2015-05-27

A Graph Based Approach for Making Recommendations Based on Multiple Data Sources

Dhaliwal, Sukhpreet

UNIVERSITY OF CALGARY

A Graph Based Approach for Making Recommendations Based on Multiple Data Sources

by

Sukhpreet Kaur Dhaliwal

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF COMPUTER SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE

CALGARY, ALBERTA

MAY, 2015

Abstract

Recommendation system is an information filtering system that predicts customer preferences. Customer preferences are extracted through analyzing the behaviour patterns of customers from multiple data sources. Graph-based models play an important role in recommendation systems to extract the customer preferences from multiple data sources. However, graph-based models have been rarely used in traditional recommendation systems. The main objective of this thesis is to use a graph-based recommender system that uses multiple data sources. A graph-based hybrid recommender model is developed to integrate content-based, collaborative filtering and association rule mining techniques. Moreover, the PageRank algorithm is used to produce a ranked list of recommendation.

Our analysis on a Retail store dataset shows the impact of using multiple data sources on the accuracy of a recommender system while handling the sparsity problem. Usage of demographic information of customers remedies the cold start problem. Grouping the products based on product type produced better results and it also showed the impact of using the different level of product taxonomy. Additionally, assembling content-based, collaborative filtering and association rule mining also showed many improvements in results. Moreover, indirect connections improve the coverage of our recommender system.

Acknowledgements

I would like to take this opportunity to thank all the people who made this work possible. My sincere thanks and appreciation goes to Dr. Reda Alhajj and Dr. Jon Rokne for supervising and inspiring me through out the whole journey of my graduate study.

I would also like to thank my friend, Khobaib Zaamout for refining the thesis and providing suggestions. Last but not the least, I would like to thank my brother Manmeet Dhaliwal and my friend Charanjeet Kaur for their endless motivation, assistance and guidance during the process of completing this thesis.

Dedication

To my beloved family

Table of Contents

List of Tables

List of Figures and Illustrations

Chapter One: **Introduction**

Decision-making is part of our daily life. Sometimes the decision making process may not be easy due to the lack of expertise or the availability of alternatives. People usually feel more comfortable to seek recommendations from others who are seen as more experienced in the domain of knowledge related to the task to be accomplished. However, the personal communication based method for seeking recommendation is impractical in the presence of too many alternatives.

Shopping has been highly influenced by the wide spread of the Internet and the Web. This leads to e-stores with the availability of several options of goods and services [1]. Therefore, we need a software tool which automate the decision making process in the presence of too much information. The software tools that use the various techniques to automate the decision-making procedure are called Recommendation systems [2].

1.1 PROBLEM STATEMENT AND MOTIVATION

The rapid development of the Internet has increased the availability of goods and services choices. This has led to information overload and the difficultly of making decision in the presence of too many choices [1]. The availability of large numbers of alternatives reduces the usability of the provided information since the customer can't possibly go through all the available alternatives. Consequently, we need an information filtering system. A recommendation system is an information filtering system. The aim of a recommendation system is to discovered irrelevant information and provided a customer with relevant information corresponding to his or her personal preferences. Recommendation system not only customize

the information according to the preferences of the target customer, it provides a platform for marketing strategies such as cross selling.

Recommendation systems predict recommendations using three basic steps: they obtain preferences from raw data of customers, computes recommendations using certain criteria or techniques, and outputs a list of recommendations to a customer [1]. Recommendation systems have a wide range of application domains ranging from health consultation to marketing in e-commerce.

The popularity of purchasing products from online stores is increasing because e-shopping provides a convenient way for a customer to buy a product. This results in large amounts of data. In order to provide personalized recommendation to a customer, recommendation techniques encounter the problem of handling large amount of data in an efficient way.

Recommendation systems are generally collecting data in three main subcategories: users (customers), items (products), and transactions [2]. Since recommendation applications are very diverse, data collected within three main subcategories can be very diverse as well. However, the essential purpose of customer related data category is to collect the information about a customer such as demographic data, browsing patterns, etc. Similarly, a product related data have characteristics of the product such as authors, publishers, price, genre, and many others for a book product record for example. Transactions record interactions between customers and a system [2]. Transactional dataset not only define the characteristics of customers and products, but also associations between customers and products. Therefore, recommendation systems handle the problem of using various types of data to get customers' preferences accurately.

Another commonly raised problem in recommendation systems is data sparsity. Data sparsity arises due to lack of information regarding customers' preferences. Data sparsity is a problem of finding reliable similar customers to a target customer, since most of the customers only rate or purchase small amounts of products [3]. Cold-start is a problem of insufficient available information regarding customer preferences due to small number of products rated by a target customer [3]. The availability of alternative stores also lead to the sparsity problem since customers only have a few transactions per store. Additionally, the availability of alternatives choices for a product decreases the number of associations between products and customers since customers tend to buy different products. Lacking sufficient information in recommendation systems lead to inaccurate and unreliable recommendations. The two widely used strategies to handle sparsity and cold start problems are, adding the missing or additional information regarding customer preferences and making better use of existing information using hybrid approaches [3].

The purpose of a recommendation system is to generate a list of the useful product for a customer to increase the utility of the customer's experience when selecting products from a given set of products [27]. However, finding a technique to estimate the utility of a product to a customer is not only depending on recommendation techniques but also on the information or data available. The most commonly used recommendation techniques in recommendation systems are content-based recommendation systems and collaborative filtering recommendation systems. Most recommendation systems use the hybrid techniques, which are the combination of content based, and collaborative filtering techniques. Hybrid techniques improve the performances and accuracy of recommendation systems [10]. Additionally, association rule mining has been used in recommendation systems for marketing techniques to increase the

revenue as well as customer satisfaction [23]. A hybrid solution makes better use of various types of data or information regarding customer preferences and produces personalized recommendations. However, a hybrid recommendation system uses a combination of many techniques and complex representation to fully utilize the product, customer and transaction information. This lead to a very complicated system and it needs significant effort to correctly incorporate various types of information into an appropriate representation. Some researchers proposed models contain all information sources and apply inductive learning techniques to find preferred recommendations [3]. Some researches attempted to incorporate differed types of information sources in customers' representations and products' representations. Therefore, a structural approach to combine various recommendation techniques and data sources lead to the exploration of graph based recommendation methods [22].

In [20] paper, the authors describe a generic graph-based recommendation approach to integrate the content-based approach with the collaborative-filtering approach in the context of digital libraries. Books and customers are represented in an extended graph and it incorporates book-to-book correlations, customer-to-customer correlations and book-to-customer correlations. They used a dataset obtained from a major Chinese online bookstore in Taiwan as exploratory domain because the application is generic and characteristics are similar to those of digital libraries. The graph based recommender system not only integrate different techniques but also use various types of data sources such as customer demographic information, customer purchase history information, book content information and book attribute information. A similarity measurement is used to define weights of links in the graph. The graph search technique becomes the recommendation activity. Although, the graph based recommendation systems are flexible and comprehensive, the graph search activity becomes very complex to produce a ranked

list of recommendation products. Additionally, the influence from indirect connections is not significantly explored yet. Therefore, the influence transfer algorithm such as PageRank makes a graph search activity efficient [37].

A customer-product graph bears two properties: propagation and attenuation, which are two key features for PageRank algorithm. The propagation property is that the relatedness of the nodes propagates through following the links, and the attenuation property is that the propagation strength decreases as the propagation goes further from the starting node. Larry Page has developed PageRank algorithm that finds the importance of a website by counting the number and quality of links to the website page [36]. We follow a similar way but leverage PageRank algorithm for recommending by exploiting customers and products links. Specifically, this is done by applying PageRank to the graph where graph is created using integration of many techniques and data sources. PageRank improves the representation of links and nodes and discovers trending and popular products. Additionally, the recommendation activity for a target customer is to extract the customer sub network and apply PageRank algorithm to produce a ranked list of products.

1.2 OBJECTIVE OF OUR RESEARCH

This research has three main objectives: 1. Integrating different types of data or information available related to customer preferences, 2. Integrating various recommendation techniques, and 3. Find the impact of indirect influence transfer. The three main objectives are designed to handle sparse datasets. In other words, a comprehensive and flexible recommendation model is needed in order to handle the sparsity and cold start problem while producing personalized recommendations. The benefits of each objective are explained below.

Customer preference information is available from various sources and this creates the problem of extracting customer preferences information from different types of data. Diversity in data leads to diversity in the relationship between customers and products. The similarity between two customers based on demographic information is different from similarity of customers based on purchase history. Similarly, similarity of products based on product category is different from the association relation between products that frequently bought together. In the sparse dataset, we can't ignore any type of available information. Therefore, our research objective is to integrate different types of customer preferences information in order to create personalized recommendation system.

The second objective of our research is to derive a way to efficiently use the extracted customer preferences information using a combination of recommendation techniques such as content-based and collaborative filtering. Content-based recommendation techniques explore products' relationships and recommend similar products to the customer's previous purchased products. However, content based can't explore the alternatives which may be potential recommendations related to customer's preferences since these have not been purchased by the customer in the past. The collaborative filtering based algorithms can handle this problem since they make predictions about a customer's preferences by compiling preferences from several customers. This technique leads to the exploration of products that are not purchased by the target customer for recommendations for new customers. Additionally, the frequent purchase of same product set creates the association rules in products since those products are likely to get purchased together. Therefore, recommendation system model should be comprehensive and should use all the recommendation techniques to utilize the customer related data or information efficiently.

The third objective of our research is to find the impact of a product on other products. The links or relationships created in the customer-product graph define the association between different entities, however the impact of one entity on another entity should incorporate the reliability or importance of that entity to the targeted entity. For example, a product that is frequently bought by many customers or has been frequently purchased with other products is a popular or trending product. Therefore, recommendation systems should incorporate the influence values of products and customers. The customer who purchase more frequently has more impact compared to the customers who buy occasionally. According to [18], customers with higher trust factor are more likely to predict reliable ratings. The authors in [18] used the indirect connections to find the trust factor of given customers and found the leaders in the customer network. Predictions using the leaders' opinion lead to better precision in the prediction system of rating. The process of ranking products in the graph network while considering the influence transfer from indirect connection is very complex task. Therefore, the recommendation graph based model should incorporate a ranking algorithm such as PageRank.

1.3 OUR CONTRIBUTION

Our approach is to integrate different types of data and recommendation techniques in the graph based recommendation systems to handle retail stores sparse datasets. In our approach, we use different types of available data: customers demographic, product taxonomy, and transactional. Additionally we integrate content based, collaborative filtering and association rule mining techniques in the graph based recommender system. PageRank algorithm is used to rank the products.

Since graph-based recommendation systems are new research fields, many interesting research directions related to it are yet to be explored. This dissertation studies three such research directions. Brief descriptions of these directions are presented below.

Integration of three types of data related to customer preferences: We used a retail store dataset, which have three types of information available. The dataset contains the demographic information of customers, product taxonomy and the purchase history of customers. In order to integrate the three types of information, the first step is to define the relationship between the customer to customer, product to product and product to customer. The customer-to-customer relationship depends on similarity between demographic properties and purchase patterns of customers. The similarity of customers based on demographic properties is calculated for new customers who do not have any previous purchase history. We also used the purchase history similarity for customers who have previously purchased history. The similarity between products is calculated based on product name which is the last level in product taxonomy. Additionally, we calculate the similarity between products based on product type, which is one level above product name in the product taxonomy. Product type groups product names without any consideration of brand names. The graph representation of customers and products includes all the relationships between customer to customer based upon demographic properties and purchase history, and the product-to-product relationships based on product similarity at the given product categorical level in product taxonomy and the association between products discovered using the association rule mining. Higher association between customers and products using the various types of data lead to better accuracy and precision in recommendation system for the sparse retail store dataset.

Integration of three recommendation techniques: A graph-based recommendation approach to integrate the content-based approach with the collaborative-filtering approach is explored by Huang et al. in [20]. Similar to the approach in [20], we integrate the content based, collaborative filtering and association rule mining techniques in our recommendation system. Content-based techniques explore product-to-product correlations, and collaborative filtering techniques discover the customer-to-customer correlations. Association rule mining generates the frequent set of products and leads to creation of the relation between products that are purchased frequently together. The integration of these three techniques establishes stronger relations in the customer-product graph network from the sparse dataset. Therefore, the stronger relationships between customers and products discover customer preferences more accurately and precisely. This comprehensive approach uses the three types of data available to distinguish the customer personal preferences. The graph based comprehensive approach also shows the flexibility of the model that can integrate different types of techniques and utilize various types of customer preferences related data/information from different sources.

