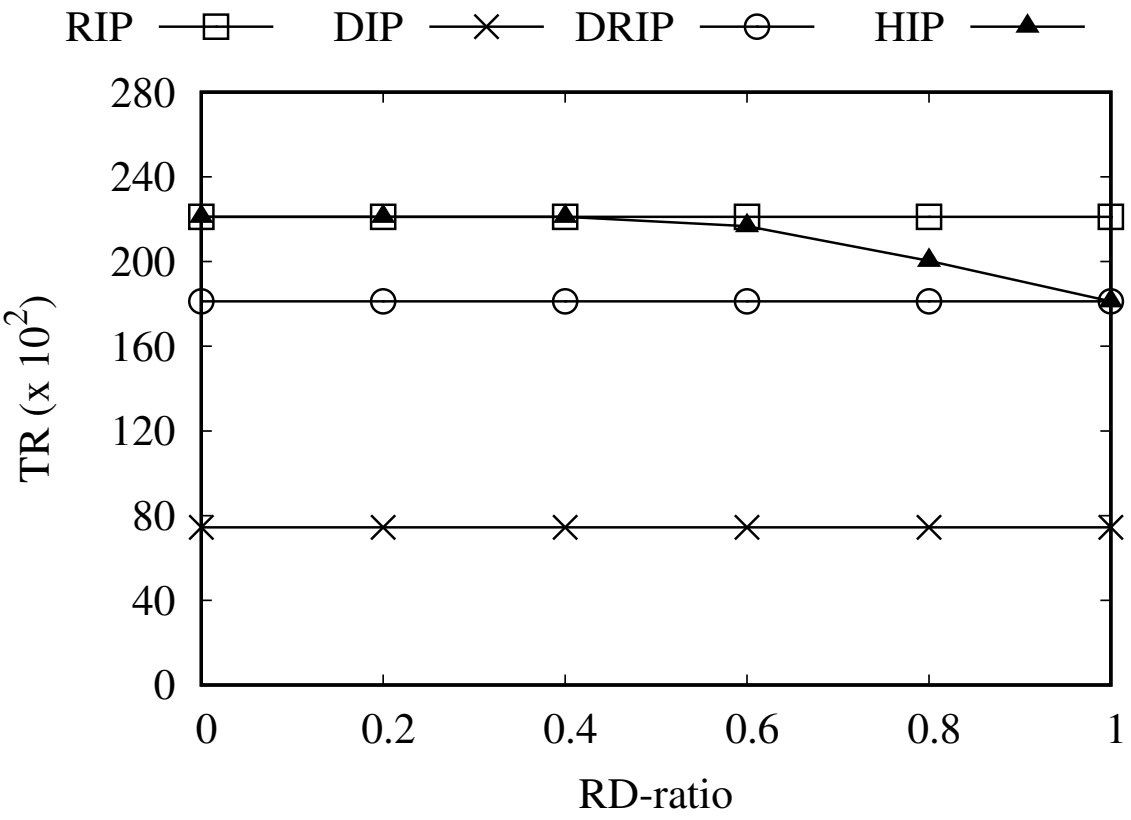

Figure 4.6 Effect of variation in $RD\text{-}ratio$ VS total revenue

approach starts to include high $DRank$ itemsets in the slots, resulting in a performance similar to that of the $DRIP$ approach.

4.5.3.2 Effect of variations in $RD\text{-}ratio$ on DRank

Figure 4.7 shows the result of $DRank$ values as the $RD\text{-}ratio$ is varied. The behavior HIP is similar to RIP at $RD\text{-}ratio = 0$, and it is similar to $DRIP$ at $RD\text{-}ratio = 1$. It can be observed that as we vary $RD\text{-}ratio$ from 0 to 0.3, the $DRank$ value of HIP is equal to RIP. Gradually, the $DRank$ performance of HIP starts increasing from $RD\text{-}ratio = 0.3$ and it reaches $DRIP$ at $RD\text{-}ratio = 1$. This indicates that after $RD\text{-}ratio > 0.3$, the high diversity itemsets are added to the slots, and hence, the performance of HIP becomes equal to $DRIP$.