PageRank, Transferring the influence from indirect connections: Larry Page invented the PageRank algorithm which traverses through millions of website and rank the websites based on their influence value. We used the PageRank algorithm to improve the representation of products and customers in our graph network in addition to ranking products for a target customer in the recommendation activity. First, we improve the representation of customers and products based upon incoming links to assign the accurate influence values to the node. Secondly, we traverse the graph in order to produce the ranked list of products for the targeted customer by extracting the sub network graph of the customer.

We evaluate our model using a Retail store dataset [38] and which is also known for its sparsity and cold start problem. However, our recommendation system shows better accuracy compared to either content based or collaborative filtering approach. Our recommendation system is evaluated using accuracy measures such as precision, recall and f1-score [40]. The improvements in precision, recall and F_score shows the benefits of using the graph based recommendation model to handle the sparse dataset. The recommendation model is also presented in the Figure 1.

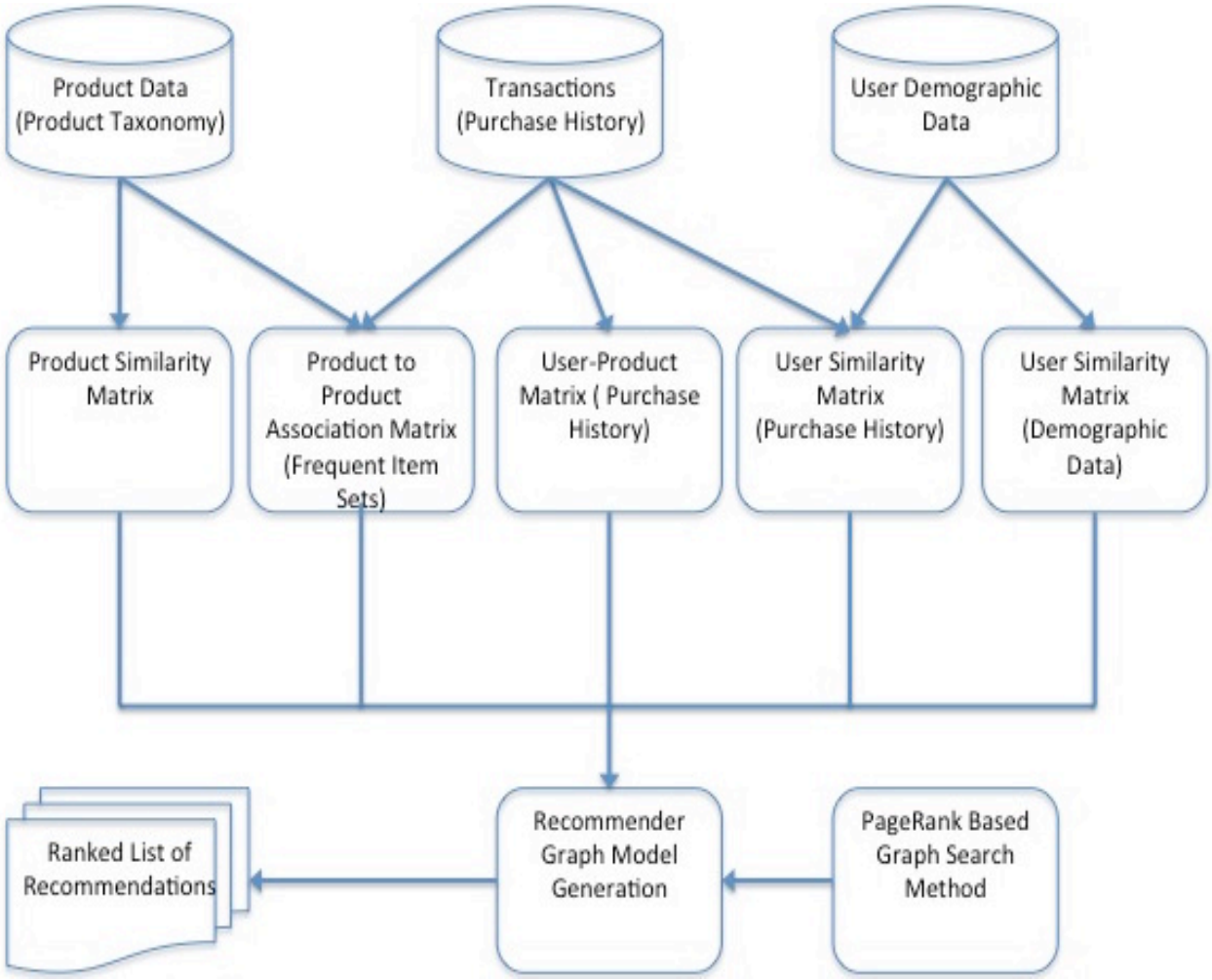

Figure 1-1 Purposed Recommender Model

Some interesting facts analyzed from a Retail store dataset and our recommendation system model:

1. The integration of techniques such as content-based, association rule mining and collaborative filtering produced better results in comparison of recommending products, which are previously bought by a customer.
2. The support and similarity impact results based on data characteristics such as higher support or similarity generates better results for dense data while lower support or similarity generates more personalized results for sparse data.
3. Quantity based similarity does not impact the accuracy of our recommendation system in comparison to similar products based similarity.
4. Integration of demographic properties of customers produces more accurate recommendations.
5. Usage of product types instead of product name produces more accurate results for our recommendation models.
6. Our ensemble approach generates more accurate recommendations compared to other models such as content-based, collaborative filtering or frequent product-set mining based models.
7. A ranked list of products allows a model to limit the total number of recommendations. Thus, the PageRank based ranking procedure handles the information overload problem in a recommendation system.

1.4 THESIS ORGANIZATION

The rest of the thesis is organized into 6 chapters. Chapter 2 gives the background information and includes the information of various data sources available in a retail store domain, and various recommendation techniques. Chapter 3 includes the comprehensive survey of recommendation systems, the hybrid techniques and the graph-based model. Chapter 3 also introduce the influence transfer algorithms used in recommendation systems. Chapter 4 provides the methodology used in our recommendation system. It includes the description of three main stages of our recommendation system: representing customers and products, creating the graph network of customer and products while assigning the appropriate weight to every link, and producing the ranked product list using the PageRank algorithm. Chapter 4 also presents the characteristics of Retail store dataset that lead to the sparsity and cold start problem. Chapter 5 presents the evaluation of our recommendation systems through comparing in addition to show the impact of integrating the techniques such as content based or collaborative filtering. Chapter 6 presents the conclusion and gives direction for future research.

Chapter Two: **Background**

The main goal of recommendation systems is to identify customer preferences and then using these preferences and certain criteria to predict future customer behaviour. Each customer has his/her own needs and opinions, which can be used to define recommendations known as personalized recommendations [1]. There are two main components of a recommendation system model: data or information available and the recommendation technique. A recommendation system should be able to make the best usage of data and information available to produce accurate recommendations. Similarly, a recommendation system should use a recommendation technique or a combination of recommendation techniques to enable the recognition of customer preferences from the available data or information. There are number of possible recommendation techniques. The integration of these recommendation techniques can be used to generate a hybrid solution. Hybrid solutions tend to use the available data or information more efficiently and hence perform better for a given problem domain and dataset. The main goal of this research is therefore to integrate several recommendation techniques using a graph based structural representation for a retail store recommendation system.

2.1 DATA TYPES

Recommendation systems are information processing systems continuously collecting data for example in the e-commerce domain. The data collected in the ecommerce domain can be categorized into three main subcategories: users, products, and transactions [2]. E-commerce applications are very diverse; hence data collected within these three main subcategories can be very diverse as well. The main purpose of the customer related data category is to collect customers' characteristics such as demographic data, browsing patterns, etc. Similarly, the product related data has product characteristics. For example books have authors, publishers,

price, genre, and many others characteristics that can be recorded. The transactions record records interactions between a customer and a system [2]. The transactional dataset not only defines the characteristics of customers and products, but also defines associations between different customers, products, and customer-products.

User Representation

Different recommendation systems have different customer information since the types of ecommerce datasets can be variable. The recommendation systems should be able to handle variable customer datasets because of the variety of customer information [2]. However, some recommendation system only collect ratings, while others have demographic information too such as age, gender, profession, income, location, etc. providing incomplete customer information. Moreover, a recommendation system may also collects the behaviour patterns of customers as well, such as navigational patterns.

To present the customer needs and opinion information, various customer profile techniques have been developed which are also known as customer profiling [4]. Since the main purpose of a recommendation system is to provide a personalized recommendation for the target customer, customer profiling is one of the major challenges, since the model of customers should present customer preferences efficiently [4]. The complexity of the customer model representation depends upon the recommendation domain under consideration. Therefore, the customer model should handle very complex data/information representations in order to be easily accessible and updatable model.

Product Representation

Products are products or services that are recommended to customers in a recommendation system. Different domains have different types of products and products within

same domain can be distinguished in terms of their properties. Low dimensional representation of products has few properties. However, high dimensional representation of products has many characteristics, such as a grocery product can have a list of the contents, price, promotion, brand, and many other properties. Many attributes can be derived from such information such as the variation in price.

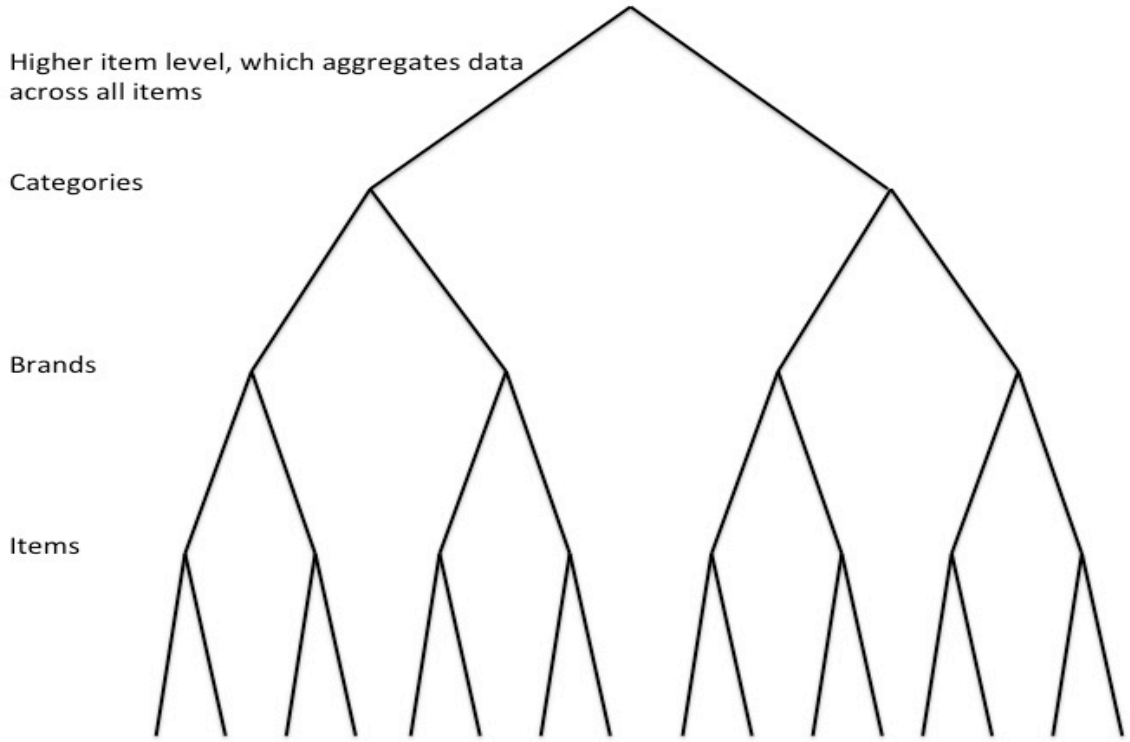

Figure 2-1 Product Taxonomy [2]

In a product dataset, the information of a product can have a complex format. For example, a product can have structural information, time information, context information, etc. Similarly, a product may have category information, description, and textual information if it is in a Retail store dataset. A cloth product has different properties from a food product. However,

products share some common properties such as price. The representation of a product should incorporate all properties and enable finding similarities between products.