Based on the results shown in Figure 4.6 and Figure 4.7, we can conclude that HIP allows for a flexible trade-off between total revenue and $DRank$. HIP provides the option to achieve total revenue similar to that of RIP by decreasing $RD\text{-}ratio$ to a reasonable extent while also providing the option to significantly improve $DRank$, subject to a trade-off in terms of revenue.

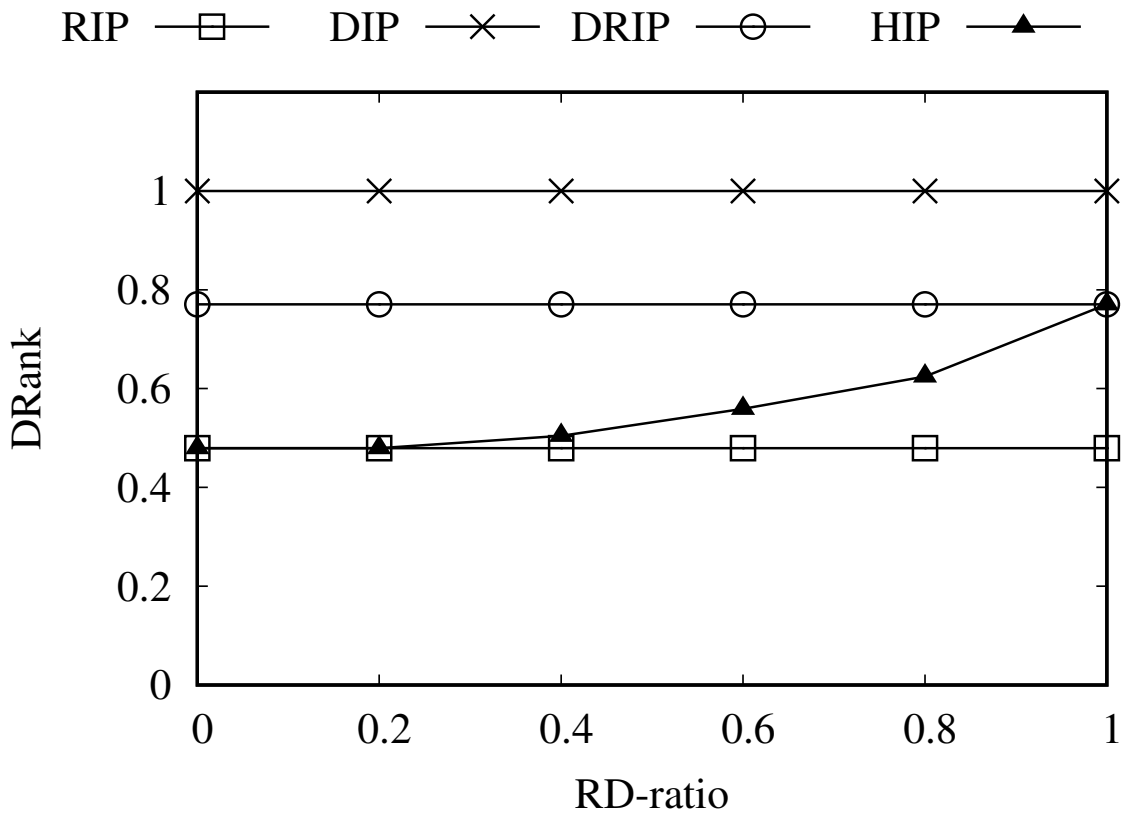

Figure 4.7 Effect of variation in $RD\text{-}ratio$ VS DRank

4.5.4 Effect on top frequent items

We also compared the performance of the proposed approach to a revenue based approach by examining the frequency of the top sold items in a store. To do this, we selected the top eighteen most sold items and compared the number of times they were sold in a revenue based approach versus a diverse revenue based approach. The results of this experiment are shown in Figure 4.8. The X-axis represents the top sold item numbers and the Y-axis shows the number of times the item was sold.

From the results, we can see that the revenue based approach is more focused on the sales of the most frequent items, with some items contributing significantly to the total revenue of the store. In contrast, the diverse revenue based approach helps to spread out the contribution, satisfying the demands of a wider range of customers. This means that the retail store's revenue is not solely dependent on a few items but caters to a diverse set of customer interests, promoting long-term sustainability.

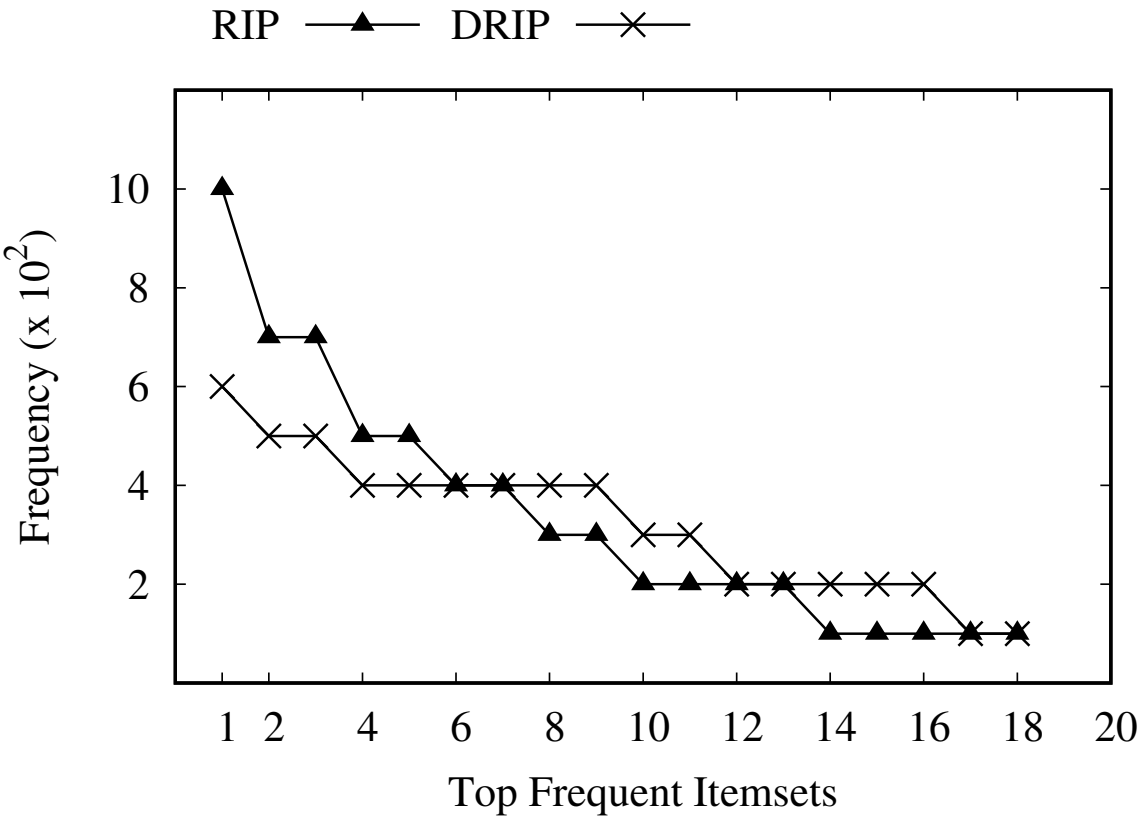

Figure 4.8 Effect of variation in top frequent itemsets VS frequency

4.6 Summary

In this chapter, we first explained the basic idea of the proposed approach. Then, we gave a detailed explanation of the proposed indexing scheme $CDRI$ (Concept hierarchy based Diversity and Revenue Itemsets) index. Further, we explained the placement approach which uses the itemset index. We also present a detailed illustrative example to re-instate how this approach can be used in real world scenarios.

Next, we provide details of all the approaches we implement as a variation of the proposed scheme for the purpose of evaluating the effectiveness of the approach. Next, we provide details about the experimental setup and present the experimental results along with observations. We evaluate the four approaches we have implemented on the basis of variation in total revenue and $DRank$ as the number of slots are increased. We also provided insight into the variation of $RD\text{-}ratio$ and how HIP approach can update its behavior thus providing flexibility to the user. The results provide confirmation of the effectiveness of the proposed approach when compared to the traditional approach of just using revenue when considering the placement of items.

Chapter 5

Summary, Conclusion and Future work

In this chapter, we give out the summary and conclusion for the work presented in this thesis. We end this chapter by discussing possible directions for future work.