Product taxonomies is an example of representing information on products. Product taxonomies can be derived for example by dividing products into sub categories based upon their price. A product taxonomy handles the sparsity problem in some situations since a category is a group of products. Moreover, the number of levels in product taxonomy can also be used to define similarly between two products.

Transactions

Transactions represent customer interactions in a system. A transaction can be a rating given to a product by a customer or a purchase order. Transactions can help deriving customer preferences. Transactions types can be diverse due to different ways of interactions between customers and systems. The representation of a transaction varies from a simple to a complex representation.

Transactions may have facilities for feedback. These feedbacks, such as ratings or comments can be stored. However, this feedback is not reliable since a customer usually does not spend time to rate or comment on a product. There are many ways of improving the process of getting feedback for ratings as well [9]. However, each method of obtains ratings has its own advantages and disadvantages [9].

Another type of transactions' dataset is a purchase history dataset that has all the purchase orders. Each purchase includes set of products and quantity of each product. Additionally, each purchase has other details such as time, date, price, total amount etc. A purchase history dataset is a more reliable dataset for extracting customers' preferences since customers are not required to spend any extra time to give their feedback. If a customer buys

same type of products frequently, it shows the customer's likability for a specific type of product.

2.2 RECOMMENDATION TECHNIQUES

Recommendation systems provide a list of useful products to a customer using an information filtering process. The filtering process attempts to extract useful information from the previously recorded information of customers. A recommendation technique generates a list of useful products for a customer to increase the utility of the customer's experience of selecting products from a set of products [27]. However, finding a recommendation technique not only depends on available techniques but also on the data sources in a given domain. The most commonly used recommendation techniques in recommendation systems are content-based and collaborative filtering. Most recommendation systems use hybrid techniques, which are combinations of content based, collaborative filtering or other recommendation techniques. Hybrid models have better performances or accuracy according to [10]. A brief explanation of content based, collaborative filtering and hybrid recommendation techniques is given below.

Content based

Content-based recommendation algorithm searches for products that are similar to the products purchased by a customer. Customer preferences are derived from the purchase history of a customer. For example, a customer's preferences profile for music purchases consists of the entire genres of the music liked by a customer. Therefore, a content-based recommendations algorithm needs proper representation of products' profiles and customer profiles. A content-based recommendation procedure can be completed in three steps [2].

CONTENT ANALYZER – In the real world, data is normally available in a raw form and needs be to pre-processed to be useful. The content analyzer pre-processes the raw data to extract

relevant information. In a retail store recommendation system, the main responsibility of a content analyzer is to extract products' profile and customers' profile from various data sources.

PROFILE LEARNER – In this step, the content-based recommendation algorithm represents the profile of products and customer in a proper style or format. A profile learner uses several machine learning techniques to derive customer preferences. For example, if the customer feedback is not given, a profile learner derives the preferences of a customer from the number of times when the customer purchases a product. Additionally, the profile learner uses a criterion to map the number of times a product is purchased into a rating. Therefore, profile learner has to have normalized data to represent customers' preferences. The profile learner should be easily updatable and able to incorporate the changing preferences of customers.

FILTERING COMPONENT – A filtering component matches a customer's profile representation (representation of customer interests and needs) to products' profiles to generate a similarity matrix based on the similarity of a product to the customer. The generated matrix of customer's preferences creates a ranked list of preferred products.

Collaborative filtering

Collaborative filtering approaches use similar customers' preferences to explore the preferences of a given customer. Unlike content-based method, the collaborative filtering methods exploits customer preferences through exploring products purchased or ranked by similar customers [9]. For example, if two customers have similar purchasing patterns, a recommendation system recommends products, which are not explored by the target customer but bought by similar customers [10]. The collaborative filtering algorithms are based on two kinds of algorithms to find similar customers: neighbourhood based and model based [2].

The neighbourhood based methods use a comparison of the properties of products and customers. Similar customers, who are essentially neighbours of a target customer, can be found using customer-based similarity [2]. In a customer-based similarity matrix, the neighbours of a customer are the customers that have similar preference. In a customer-based similarity matrix, the value of similarity between two customers can be derived from the number of products purchased by both customers.

Another type of collaborative filtering methods is a model based recommendation methods. In a model-based method, similarity of customers cannot be derived through comparing the customers' attributes, but training a predictive model that assigns a rating to a product. Model based techniques such as Bayesian probability, neural network, support vector machine and many other techniques use the latent properties or attributes of customers and products. For example, a model based technique can create clusters of similar customer, who like the same type of music without defining the attributes to group customers. Therefore, model-based techniques can find some interesting patterns in data, which are not discovered previously by other methods or stated already.

Although, the model-based techniques can discover new patterns, it is difficult to calculate reliability of those patterns [2]. However, neighbourhood methods are simpler and easier to justify. Moreover, the efficiency of neighbourhood methods is better than model based methods since neighbourhood methods do not have any model to train. Therefore, the correct representation of customer or product profiles in neighbourhood based recommendation system produces very accurate prediction in a timely and efficient way. Neighbourhood methods also are more stable since adding new customer or product in the model does not impact them and are

suitable for commercial applications, which have large datasets and with addition of new information on regular basis [2].

Hybrid Solution

Hybrid recommendation systems are combinations of many recommendation techniques. There are many ways of integrating these techniques [10]. An integration of different techniques tends to improve the customer-product interactions matrix thus generates recommendations that are more precise and ore accurate. A hybrid model of a recommendation system should be able to integrate different types of data or information available related to customers' preferences. As stated in the previous section, there are various types of data sources that create different types of relations between customers and products. For example, a relationship between two customers can be derived from demographic similarity or purchasing similarity. Similarly, a relation between customer and product can be from purchasing patterns or navigational patterns. Therefore, recommendation system should able to model all entities and relations between entities.

Another property of recommendation models is comprehensiveness. If a recommendation model integrates several recommendation techniques, it will lead to a more comprehensive approach, which will handle a variety of data and information. Content-based approaches utilize the product-to-product related information efficiently to identify similar products and collaborative filtering approaches utilize the customer behaviour patterns such as purchasing or navigation patterns to find similar customers. Therefore, one certain technique may handle a particular type of data more efficiently than another technique. A combination of many techniques leads to an efficient recommendation solution that maximizes the utilization of available data.

Another important property of the recommendation models is the procedure for creating a ranked list of products. A ranking procedure for comprehensive recommendation models becomes an important task since recommendations are generated using different types of recommendation techniques. The right combination of multiple recommendations techniques may produce better recommendations. However, a ranking process also becomes very complex activity if the underlying structure of the preferences' model of a customer is not structural and efficient to use. A graph based hybrid solution overcomes many of these problems. A detailed explanation of a graph based solution provided in the methodology chapter.

2.3 ASSOCIATION RULE MINING

Association rule mining has been used in recommendation systems for marketing to increase revenue as well as customer satisfaction [23]. Association rule mining is an important techniques used to discover interesting patterns in data. Association rule mining is used for cross-selling or promotional techniques in recommendation systems. Finding frequent sets in a transactional dataset is known as market basket analysis. Agarwal introduced market basket analysis in 1993 to increase sales of products through cross selling [23]. Association rule mining may discover rules or patterns between products that are not apparent initially.

Mining association rules from a large business database, such as a transactional dataset, has been an important topic in the area of data mining. This topic is motivated mainly due to the application to market basket analysis to find relationships between products purchased by customers, that is, what kinds of products tend to be purchased together [23]. Such information is useful in many aspects of market management, such as store layout planning, target marketing, and understanding customer behaviour. Traditional association rules mining (ARM) techniques depend on a support confidence framework in which all products are given same importance by

considering the presence of a product within a transaction. The goal of such techniques is to extract all the frequent product-sets, then generate all the valid association rules A → B from frequent product-set A and B whose confidence has at least equal to threshold value. In other words, given a subset of products in a product set, we need to be able to predict the probability of the purchase of the remaining products in a transactional database.

An association rule is an expression of the form A → B, where A and B are sets of products. Such a rule reveals that transactions in the database containing products in A tend to contain products in B. The probability of a transactions containing A also containing B, is called the confidence of the rule. The support of the rule is the fraction of the transactions that contain all products in both A and B. In other words, the support is the frequency of the given dataset and confidence is the occurrence of product in transaction dataset when the other product of frequent product set also appears in the same transaction.

For example, an association rule Bread → Jam (sup = 30%, conf = 60%), says that 30% (support) of customers purchase both bread and jam together, and 60% (confidence) of customers who purchase bread also purchase jam.

For an association rule to hold for a specific case, the support and the confidence of the rule should satisfy a customer-specified minimum support called minsup and minimum confidence called minconf, respectively. The problem of mining association rules is to discover all association rules that satisfy minsup and minconf. This task is usually decomposed into two steps:

1. Frequent product-set generation: generate all product-sets that exceed the minsup.
2. Rule construction: construct all association rules that satisfy minconf from the frequent product-sets in Step 1.

However, the frequent set mining is a very expensive operation and also requires a large memory since the operation is required to go through whole dataset in order to find the frequent product sets which have support and confidence above certain level [24]. Intuitively, to discover frequent product-sets, each transaction has to be inspected to generate the supports of all combinations of products, which, however, will suffer for lots of I/O operations as well as computations. Therefore, most early work was focused on deriving efficient algorithms for finding frequent product-sets [24]. The well-known Apriori algorithm as explained in [24] relies on the observation that a product-set can be frequent if and only if all of its subsets are frequent and thus a level–wise inspection proceeding from frequent 1-product-sets to the maximal frequent product-set can avoid large numbers of I/O accesses.

Additionally, a personalized recommendation system can limit the coverage of data to increase the performance of association rule mining algorithm [10]. Despite the great achievement in improving the efficiency of mining algorithms, the existing association rule models used in all of these studies incur some problems for the retail store datasets. First, it is more useful to find association between categories of product in comparison to find associations at the primitive concept level [39]. Secondly, the frequencies of products are not uniform. Some products occur more frequently in transactions while others rarely appear which prevents the discovery of general trends among categories. Therefore, we used the multi-level association rule mining as described in [39] in this thesis.

2.4 SPARSITY AND COLD START PROBLEMS

Collaborative recommendation system faced the problem of sparsity due to unavailable data or information [3]. The content-based approach can produce the recommendations for a sparse dataset but recommendations are not personalized enough according to customer

preferences. In the retail store domain, the sparsity problem leads to weaker connections within customers and products. The details of how sparsity problem lead to non-personalized recommendations is given below.

In a retail store dataset, the customer information represents single or multiple types of information such as demographic information, customer interest patterns, customers behaviour patterns and many others. All the available data or information of customer must shows correlation to customer preferences that is what products customer will buy. Therefore, a similarity measure that finds customers with similar preferences need to get established. A similarity measure that uses the demographic data of customers compares the demographic properties of customers and assigns a similarity value. The customer demographic attributes that show the stronger relations to customers' preferences should be selected to find similar customers. For example, a set of customers who belongs to same locations have similar preferences and another set of customers who belongs to same age group but from different locations shows similar preferences. Therefore, a similarity measurement should have an efficient grouping criterion that uses all the attributes properly and also able to handle the scalability issues. After the selection of attributes, the procedure of calculating a similarity value also becomes complex since attributes are available in different types such as numeric, text etc. Additionally, the selection of important sources of information also affects the weight of each attribute in a similarity measurement function since the purchasing patterns' similarity could produce more accurate recommendation compared to a similarity based on demographic data. Therefore, recommendation systems have to deal with customer data that have many attributes and comes from different sources.

If two customers are buying the same type of products, it is another way to establish a similarity between customers. However, defining product similarity measurements faces same problems that are faced by the customers' similarity measurements. Products' similarities also face the problem of finding a right set of attributes and integrating different types of attributes. Products have many attributes such as price, description, category, content, etc. Extracting the correct attributes that distinguish products properly is a complex procedure. Since the attributes of a product have different types of presentations or data types, the task of establishing a similarity measurement becomes difficult. Another factor plays an important role is the product taxonomy in a retail store domain. It is unlikely a customer buys same product when there are many other product belongs to same category. However, the customer can show consistent patterns of buying products from a same category of products. Therefore, similarity between customers can be derived from the similarity of purchasing products from same category.

Another problem that arises due to sparsity in transactional dataset is that it leads to lower number of associations between products because customers do not buy the same set of products. Similar to the problem of finding similarity between two customers based who buys the same products from a large dataset of products, the association between two products rarely exist since customers buys different products. However, the product taxonomy can also play a better role to find the association between products. Customers can buy products from a frequent set of categories. For example, a customer who buys bread, egg and milk frequently can buy products of different brands.

The second most common problem faced by recommendation system is the cold-start problem. The cold-start problem in collaborative filtering models arises due to new customers who do not have pervious data or information to establish their preferences. However, if the

demographic data of customers explains the preferences of customers, it could remove the problem of cold-start.

2.5 GRAPH BASED SOLUTIONS

Graph based solutions are hybrid solutions for recommendation systems. As stated by Easley and Kleinberg in [41],"A graph is a way of specifying relationships among a collection of products. A graph consists of a set of objects, called nodes, with a certain pairs of these objects connected by links called edges." Two nodes are neighbors if they are connected by an edge in a graph [41]. Graphs are useful because they can serve as mathematical models of network structures [41]. According to [22], patterns in data can be modeled as a network and represent the information through vertices and edges. Vertices can be entities such as people, movie. Edges can be relations between entities such as an act of viewing a movie by a person [22]. A structural way to combine and present the information in a graph based solution leads to better customer preferences modeling. The ability to represent many types of customers' preferences information in a graph based recommender system model handles the sparsity problem more efficiently in comparison to other models [3]. Another advantage of using a graph-based representation is to use the graph based search criteria to rank the recommendations for a customer. There are many developed graph-searching methods that efficiently traverse a graph. Therefore three main components of graph based recommendation systems are:

1. Nodes or vertices: represent customers and products.

2. Edges: Those are the links between nodes and represent relations between different entities.

3. Graph search method: traverse a graph to produce a ranked list of recommendations.

There are several advantages of using graph based recommendation techniques. The nodes are able to represent different types of entities and edges are able to create many types of relationships between entities. Additionally, the graph searching methods derive recommendations using certain criteria. The independency of each component of a graph based recommender model produces a flexible solution that allows the replacement of one component without affecting other components. Therefore, there are many advantages of using graph-based methods for recommendation systems.

An advantage of nodes in a graph-based recommender model is the flexibility in handling various types of data or data sources [21]. The graph-based model allows the usage many types of entities. However, only the required entities should be represented in a graph because a larger number of different entities lead to a complex graph structure. Similar to nodes, edges can represent various types of relations between entities. The relations between entities can be derived using various recommendation techniques. The graph representation allows us to represent each relationship using different edge. Additionally, a model can give more importance to certain type of recommendation techniques through assigning the different proportion of weight to different types of edges. Therefore, nodes and edges can represent very complex data into a structural way.

Another benefit of using a graph-based model is the influence transfer from indirect connections [19]. For example, a product purchased by many customers should be more reliable choice compared to other products for a target customer even if the target customer purchased the popular product as frequently as other products. Therefore, each node should have the trust or reliable value as explained in [18]. Producing recommendations based on more reliable entities

increases the accuracy of predictions. PageRank is an example of an algorithm that assign an influence value to each node depending on the number of relations and weight of each relation.

Moreover, graph-based presentations can also use graph search methods. Graph searches methods can be modified to implement certain criteria when traversing a graph. PageRank can be used to traverse a graph to rank recommendations for a customer. [30] The PageRank algorithm produces a list of websites based on the importance of a website relative to the search topic. Similarly, the PageRank algorithm can be useful to identify recommendations for a customer based on the importance of entities and the relativity to the customer [37].

According to Easley and Kleinberg in [41],“ we view of PageRank dynamically as a kind of fluid that circulates through the network, passing from node to node across edges, and pooling at nodes that are the most important.” Easley and Kleinberg stated three steps to compute PageRank.

1. In a network with *n* nodes, they assign all nodes the same initial PageRank, set to be *1/n*.
2. They choose a number of steps *k*.
3. Then they perform a sequence of *k* updates to the PageRank values, using the following rule for each update:
    a. Each page divides its current PageRank equally across its out-going links, and passes these equal shares to the pages it points to. (If a page has no out-going links, it passes all its current PageRank to itself.) Each page updates its new PageRank to be the sum of the shares it receives.

Because PageRank is conserved throughout the computation, we can limit the process with a simple interpretation. In a limited process of PageRank, the limiting PageRank values regenerate themselves exactly or closer to the values when they are updated [41].

The PageRank algorithm is an iterative process that keep calculating the influence value of nodes until they become stable or there is no significant change in the values. As described in [36], PageRank starts by equally distribute the probability to all of the nodes and then recursively recalculating influence values. While recalculating the influence value of a node, PageRank distribute the influence value of a node to other nodes that have incoming connection from this node. In other words, a node gets new influence value depending on the number of incoming links. For example, if we have four node A, B, C and D. Each node gets .25 initial values. Suppose, B has links to A and C, C has link to A, and D has links to all. Afterwards, the PageRank of A is calculated as PR(A) = PR(B)/2 + PR(C)/1 + PR(D)/3 = .46. Since all nodes get new influence values, PageRank keep updating nodes' influence values until there is no significant change in the values.

Chapter Three: **Related Work**

Recommendation systems gained a significant spot in the research area since last few decades [1]. Recommender system is a problem-rich research area since there is an abundance of practical application. Recommender system helps customers deal with information overload and provide personalized recommendations. The four ways to create customization presented in Joe Pine book [11], lead to the problem of information overload since companies are able to produce many products to meet multiple needs of customers. However, recommender systems are able to achieve mass customization though providing recommendation based on customer personal preferences as stated in the taxonomy of recommendation systems in [12].

The customer's personal preferences are available in various formats. The extraction of customer's personal preferences is the most important task in the recommendation process [1]. Wei et al. present four types of input data sources of customer's preferences in e-commerce domain: 1. Demographic information of a customer, 2.Rating data, 3.Behaviour patterns, and 4. Transactional datasets. Many recommender systems use a number of data sources to produce personalized recommendations [1]. The diversity in sources of data makes the extraction of customer preferences activity a difficult task in recommender systems. We used the Retail store dataset, which contains three types of data: 1. Demographic information of a customer, 2. Categorical information of products, and Transactional dataset, it has the purchase history of customers.

As explained in the background chapter, a recommendation system model should be comprehensive and should use all the recommendation techniques to utilize the customer related data or information efficiently. Therefore, another important factor is a recommendation method

in recommender systems. Adomavicious in [10] classify recommendation methods into three categories: 1. Content based, 2. Collaborative filtering, and 3. Hybrid methods. We used the graph-based hybrid approach. Our comprehensive approach combines the content based, collaborative filtering approaches and association rule mining.

In this chapter, we provided the current state-of-art of recommendation systems. In the next section, we discussed the existing hybrid recommendation systems that handle various types of problems. Then, we discussed the existing graph-based recommendation system to show the flexibility and comprehensiveness of graph-based recommendation models. Furthermore, we discussed the importance of influence transfer from indirect connection in a graph-based recommendation model. Additionally, we discussed the two major problems in recommendation systems: sparsity and cold-start. We provided the existing approaches to handle sparsity and cold-start problems. At last, we discussed our approach.

3.1 RECOMMENDATION TECHNIQUES

One of the recommendation techniques is content-based. A recommender system using the content-based approach recommends products based on the product-to-product correlation matrix [20]. The product-to-product correlation based approaches recommend products similar to those a given customer purchased in the past. Product-to-product correlation based recommender systems use similarity between products to predict recommendations similar to the previously indicated preferences of a customer. As stared in [12], Reel.com's Movie Matches, Moviefinder's Match Maker, CDNOW's Album Advisor and Amazon.com's Customers "Who Bought", used content-based techniques to find products, which complements the past experience of a customer. A recommendation system not only depends on the past behaviour of a customer but also on the current preferences of the customer as well. For example, Reel.com's

Movie Matches, Moviefinder's Match Maker, and CDNOW's Album Advisor use the information of a product currently looked by the customer. Therefore, recommendation systems can make use of products in the shopping basket too. Moreover, a customer manually enters his/her interest to find products that meet or come closer to the indicated criteria, such as the CDNOW's Album Advisor gives the albums, which have the artists, indicated by the customer.

Collaborative recommender systems use the people to people correlations [12]. The people-to-people or customer-to-customer correlation methods generate recommendations based associations between customers. Associations between customers derived from the similarities of purchasing habits or demographic properties. For example, adults with kids buy baby products while other customers do not need baby products. Therefore, the demographic information can create groups of customers with similar behaviour patterns. As stated in [12], Amazon.com's Book Matcher, CDNOW's MyCDNOW, Moviefinder's We Predict, and Levis's Style Finder find similar customers similar. Therefore, the content coverage of the collaborative filtering techniques is better than the content based techniques since a customer can discover new products which can't be discovered using the past experience of a customer.

In the collaborative filtering techniques, similarity of customers can derived from ratings such as CDNOW's MyCDNOW, Moviefinder's We Predict, and Levis's Style Finder. On the other side, Amazon.com's Book Matcher uses the purchase frequency to derive the rating of a given product. Therefore, the rating patterns can derive from the buying or click-stream behaviour patterns of customers as well. The derived rating patterns are more reliable since these are usually automated and customers do not need to make any extra effort to record their preferences. Various recommender systems used in real world are based on product-to-product or people-to-people correlations given in [12] are presented in Table 1.

Table 3-1 Techniques Used in Recommender Systems [12]

Recommendation Technique	*Technique*
Amazon "Customer who bought like"	Item-to-item Correlation
Amazon "Book Matcher"	People-to-people Correlation
CDNOW "Album Advisor"	Item-to-item Correlation
CDNOW "My CDNOW"	People-to-people Correlation
Levis "Style Finder"	People-to-people Correlation
Moviefinder "Match Maker"	Item-to-item Correlation
Moviefinder "We Predict"	People-to-people Correlation
Reel "Movie Matches	Item-to-item Correlation

As stated above, content based and collaborative filtering techniques utilize different types of data to extract customer preferences, the integration of these two techniques leads to hybrid approaches which can utilizes data or information in an efficient way [5]. Content based and collaborative filtering techniques have many limitations such as limited coverage, sparsity, new customer etc. The limitations of content-based and collaborative filtering approaches can be removed through integrating both of these techniques into one model [10].

According to Adomavicious, different ways of integrating the content based and collaborative filtering recommendation techniques can be classified into four categories:

1. We can implement content based and collaborative filtering methods separately. Then, we can combine recommendations produced by both techniques in the end [25]. For example, an ensemble approach of combining many recommendation techniques gives the advantage of choosing a technique, which gives more accurate recommendations [25]. Recommender system used in [25], selects a recommendation technique based on recommendations, which are more consistent with the past ratings of a customer.
2. Another way is to integrate the content-based technique into collaborative technique, which handles sparse data as mentioned in [5, 7].

3. We can incorporate some collaborative characteristics into content-based recommender model [6]. Combining the collaborative filtering information into content-based system increase the coverage of content-based methods [6]. Additionally, the scalability problem of a content-based method can be removed using the collaborative filtering characteristics as Soboroff and Nicholas used for the dimensionality reduction in [6].
4. The fourth way is to incorporate both content based and collaborative methods using a general unifying model.

Additionally, many collaborative recommendation methods used a Retail store dataset and have been improved to handle the sparsity and limited coverage problem. The domain knowledge integration leads to better accuracy in recommendation systems [14]. The domain knowledge can derive new information using some criteria suitable for the targeted domain problem such as sequential patterns discovering [17]. Deriving new information from available information improves profile of customers and allows a recommendation system to implement marketing strategies, such as product bundling [15], product profitability [16]. We used the combination of content based, collaborative filtering and association rule mining techniques in our hybrid model. A brief survey of some hybrid recommendation techniques is given below.