5.1 Summary

Retail industry is an integral part of a nation's economy. Over the last several decades, the increased demand for retail stores has led to the advent of mega retail stores, which store several thousand items and span massive areas. Existing research in the retail domain indicates that product placement in retail stores considerably impacts the sales of the items and revenue generated by the retail stores. The task of determining the placement of products on the shelves of retail stores in order to maximize revenue serves as a challenging problem with significant business value. As a result, several research efforts have been presented in the domain focusing on product placement in retail stores to improve retailer revenue. The traditional approaches employing dynamic programming and optimization methods have been proposed for product assortment and shelf space allocation in retail stores. However, such methods depend on a large number of parameters requiring high computing power. Also, these methods are inadequate to leverage the massive amount of historical data presented for retail stores in terms of customer purchase transactions. Data mining has played a significant role in the retail industry as it helps to learn the buying pattern of consumers and can deal with the massive amount of data. Research efforts have been made that employ data mining techniques like frequent pattern mining and utility pattern mining towards improving retail product placement.

Several approaches of *pattern mining*, including frequent pattern mining and utility mining, have been explored for improving product placement in retail stores. *Pattern mining* indicates extraction of patterns of items (itemsets) from a transactional database. *Frequent pattern mining* is the term for extracting patterns based on the frequency (number of times) of items occurring together. But, the frequency is an objective measure and is not able to capture the business value (that can be price, usefulness, etc.) that is assigned to items. To help with such business metrics, *utility mining* techniques are presented that extends the concept of frequent pattern mining by considering the utility (e.g., price,

satisfaction, weight, profit, etc) for extracting high-utility itemsets. A number of research efforts have been proposed for improving product placement in retail stores by employing these data mining based approaches.

Another strategy to improve product placement is to provide customers with a wider (diverse) variety of items so that they have more choices of the items that are available to them for purchase. This variety can be in terms of different brands for the same product and in terms of different products. There have been efforts to define diversity objectively. One such method is to use concept hierarchy to determine diversity for a set of items. Concept hierarchy is a structure representing the relationship between items and their categories, where the items occupy the leaf nodes and intermediate nodes represent the categories of the items. The retail stores inherently have a structure of categories, so the idea of using a concept hierarchy can be useful. However, no existing approaches use utility and diversity to improve product placement in retail stores.

In this thesis, we extend the existing utility mining based approach for product placement to include concept hierarchy driven diversity in order to solve the problem of product placement in retail stores. We have made the following significant contributions which positively impact the product placement in retail stores. We introduce the problem of concept hierarchy based diverse itemset placement in retail stores. We present a framework and schemes for facilitating efficient retrieval of the diverse top-revenue itemsets based on concept hierarchy. We present a method for using the retrieval structure to place itemsets in retail store shelves. The research aims to improve product placement in retail stores through the integration of these approaches.

This thesis also presents a performance evaluation of the proposed approach for optimizing product placement in retail stores through the integration of utility mining and concept hierarchy based diversity. The evaluation has conducted the experiment on a real dataset, namely instacart market basket analysis retail dataset (containing 49,688 items and 3,214,874 transactions). Our performance evaluation of the dataset demonstrates the real world effectiveness of the proposed scheme in terms of total revenue and diversity of the itemsets placed.

5.2 Conclusion

Product placement in retail stores is a challenging problem and has garnered significant research attention. Several methods have been proposed in an attempt to solve the problem ranging from traditional methods to data mining based methods. The focus of existing data mining based approaches has been to improve the revenue of retail stores by placing high utility items. This thesis highlights the importance of considering diversity in product placement to attract a wider audience and ensure the sustainability of retail stores. By leveraging concept hierarchy to objectively measure diversity, this research proposes a new approach that combines utility mining with diversity to optimize product placement in retail stores.

In this thesis, we introduced an approach for optimizing product placement in retail stores by combining utility mining with concept hierarchy-based diversity. By incorporating this diversity measure

into the utility calculation, the proposed approach demonstrates effectiveness in increasing diversity while maintaining revenue, as shown through experiments on a real dataset.

5.3 Future work

The problem of itemset placement has multiple facets and scope for improvement. Here, we list some of the methods which can be explored in future work for the proposed approach:

1. We have targeted a scoped-down problem of itemset placement in retail stores. One potential direction for future work is to extend the proposed approach to consider seasonal patterns in customer purchasing behavior and adapt product placement decisions accordingly. This could involve incorporating a seasonal view of customer purchase transactions to make more informed placement decisions in real-world retail stores.