Kim et al in [8] combines the content and collaborative technique to maximize their respective strengths and overcome their drawbacks. The proposed approach creates the clusters of customers to get a group-rating matrix. The group-rating matrix is used to overcome the problem of limited rating. If the rating for a product from a customer is not available, the system uses the group rating. This solution efficiently handles the sparsity and cold start problem. Moreover, the weight to the group rating keep decreasing while the system collects more customer ratings. The framework of the proposed solution is shown in Figure 1.

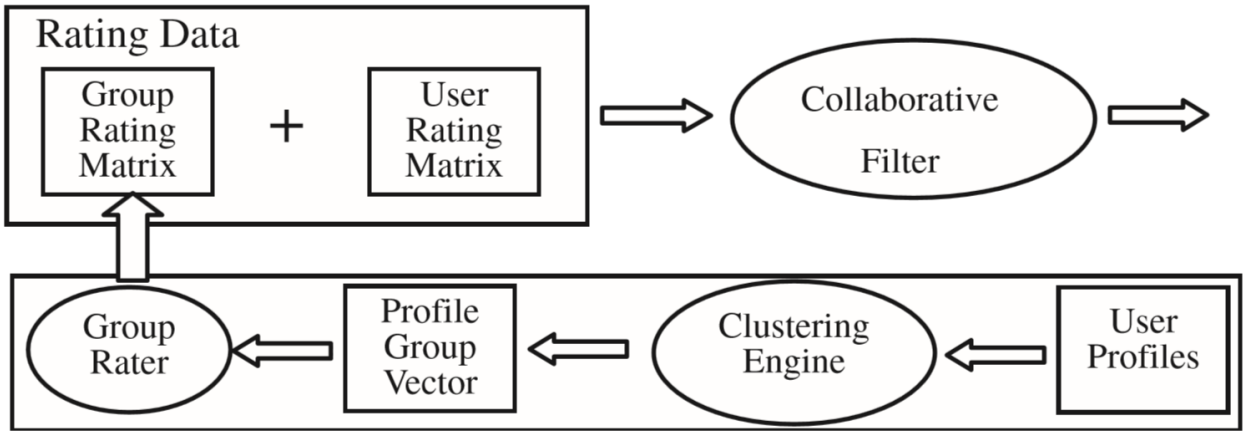

Figure 3-1 Recommendation Model proposed in [8]

Zhang and Shi in [14] have developed a recommendation system with higher accuracy through integrating the domain knowledge. In a retail store dataset, the transactional dataset expresses behaviour patterns of customers while having the details of products. Zhang and Shi [14] improved a collaborative filtering algorithm through integrating the customer profiles based on domain knowledge as well as the taxonomy of product dataset. Similarity of products is based on the deepest level of product taxonomy at which products belongs to a same category. The product ontology used in this paper is shown in Figure 2. Additionally, they explore new products by giving more importance to products that are not explored by the target customer.

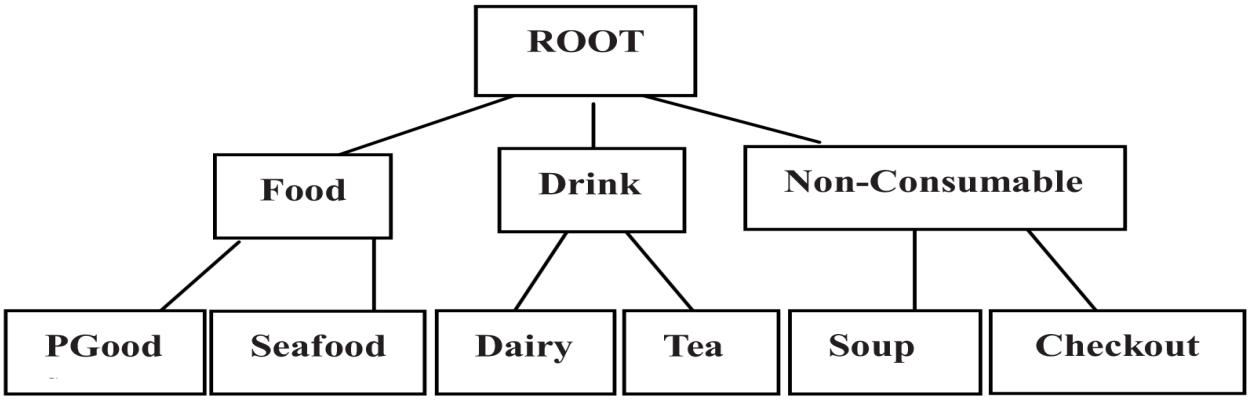

Figure 3-2 Product Taxonomy used in [14]

Guo-rong and Xi-Zheng in [15] proposed an improved collaborative filtering based recommendation system for product bundling for marketing strategies such as promotions. The proposed recommendation model has three main components: creation of customer groups using the adaptive resonance theory (ART), finding association rules, and bundling. Guo-rong and Xi-Zheng used the ART to use high dimensional data to produce similar customers' clusters. Collaborative filtering method uses the clusters of similar customers to produce recommendations. The association rule mining method uses only those transactions of similar customers. Thus, it makes the association mining method faster. Products are classified into hot, general, and dull sale through analyzing their selling frequency. The next step is to identify the class of a product and it use the products classes preferred by a customer in past. The recommendation system produces a list of ranked products.

Huang and Huang [17] discovered the sequential patterns in a transactional dataset to produce an improved two-stage collaborative filtering recommendation system. Two customers are similar if they buy same products or sets of products during a certain time periods. For example, if a customer C1 buys set A during time period t1 and set B during time period t2, another customer C2 is similar to C1 if customer C2 buys same sets of products during same time period. Instead of considering products, the product category is used to find similar customers. The consideration of product category handles sparsity problem as well as makes customers' profiles more accurate and complete. GA based clustering is used to handle the problem of high dimensional data and avoid the local optimal problem. After generating the clusters of customers, the sequential patterns in each cluster are identified. At the time of the recommendation, a customer gets assigned to a cluster, which eventually identified the sequences

of products purchased by customers of the assigned cluster. Top-M category list of products is produced and top-N product list is produced through identifying the frequent selling products.

Chen et al [16] improve the collaborative filtering methods through integrating the product probability without affecting the efficiency or accuracy of the recommendation system. These types of improvement generate more revenue through cross selling products. They compare personalized recommendation systems to non-personalized recommendation systems. The accuracy of predictions is lower in non-personalized recommendation systems, which are usually the content-based recommendation systems. However the accuracy of personalized recommendation systems does not get affected when the product profitability is integrated.

3.2 GRAPH BASED RECOMMENDATION SYSTEMS

As shown in the previous section, recommendation techniques can be improved using some additional methods. However, the researches only make some improvements on a specific problem and purposed solutions still have limitations [20]. Therefore, if a hybrid recommender system can be created to incorporate many recommendation techniques for a retail store data, it can eliminate many limitations. One of the ways to make a recommendation system efficient is to define the information of customers' preference into a structural way. Many graph techniques have been discovered to store information in a structural way.

According to [22], there are three important scenarios where we can improve recommendation systems in order to provide better recommendations.

1. The first scenario is bringing the people together. To bring people together, a well-defined structure is required to establish all types of relationships between people.
2. The second case is emphasizing on modelling the relations between people and artefacts. The discovery of correct attributes and the criteria to establish relations between the

people, artefacts and people to artefacts will lead to a solution which has better explanation and believable recommendation techniques.

3. The third significant scenario is learning customer preferences in a timely manner. The recommendation system should able to quickly learn customer preferences. Therefore, recommendation system needs a structural representation of information in order to store the extracted customers' preferences on regular basis.

To achieve above improvements, graph techniques are very popular to establish the structural representation of information and there are many graph search techniques to generate recommendations.

The authors connect people to artefacts through jumps in [22]. The investigation of the implicit graph structure underlying in a recommendation system explains the relations produced by recommendation techniques. The criteria used in recommendation algorithms can be described through the connections in bipartite graphs. The framework proposed by the author distinguishes the algorithms used in recommendation systems through exploring the connections within entities in a social network. The random graph models are used to represent the properties of recommender graphs and social graphs. Hammock jump width is one of the properties of a recommendation technique and a connection in the social network graph. The random bipartite graph model chosen as the original model from which a social graph and a recommender graph model are derived using the skip jump as shown in Figure 3. Therefore, the cluster of similar customers is derived from calibrating the hammock width. Additionally, the connectivity between people and artefacts can be visualized within a social network graph. Finally, the calibration of minimum number of rating can be generated through exploring the rating patterns.

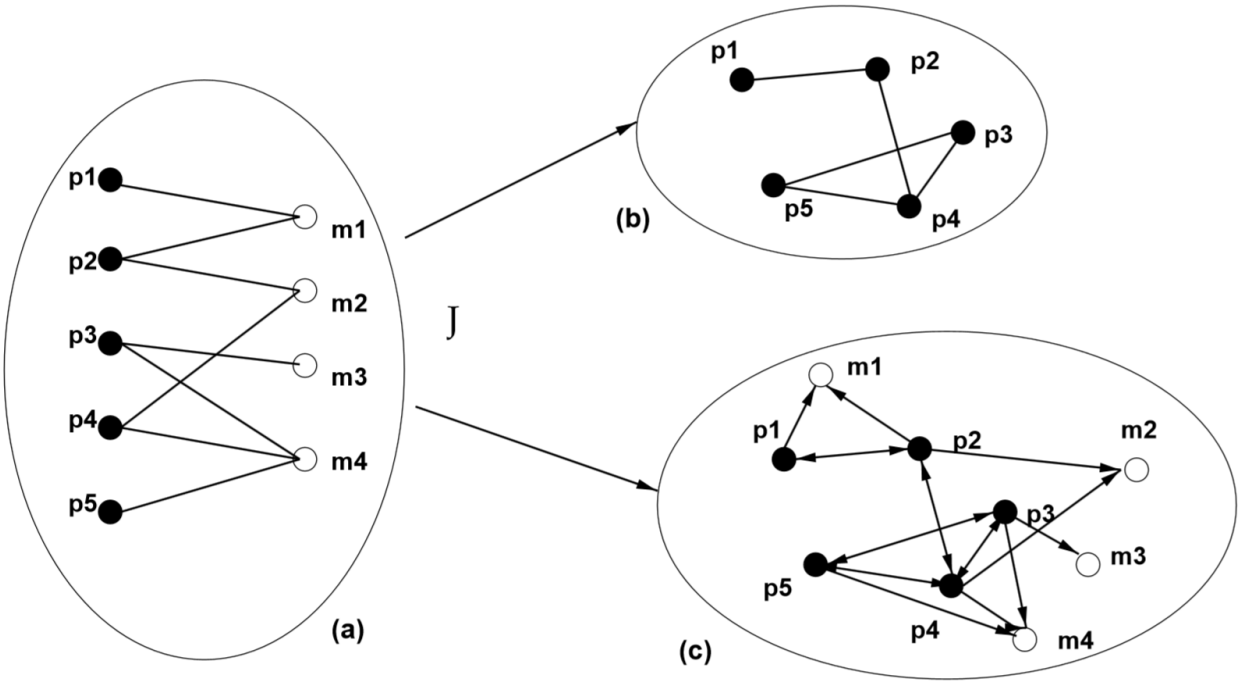

Figure 3-3 Social Model and Recommender Graph Model derived in [22]

Sawant used a weighted bipartite graph projection based collaborative filtering recommendation system in [26]. The weighted bipartite graph created a network of customers and businesses for a Yelp dataset. The Bipartite graph creation algorithm is a network based resource allocation process to produce a similarity measurement between users and businesses. The rating prediction of a business becomes the graph traverse activity in a weighted graph of users and businesses. Since the user-business graph can produce the rating for every business, the recommender system can generate a list of businesses preferred by a user. Moreover, Sawant handles the problem of sparsity using clustering since the rating of each business is not available in the Yelp dataset. Sawant clusters similar businesses using the k-means clustering and created s bipartite graph of users and clusters extracted from k-means clustering. However, a bipartite

graph based recommender only uses the immediate neighbours for generating recommendations. Similar to this technique, we cluster products into categories to handle sparsity problem.

In the study conducted in [21], authors handle the rich social media information. However, the major challenge in the rich social media is integrating the different types of information. Two challenges handled in this paper are:

1. Developing a framework, which incorporate different types of objects and different types of relationship between objects.
2. Another challenge is modelling the relationships between more than two objects, which are more complicated than a pairwise relationship.