2. Another possible direction for future research is to explore alternative approaches for determining the mutual relationship between items in the process of extracting item sets. Frequency and frequency-driven utility have been used in previous research, but these approaches have limitations such as the generation of redundant patterns and the inability to effectively capture correlations among items. Correlated pattern mining, which measures the correlation of itemsets, could be a promising alternative interestingness measure to consider in this context.

3. The proposed approach considers that different shelves (lower, upper, etc.) of retail stores have the same effect on itemset placement. Practically, an itemset placed on lower, middle, upper, or customer counter shelve may result in different visibility to customers and thus result in different outcomes in terms of revenue. We can extend our approach to capture this variation in the visibility of retail shelves.

Chapter 6

Related Publications

- Gaur, Pooja, P. Krishna Reddy, M. Kumara Swamy, and Anirban Mondal. "A Revenue-Based Product Placement Framework to Improve Diversity in Retail Businesses." In International Conference on Big Data Analytics, pp. 289-307. Lecture Notes in Computer Science, vol 12581. Springer, Cham, 2020.

Bibliography

[1] A. Achar, A. Ibrahim, and P. Sastry. Pattern-growth based frequent serial episode discovery. *Data & Knowledge Engineering*, 87:91–108, 2013.

[2] A. Achar, S. Laxman, and P. Sastry. A unified view of the apriori-based algorithms for frequent episode discovery. *Knowledge and Information Systems*, 31(2):223–250, 2012.

[3] C. C. Aggarwal, Y. Li, J. Wang, and J. Wang. Frequent pattern mining with uncertain data. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 29–38. ACM New York, NY, United States, 2009.

[4] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216. ACM, 1993.

[5] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499. ACM, 1994.

[6] E. E. Anderson. An analysis of retail display space: theory and methods. *Journal of Business*, pages 103–118, 1979.

[7] B. Becker. Adult. UCI Machine Learning Repository, 1996.

[8] B. Bentalha, A. Hmioui, and L. Alla. The digitalization of the supply chain management of service companies: a prospective approach. In *Proceedings of the 4th International Conference on Smart City Applications*, pages 1–8. ACM, 2019.

[9] N. Borin and P. Farris. A sensitivity analysis of retailer shelf management models. *Journal of Retailing*, 71(2):153–171, 1995.

[10] N. Borin, P. W. Farris, and J. R. Freeland. A model for determining retail product category assortment and shelf space allocation. *Decision Sciences*, 25(3):359–384, 1994.

[11] K. Bradley and B. Smyth. Improving recommendation diversity. In *Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science, Maynooth, Ireland*, volume 85, pages 141–152. Citeseer, 2001.

[12] T. Brijs, G. Swinnen, K. Vanhoof, and G. Wets. Using association rules for product assortment decisions: A case study. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 254–260. ACM, 1999.

[13] W. Brown and W. Tucker. The marketing center: Vanishing shelf space. *Atlanta Economic Review*, 11(10):9–13, 1961.

[14] A. Bultez and P. Naert. Sh. arp: Shelf allocation for retailers' profit. *Marketing Science*, 7(3):211–231, 1988.

[15] A. Bultez, P. Naert, E. Gijsbrechts, and P. V. Abeele. Asymmetric cannibalism in retail assortments. *Journal of Retailing*, 65(2):153, 1989.

[16] P. Chaudhary, A. Mondal, and P. K. Reddy. A flexible and efficient indexing scheme for placement of top-utility itemsets for different slot sizes. In *International Conference on Big Data Analytics*, pages 257–277. Springer, 2017.

[17] P. Chaudhary, A. Mondal, and P. K. Reddy. A diversification-aware itemset placement framework for long-term sustainability of retail businesses. In *International Conference on Database and Expert Systems Applications*, pages 103–118. Springer, 2018.

[18] P. Chaudhary, A. Mondal, and P. K. Reddy. An efficient premiumness and utility-based itemset placement scheme for retail stores. In *International Conference on Database and Expert Systems Applications*, pages 287–303. Springer, 2019.

[19] P. Chaudhary, A. Mondal, and P. K. Reddy. An improved scheme for determining top-revenue itemsets for placement in retail businesses. *International Journal of Data Science and Analytics*, 10(4):359–375, 2020.

[20] M.-C. Chen and C.-P. Lin. A data mining approach to product assortment and shelf space allocation. *Expert Systems with Applications*, 32(4):976–986, 2007.

[21] Y.-C. Chen, W.-C. Peng, and S.-Y. Lee. Ceminer–an efficient algorithm for mining closed patterns from time interval-based data. In *2011 IEEE 11th International Conference on Data Mining*, pages 121–130. IEEE, 2011.

[22] M. Corstjens and P. Doyle. A model for optimizing retail space allocations. *Management Science*, 27(7):822–833, 1981.

[23] M. Corstjens and P. Doyle. A dynamic model for strategically allocating retail space. *Journal of the Operational Research Society*, 34(10):943–951, 1983.

[24] H. Enghoff. What is taxonomy? an overview with myriapodological examples. *Soil organisms*, 81(3):441–451, 2009.

[25] M. Etgar and D. Rachman-Moore. Market and product diversification: the evidence from retailing. *Journal of Marketing Channels*, 17(2):119–135, 2010.

[26] T. Flamand, A. Ghoniem, M. Haouari, and B. Maddah. Integrated assortment planning and store-wide shelf space allocation: An optimization-based approach. *Omega*, 81:134–149, 2018.

[27] P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas. A survey of sequential pattern mining. *Data Science and Pattern Recognition*, 1(1):54–77, 2017.

[28] H. K. Gajjar and G. K. Adil. A dynamic programming heuristic for retail shelf space allocation problem. *Asia-Pacific Journal of Operational Research*, 28(02):183–199, 2011.

[29] M. Hamada, K. Tsuda, T. Kudo, T. Kin, and K. Asai. Mining frequent stem patterns from unaligned rna sequences. *Bioinformatics*, 22(20):2480–2487, 2006.

[30] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, 2007.

[31] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *ACM Sigmod Record*, 29(2):1–12, 2000.

[32] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *ACM sigmod record*, 29(2):1–12, 2000.

[33] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, 2004.

[34] P. Hansen and H. Heinsbroek. Product selection and space allocation in supermarkets. *European Journal of Operational Research*, 3(6):474–484, 1979.

[35] M. A. Hariga, A. Al-Ahmari, and A.-R. A. Mohamed. A joint optimisation model for inventory replenishment, product assortment, shelf space and display area allocation decisions. *European Journal of Operational Research*, 181(1):239–251, 2007.

[36] V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. *Acm Sigmod Record*, 25(2):205–216, 1996.

[37] C. Hart. The retail accordion and assortment strategies: an exploratory study. *The International review of Retail, Distribution and Consumer Research*, 9(2):111–126, 1999.

[38] L. Henderson et al. Automated text classification in the dmoz hierarchy. Technical report, TR, 2009.

[39] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[40] R. A. Huebner. Diversity-based interestingness measures for association rule mining. *Proceedings of ASBBS*, 16(1), 2009.

[41] jeremy stanley, M. Risdal, sharathrao, and W. Cukierski. Instacart market basket analysis, 2017.

[42] K. A. Kaufman and R. S. Michalski. A method for reasoning with structured and continuous attributes in the inlen-2 multistrategy knowledge discovery system. In *KDD*, pages 232–237. Springer, 1996.

[43] M. Kumara Swamy and P. Krishna Reddy. Improving diversity performance of association rule based recommender systems. In *Database and Expert Systems Applications*, pages 499–508. Springer, 2015.

[44] M. Kumara Swamy and P. Krishna Reddy. A model of concept hierarchy-based diverse patterns with applications to recommender system. *International Journal of Data Science and Analytics*, 10(2):177–191, 2020.

[45] M. Kumara Swamy, P. Krishna Reddy, and S. Bhalla. Association rule based approach to improve diversity of query recommendations. In *International Conference on Database and Expert Systems Applications*, pages 340–350. Springer, 2017.

[46] M. Kumara Swamy, P. K. Reddy, and S. Srivastava. Extracting diverse patterns with unbalanced concept hierarchy. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 15–27. Springer, 2014.

[47] J. Lee and R. Goodwin. Ontology management for large-scale e-commerce applications. In *International Workshop on Data Engineering Issues in E-Commerce*, pages 7–15. IEEE, 2005.