To handle the above challenges, the authors used a hyper graph. An edge in a hyper graph represents relations between more than two objects. The acoustic signals similarities in the music tracks are represented through one of the relations between the music tracks. The hyper graph model shows relations more clearly than an ordinary graph as given in Figure 4. Whereas the influence of immediate neighbours is used in this study, but the influence of distant neighbours is still not considered in this study.

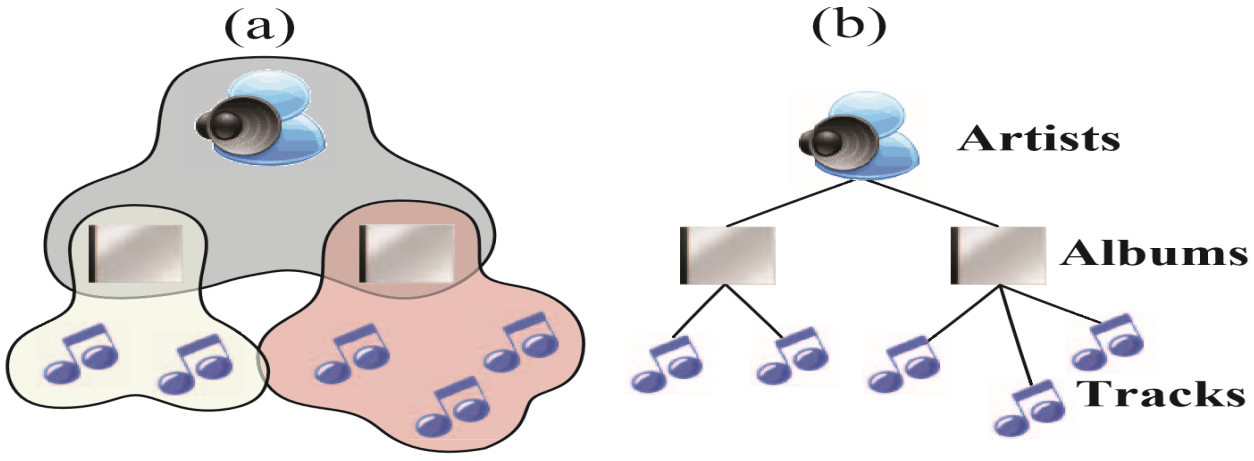

Figure 3-4 Hyper Graph used in [21]

The authors investigate the integration of content based and collaborative recommendation technique using a graph-based recommendation model in [20]. A Two layered graph is used to incorporate the book-to-book, book to customer and customer-to-customer correlations. They used the books' information, demographic data about customers, and orders (transactional dataset). A two layered graph as shown in the Figure 5 is consist of a customer layer which show the correlation within customers and a book layer which show the correlation between books. The links between these two layers are representing the purchase history. The graph search method produces the recommendations. Furthermore, this study also compares the low degree associations and high degree associations. The low degree association based predictions are produced using the customer purchase history and similar customers. A Hopfield network spreading algorithm is used for the graph search procedure to produce the high degree association predictions. A Hopfield network algorithm works efficiently in the concept information retrieval from different sources for the target node while providing the sufficient network coverage. However, this study concludes that the high associations do not impact the accuracy of predictions at a significant level. The reason behind no significant improvement using the high degree association can be dense data set. However, the high degree associations might work efficiently in sparse dataset since the high degree associations establish the connections between nodes, which were not connected using low degree association.

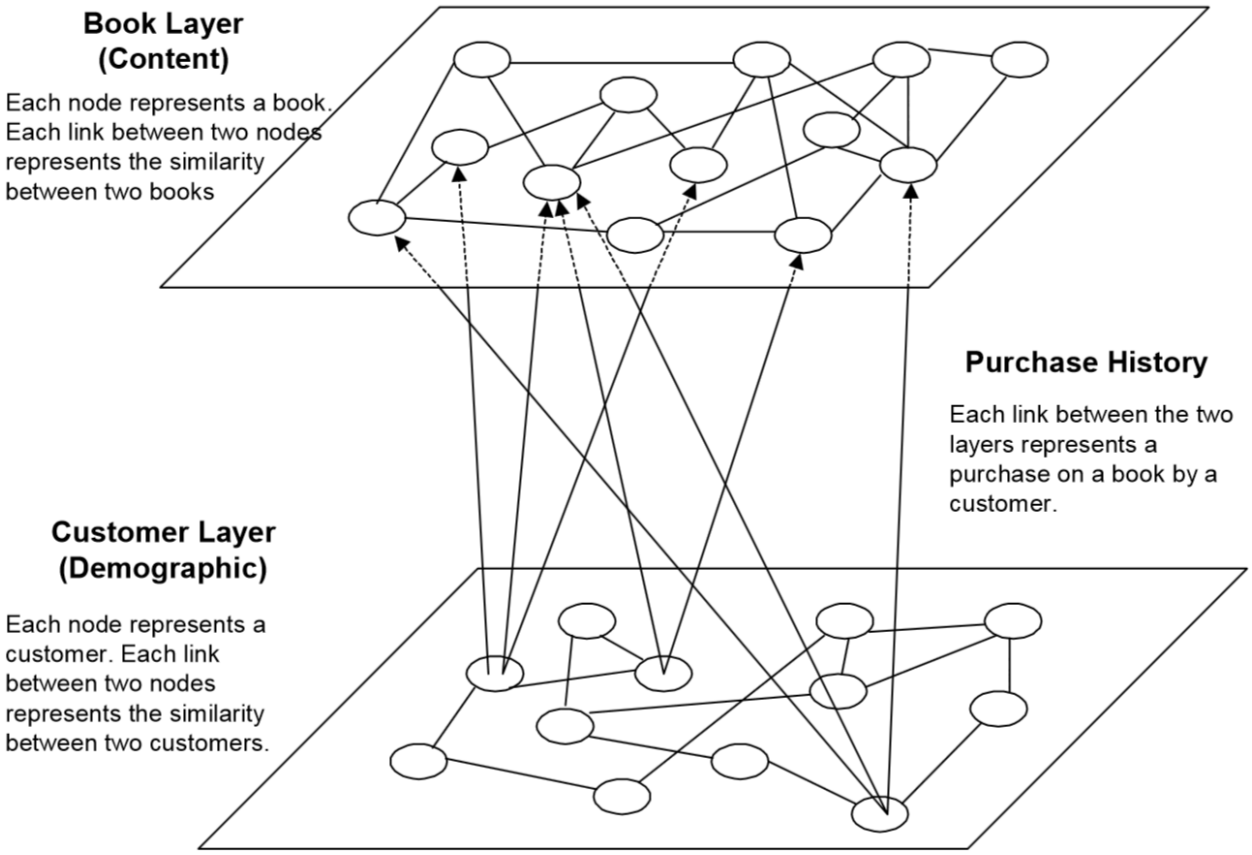

Figure 3-5 Two-Layered Graph Based Recommender Model [20]

3.3 INFLUENCE TRANSFER FROM INDIRECT CONNECTIONS

The studies also have been exploring the influence from indirect connections in a graph based recommendation models to find the impact of influences from indirect connections on the accuracy of predictions.

Follow the leader [18]: In this study, the authors explore the influential customers based upon their credibility. The credibility of a customer within a particular context depends on the expertise level of the customer in the given context as well as the trust gained from other customers. The investigation of customers leads to the discovery of the most influential customers within a particular context who have high credibility scores. The credibility of

customers depends on the trust rating given to a customer by other customers directly or indirectly. For example, the indirect influence is calculated as 1/5 * (0.6 * (1 + 0.8) + 0.8 * (0.4 + 0.6 + 0.6)) = 0.472 in Figure 6. Additionally, the credibility of a customer also depends on the customer's rating precision such as deviations of ratings from the average rating. Therefore, the trust factor of a customer can be derived from the customer's past behaviour. However, this study only used the leaders to predict the rating of a product. This technique used limited number of similar customers. Therefore, this technique ignores customers with less credibility. However, customers with less credibility can be useful.

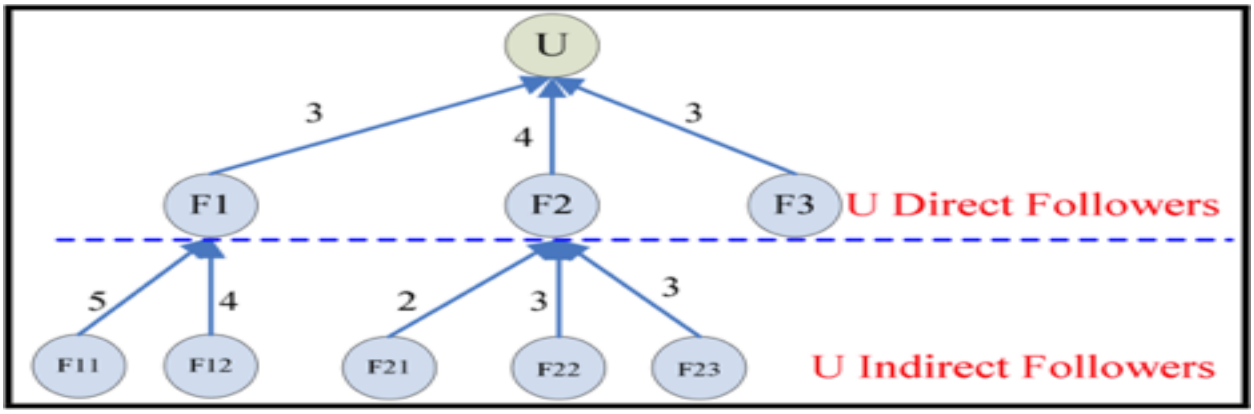

Figure 3-6 Indirect Influence Transfer [18]

SNRS (Social network based recommendation system)[19]: The authors investigate the tendency of friends choosing a same product or giving same rating to a product. Additionally, the connections to friends also help to explore the distinct friends' connection as well. The content coverage of recommendation system increases through exploring the options used by friends and distant friends. The increase in the content coverage handles the sparsity and cold start problems. The influence of a friend or distant friend should only be taken into consideration if the given friend has enough knowledge in such area. However, the social information of customers is not

readily available in e-market. Therefore, establishing the connections between customers and products should be derived from easily accessible data sources such as transactional dataset.

Many researches have been focused on using the graph techniques in recommendation systems, but the impact of influence transfer techniques such as HITS, page rank etc. is still not explored yet. Zhang et al used the topical PageRank algorithm to produce a ranked list of recommendations based on the product correlation graph [30]. The correlation graph has the two properties: propagation and attenuation. These two properties are required by page-rank algorithm. Similarly, Wang et al discovered the impact of a customer on a social group in order to better reflect the aggregate impact of the whole group using the PageRank algorithm [37].

3.4 SPARSITY IN RECOMMENDATION SYSTEMS

One of the most common problems in collaborative filtering models is the unavailability of data to extract preferences of customers [10,13]. To find if two customers are similar, the recommendation system compares the ratings of products. Both customers should rank the same (common) products. In real world, the product data set consists of a large amount of different products. Customers do not rate many products. Therefore, the size of a set of common products is very small or empty. Consequently, recommendation systems find very few similar customers and produce very limited number of recommendations.

Many researches have been attempted to eliminate the sparsity problem. The solution proposed in [28] is a product based collaborative filtering approach. The proposed solution handles both the sparsity and scalability problem. Another approach of handling the sparsity problem is dimensionality reduction through generating a dense customer-product interaction matrix. However the dimensionality reduction increases performance of some recommendation systems, but perform poorly in others due to loss of potentially useful information [29]. The

other way of improving a customer-product interaction matrix is association retrieval through considering both direct and indirect paths between customers and products [34]. Researches have also combined the content based and collaborative approach to alleviate the sparsity problem [8,20].

Another category of attempted approaches to handle the sparsity problem is the bipartite graph based recommendation systems [22]. These approaches develop the similarities between customers or products using the graph based techniques. For example, similarity between two customers can be the average commute path between customers [31]. Another type of this similarity is minimal hop distance, spread activation of the nodes, hammock jump [22] etc. The main drawback of these approaches is that there is no better interpretation of a similarity measurement in context of the prediction problem [20].