[48] H.-F. Li and S.-Y. Lee. Mining top-k path traversal patterns over streaming web click-sequences. *Journal of Information Science & Engineering*, 25(4), 2009.

[49] K.-C. Lin, I.-E. Liao, and Z.-S. Chen. An improved frequent pattern growth method for mining association rules. *Expert Systems with Applications*, 38(5):5154–5161, 2011.

[50] M. Liu and J. Qu. Mining high utility itemsets without candidate generation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 55–64. ACM, 2012.

[51] Y. Liu, W.-k. Liao, and A. Choudhary. A two-phase algorithm for fast discovery of high utility itemsets. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 689–695. Springer, 2005.

[52] S. M. McNee, J. Riedl, and J. A. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems*, pages 1097–1101. ACM, 2006.

[53] A. Mondal, R. Mittal, P. Chaudhary, and P. K. Reddy. A framework for itemset placement with diversification for retail businesses. *Applied Intelligence*, pages 1–19, 2022.

[54] E. Özkural and C. Aykanat. A space optimization for fp-growth. In *FIMI*. Citeseer, 2004.

[55] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *International Conference on Database Theory*, pages 398–416. Springer, 1999.

[56] B. Smyth and P. McClave. Similarity vs. diversity. In *International Conference on Case-based Reasoning*, pages 347–361. Springer, 2001.

[57] R. Srikant and R. Agrawal. Mining generalized association rules. In *In Proceedings of the 21th International Conference on Very Large Data Bases*, pages 407–419. IBM Research Division Zurich, 1995.

[58] V. S. Tseng, B.-E. Shie, C.-W. Wu, and S. Y. Philip. Efficient algorithms for mining high utility itemsets from transactional databases. *IEEE Transactions on Knowledge and Data Engineering*, 25(8):1772–1786, 2012.

[59] V. S. Tseng, C.-W. Wu, P. Fournier-Viger, and S. Y. Philip. Efficient algorithms for mining the concise and lossless representation of high utility itemsets. *IEEE Transactions on Knowledge and Data Engineering*, 27(3):726–739, 2014.

[60] V. S. Tseng, C.-W. Wu, B.-E. Shie, and P. S. Yu. Up-growth: an efficient algorithm for high utility itemset mining. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 253–262. ACM, 2010.

[61] S. M. Wigley. A conceptual model of diversification in apparel retailing: the case of next plc. *Journal of the Textile Institute*, 102(11):917–934, 2011.

[62] W. Woolberg, W. Street, and O. Mangasarian. Breast cancer wisconsin (diagnostic). UCI Machine Learning Repository, 1995.

[63] M.-H. Yang. An efficient algorithm to allocate shelf space. *European Journal of Operational Research*, 131(1):107–118, 2001.

[64] M.-H. Yang and W.-C. Chen. A study on shelf space allocation and management. *International Journal of Production Economics*, 60:309–317, 1999.

[65] H. Yao and H. J. Hamilton. Mining itemset utilities from transaction databases. *Data & Knowledge Engineering*, 59(3):603–626, 2006.

[66] H. Yao, H. J. Hamilton, and L. Geng. A unified framework for utility-based measures for mining itemsets. In *Proc. of ACM SIGKDD 2nd Workshop on Utility-Based Data Mining*, pages 28–37. Citeseer, 2006.

[67] F.-y. Ye, J.-d. Wang, and B.-l. Shao. New algorithm for mining frequent itemsets in sparse database. In *2005 International Conference on Machine Learning and Cybernetics*, volume 3, pages 1554–1558. IEEE, 2005.

[68] L. Yijun. Concept hierarchy in data mining: Specification, generation and implementation. *MSc Thesis*, 1997.

[69] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, pages 22–32. ACM New York, NY, United States, 2005.

[70] J. Zou, J. Xiao, R. Hou, and Y. Wang. Frequent instruction sequential pattern mining in hardware sample data. In *2010 IEEE International Conference on Data Mining*, pages 1205–1210. IEEE, 2010.

[71] F. S. Zufryden. A dynamic programming approach for product selection and supermarket shelf-space allocation. *Journal of the Operational Research Society*, 37(4):413–422, 1986.