3.5 COLD START PROBLEM IN RECOMMENDER SYSTEMS

Another problem faced by recommendation systems is the cold start [10,13]. Chen and He used the demographic based similarity to find similar customers to a customer who has not yet rated [32]. The similarity method is based on the assumption that if two people are similar in the age, occupation, income, gender or other attributes, then they may have common interest and prefer same types of products. Therefore, the recommendation algorithm filter out customers attributes from the registrations and generate a keyword set of customer attributes. The customer similarity method calculated the demographic similarity between two customers based on the number of common keyword and their weight.

Safoury and Salah suggested utilizing the demographic information of customers to recommend products to new customers in order to eliminate the cold start problem [33]. Safoury and Salah analyzed the demographic attributes of customers and found a set the attributes to

produce more accurate recommendations. Attribute analysis performed some statistical analysis such as distribution of data based on given attributes. Attribute analysis selected a set of attributes for predicting customer preferences.

Wang design a demographic recommender system to recommend attractions to tourists [35]. This system categorizes tourists using their demographic information and makes recommendations based on demographic similarity. Their preliminary results showed improvements in the accuracy of system. However, other information such as textual reviews can also improve the accuracy of recommendation systems.

3.6 OUR ENSEMBLED REOMMENDER SYSTEM MODEL SOLUTION

As explained earlier, the integration of recommendation techniques in graph-based models handles sparsity and cold start problems. Using same concept, we integrate content-based, collaborative filtering and association rule mining techniques in a graph-based model. Our recommender model is an ensemble technique. We used a retail store dataset to show the impact of integrating categorical information of products and demographic information of customers. Unlike other researches, we explored three areas using a retail store dataset.

1. Integrating demographic information of customers, categorical information of products and transactional information.
2. Integrating content-based, collaborative filtering and association rule mining recommendation techniques.
3. Integrating the influence transfer from indirect connections to rank products and produce a list of ranked recommendations. We used PageRank algorithm to rank products.

Chapter Four: **Methodology and Data Characteristics**

4.1 GRAPH BASED RECOMMENDATION SYSTEM MODEL

In this research, we purposed a graph based recommender system to integrate different recommendation techniques. Similar to the approach presented in [20] for digital library, we integrate the content based and collaborative filtering method into our recommender model. Additionally, the association rule mining technique is also integrated in our recommender model and it creates another type of associations between products. Moreover, our recommender system extracts the customer preferences through utilizing three different types of data sources available:

1. Demographic information of customers,
2. Categorical information, of products
3. Transactional information.

In the context of super market, a graph-based model incorporates customer to customer, product to product, and product to customer correlation. The graph-based representation of customers and products allows us to integrate the influence transfer from indirect connection using the PageRank algorithm. The PageRank algorithm generates a ranked list of recommendations for a customer.

Our approach is composed of three stages of computation. In the first stage, we represent customers and products using various approaches. To represent customers efficiently, we clustered customers into categories. The customer groups have either similar demographic properties or similar purchasing patterns. If the customer does not have a purchase history, similarity between two customers is derived using the demographic. Similarity between two customers is derived from the purchasing patterns as well. Therefore, the usage of two different

criteria to find similar customers in a sparse dataset not only handles the cold start problem but also uses a transactional dataset to produce more accurate results for existing customers. Products are represented using products' name directly. However, the patterns in customer preferences are not apparent because customers do not buy a same product frequently. The frequency of buying same products is lower because there are many other options available for a same type of products. Products which have similar type, can have different size, brand or price. Therefore, we group products based on the types of product. Products of same group have same type but different brand names. The integration of category associations between products creates many connections between products and is able to produce connections in a sparse dataset. Additionally, the frequent sets of product's types are also discovered in the first stage of computation using the association rule mining. Therefore, we created five matrices in first stage:

1. User (customer) Similarity Matrix based on demographic properties: This is a matrix of scores that represents the similarity between customers. Each element of the matrix contains a measure of similarity between two customers. The value of element in the matrix is 1 if the two corresponding customers have similar demographic properties. Otherwise, the value of element in matrix is 0.
2. User (customer) Similarity Matrix based on purchasing patterns: This is a matrix of scores that represents the similarity between customers. Each element in the matrix contains a measure of similarity between two customers. The value of element in the matrix is 1 if the two corresponding customers have similar purchasing patterns. Otherwise, the value of element in matrix is 0.
3. Product Similarity Matrix: This is a matrix of scores that represents the similarity between

products. Each element in the matrix contains a measure of similarity between two products. The value of element in the matrix is 1 if the two corresponding products belong to same category. Otherwise, the value of element in matrix is 0.

4. Product-to-Product Association Matrix: This is a matrix of scores that represents the association between products. Each element in the matrix contains a measure of association between two products. The value of element in the matrix is 1 if the two corresponding products belong to a same frequent product set. Otherwise, the value of element in matrix is 0.
5. User-Product Matrix: This is a matrix of scores that represents the relation between products and customers. Each element in the matrix contains a measure of relation between a customer and a product. The value of element in the matrix is 1, 2, 3 or 4 if the given customer purchased the given product. Otherwise, the value of element in matrix is 0.

At the second stage, we model products, customers, and transactions in an extended graph. By using all types of relations between customers and products calculated in first stage, we create links between customers and customers in a two-mode graph. The two types of entities in the two-mode graph are customers and products. The customer-to-customer links have the weight based on the similarity calculated using demographic properties or purchasing properties. Similar to customers, the product-to-product links have the weight, based on the similarity between products using the products' categorical information. Additionally, another type of relation between two products is derived from association rule mining. Therefore, two products purchased frequently together have a link between them and the normalized weight of links is

based on the number of times two products are purchased together. Besides the customer-to-customer and product-to-product links, another type of links is between customers and products based on the purchase history. The normalized weight of a customer to product link is calculated from the number of times a customer bought a product.

In our model, the recommendation activity becomes a graph search task. The five types of links in the graph model are traversed to find products that are associated to a given customer. Many types of graph search methods can be used to identify recommendations. However, we investigate the indirect influence transfer impact using the PageRank algorithm. With the integration of indirect influence transfer process in our recommendation process, the accuracy of finding the customer preferences increases because the prediction procedure gives more importance to reliable connections. For example, the trending or popular products also have more weight towards finding the customer preferences because many customers have liked these products, which indirectly implies the likability or quality of these products. The model of our recommender system is given in Figure 1.

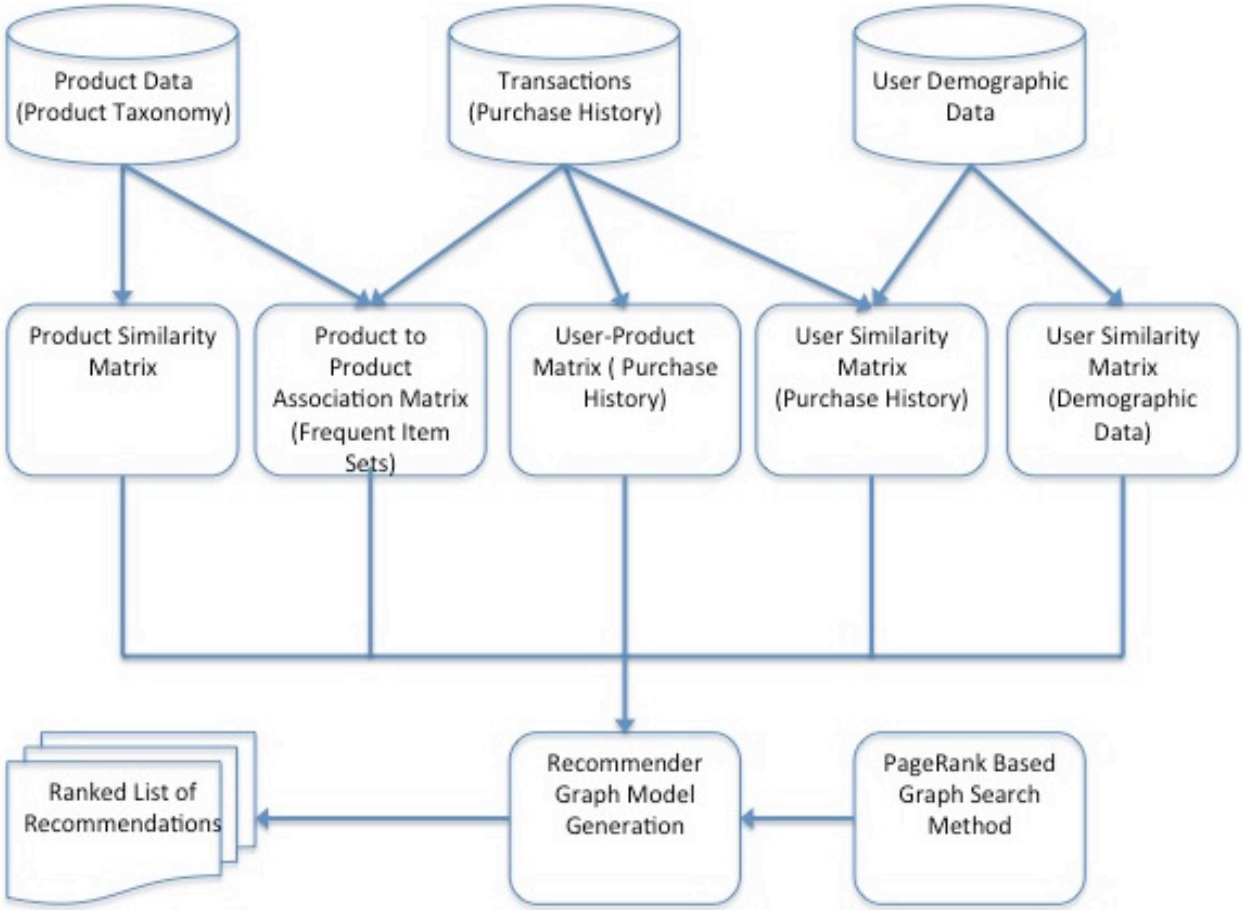

Figure 4-1 Purposed Recommender Model

We believe our model is flexible, comprehensive and modular. Firstly, the weight of links computed in the first stage can be adjusted to reflect the importance of certain aspects of the data. For example, if we want to give more importance to purchase history, we can increase the weight of customer to products links using a certain criteria such as double up the weight. Additionally, we can define similarity of products or customers using a different category of products or customers for a sparse dataset. Similarly, we can define similarity of products or customers using the deeper level of product or customer taxonomy for a dense dataset. The flexibility is supposed because we can control the parameters easily without building new models.

Secondly, this model uses three different recommendation techniques. The content based, collaborative filtering, and hybrid approaches are used in this comprehensive model. We can use only the customer to customers links based on demographic properties or purchase history to make the recommendation purely collaborative. Similarly, only considering the product-to-product links will make this model content based recommendation model. Moreover, the combinations of approaches such as collaborative with association rule mining, content based with association rule mining, leads to many hybrid solutions using only one single model. Using all the techniques present in the model, leads to a comprehensive approach. Our model can handle various types of data using a certain technique for each data type that maximize the utilization of data or information.

Thirdly, this model is modular and allows for future expansions. Since the three computation stages described above are independent from each other, we can use different algorithm at each stage without changing the recommendation model. For example, we can change the algorithms in stage one without affecting the stage two and/or three. We can also use different graph search techniques in stage three for better performances. The modularity of our model allows the comparisons of the different combination of methods as well.

4.2 REPRESENATATION AND ALGORITHM DETAILS

This section explains algorithms used to create our recommendation graph model. It contains details of these five main steps, which are given below.

1. Representing customers and computing similarity between customers based on demographic properties and purchase history,
2. Representing products and computing similarity between products based on product category,
3. Finding the association rules between products to find frequent product sets,
4. Creating a graph network of customers and products, and applying the PageRank to improve customers and products representation through including indirect influences,
5. Using the PageRank algorithm to produce a ranked list of recommendations for a given customer.

4.2.1 Customer Representation and the Similarity Calculation

Customers have two types of information, namely, demographic information and purchasing history. Demographic information contains the location, family, financial, and other types of information such as age, gender etc. Purchasing history is the transactions made by a customer. A transaction has a list of products bought by a customer.

We selected a particular set of demographic attributes. There are two reasons behind choosing certain demographic attributes. Firstly, we usually do not have much demographic information provided by customers. Secondly, choosing the best set of demographic attributes is very computational and complex. Therefore, finding a feasible way to select the demographic attributes is out the scope for this study. We only want to establish relations between customers

based on demographic information to see the impact of demographic information on the accuracy of our recommendation system.

The similarity between customers based on the demographic information is calculated by deciding a threshold value. Two customers are similar if they have same value for all the demographic attributes. Since we have five demographic attributes to group customers, we have 54 groups of customers based on demographic information. In each group of customers, customers have same value for all of the demographic attributes.

Another type of information available for a customer is the transactional information. It has record of all transactions made by that customer. Transactions show the purchasing habits of a customer. The purchasing habits of a customer are usually the information regarding the types of products. The customer likability towards a product is a pattern and the pattern shows if the customer tends to buy a certain type of products more frequently.

In our model, the similarity of purchasing habits between two customers depends on number of similar products purchased by both customers. Additionally, similarity also depends on the total number of products purchased since the relationship between two customers should indicates the probability of buying a similar product out of the total number of products purchased by both customers. Although two customers can purchase many similar products, customers can buy many different products as well. Therefore, a similarity function must consider the number of different products as well. The equation 1 calculates similarity between two customers: customer1 and customer2.

$$S(\text{Customer1}, \text{Customer2}) = (2 * |P1 \text{ and } P2|) / (|P1| + |P2|) \quad \text{.....Equation (1)}$$

Where P1 is the set of products bought by customer1 and P2 is the set of products bought by customer2. The numerator is twice the number of total number of similar products. The denominator is the number of total products bought by both customers. Therefore, the similarity function indicates the probability of buying similar products. For example, similarity between customer1 and customer2 is calculated as S(Customer1, Customer2) is (2*3) / (4+4) = 0.75 and similarity between customer2 and customer3 is calculated as S(Customer1, Customer2) is (2*2) / (4+3) = 0.57 for the customers presented in table 1.

Table 4-4-1: Example

Product	Customer 1	Customer 2	Customer 3
1	1	1	0
2	2	1	1
3	1	2	2
Total	4	4	3

Another factor, which impacts the similarity of customers based on the purchasing patterns, is the frequency of buying similar products. For example, as shown in Table 1, customer1 have purchased 3 distinct products, customer2 purchased 2 distinct products and customer3 have 2 distinct products. Although customer1 and customer2 have more common products, customer2 and customer3 have more similar purchasing patterns since they are buying same quantity of products.

Similar to equation 1, the equation 2 calculates similarity between two customers while considering the quantity of purchased products.

S (Customer1, Customer2) = (2 * |P'1 and P'2|) / (|P'1| + |P'2|)……Equation (2)

Where P'1 and P'2 is the sets of products bought by customer1 and customer2 respectively. Unlike P1, P'1 has the same products multiple times. For the example given in Table 1, customer1's P'1 set consist of {1, 2, 2, 3}, customer2's P'2 set contains {1, 2, 3, 3} and customer3's P'3 set contains {2, 3, 3}. Therefore, |P'1 and P'2| is calculated as 3 and |P'2 and P'3| is 3. The similarity of S (Customer1, Customer2) is (2*3) / (4+4) = 0.75 and S(Customer2, Customer3) is (2*3) / (4+3) = 0.86.

A threshold value is chosen to define if two customers are similar or not. For example, if similarity 10% is used, two customers are similar if the similarity between these two customers is more than 0.10.

4.2.2 Product Representation and the Similarity Calculation

Products are presented in two ways in our recommendation model. Firstly, we consider product names. Because names are associated with brand and such, there are very few links between products derived from association rule mining. However, there are many links between products based on similarity since similarity between products is calculated if products belong to same product category that is the level 3 in products taxonomy present in next section (Data Characteristics). Customers do not tend to buy the same products since there are varieties of products similar to the given product but could have better price or quality. Additionally, each product has distinct name. A product name consist of a brand name and a product type, such as x nuts, y nuts and z nuts. Usually, customers do not show similar purchasing patterns for a particular brand of products in a grocery store. However, customers buy same type of products. Therefore, we categorized products in product types. A particular type of product contains

products of same type but allows different brands. Additionally, the usage of product types shows the impact of grouping products by type on the accuracy of recommendation systems.

4.2.3 Association Rule Mining to find the Frequent Product-Sets

If two product types are getting purchased together frequently, a customer who is purchasing one of the product types will likely be interested in other product types as well. As explained in the background chapter, we used the Apriori algorithm given in [39] to discover the frequent product types sets. The links between product types have the weight equal to the normalized number of times when the given set of product types is bought together. In order to normalize the weight of links, we assigned weight 1 to simplify the process.

4.2.4 Customer-Product Relationship based on Customer Purchasing Patterns

Another types of relationships derived between customers and products are based on the purchasing patterns of customers. A product is linked to a customer if the customer buys the given product. However, customers have more preferences for some products compared to others. There are various reason of variations in the preferences of customers such as some products are more frequently used compared to other products, or customer have some special needs or preferences. Products of type food are more frequently used compared to household products such as cleaning products. Similarly, customers with kids buy baby food products. Therefore, the weight of a link between a customer and a product should depend on the number of times the customer purchased that product. In our recommendation systems, the weight of customer to product links is equal to the number of times the given customer bought the given product. Furthermore, we normalized the weights of links through distributing the weight from 1 to 4.

4.2.5 Graph Network Creation

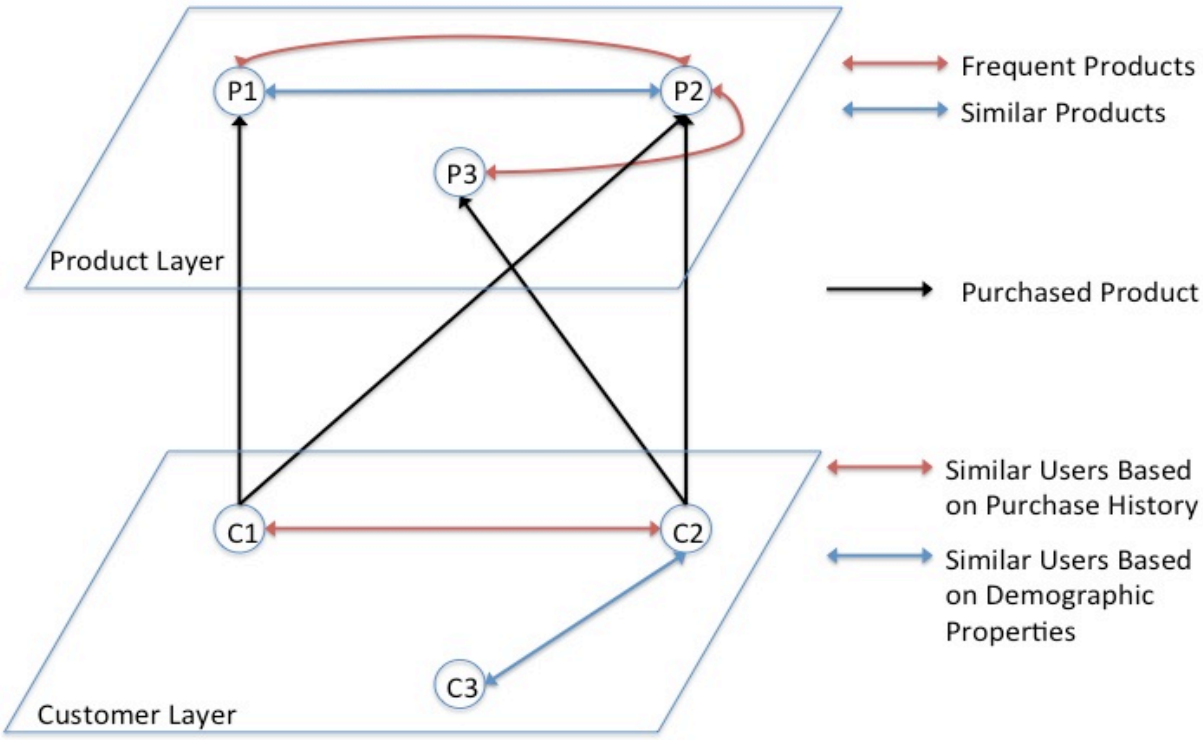

Figure 4-2 Graph Representation of Our Recommender Model

A graph is created using five types of links discovered using above algorithms. There is a link between two customers if they are either similar based on demographic properties or purchasing patterns. A customer can have both types of links to other customers, but the links based on purchasing patterns will be used if the customer has a purchase history. Products have two types of links between them. If two products belong to a same category, they have a link between them. Another type of links between products shows if two products belong to a same frequent product set. The fifth type of links is between a customers and a product. There is a link

between a customer and a product if the customer bought the given product. Therefore, our graph model has multiple relationships between two types of entities: customers and products. The various types of links and graph presentation can be viewed as given in Figure 2.

Furthermore, the PageRank algorithm is used to give an influence value to products to find trending or popular products. The PageRank algorithm depends on incoming links to give the influence value to an entity. Using similar step, the number of links to products is used as incoming links, and a product gets influence values using the PageRank algorithm. We use the PageRank to assign an influence value to each product. We did not apply the PageRank on customers.

4.2.6 PageRank Algorithm based Recommendation Graph Search Method

Searching products and ranking products to create a recommendation list for a customer becomes a graph search activity. In the previous step of creating a graph model for our recommendation system, the PageRank was used to assign a influence value to each product. In this step, we traverse the graph based recommender model to produce a ranked list of recommendation products through extracting the target customer's sub-graph.

4.3 DATA CHARACTERISTICS

In this section, we present the dataset used for the experiment purpose and the characteristics of the dataset. This section consists of two sub sections. In the first section, we give a detailed description of various types of data available in the dataset and discuss their different properties. In the second section, we present problems found in this dataset that form challenges to the recommendation task.

We used the Mondrian's FoodMart, which is a public database stored in Microsoft SQL Server 2000 [17] and obtained in MySQL dump file from [38]. This database contains transactions for retailer across North America and Mexico. In particular, there are 269,720 sales records for 10,281 customers over two-year period (1997-1998). The database contains 1560 distinct branded products (311 unbranded products and five to six different brands for each product).

4.3.1 CHRACTERISTICS OF DIFFERENT TYPES OF DATA

In the following sections, we discussed in details the three different types of data: Customer, Product, and Transactions. We discussed their various properties, which can be used in the recommendation task. The following section is divided into three parts. In the first part, we discussed customers' data. In the second part, we discussed the product data. In the third, we discussed transaction data.

Customers' Data

This dataset contains 10,281 customers' records with 27 attributes which represents demographics information, such as address and contact information, age, gender, marital status, occupation, yearly income, total children and more. We eliminated and merged some of attributes to produce a subset that is informative and useful for groceries stores

recommendations. Attributes such as membership number, name, address, phone, city, state, and postal code were eliminated since they do not contribute to the recommendation task or they are far too specific for a groceries store recommendation. Country on the other hand is potentially more indicative and useful in the recommendation task since customers in different countries will have different needs depending on their culture, holidays, weather and other factors.

Education, occupation, and income attributes were also eliminated since these attributes will have no impact on purchases. None of products can be considered luxurious or expensive, so that income may need to be considered an important factor in the preferences of purchases. Income may impact the quantity of purchases but not the preferences. Also, we do not have products of academic or intellectual value that we may consider education or occupation to be indicators. We also eliminated house owner and number of cars owned attributes for the same reasons.

We eliminated total children attribute since a customer that has a number of children that are not living with him\her would not change his\her purchase habits. On the other hand, having a child at home may impact purchasing habits. Therefore, we extrapolated a new attribute from number of children at home attribute, which indicates if the specific customer has children at home or not. Obviously, if a customer has a child at home, we expect to see few groceries products that indicate that. We kept gender and age for the same reason.

After constraining and merging the attributes, we ended up with a total of five attributes: country, marital status, gender, has children at home, and age group. Based on these attributes we have a total of 54 clusters each cluster represents one unique assignment of the 5 attributes. Figure 3 below shows the number of customers per cluster. From this figure, we can see that the

distribution of customers per cluster is not even. The majority of customers fall under the first 20 clusters.

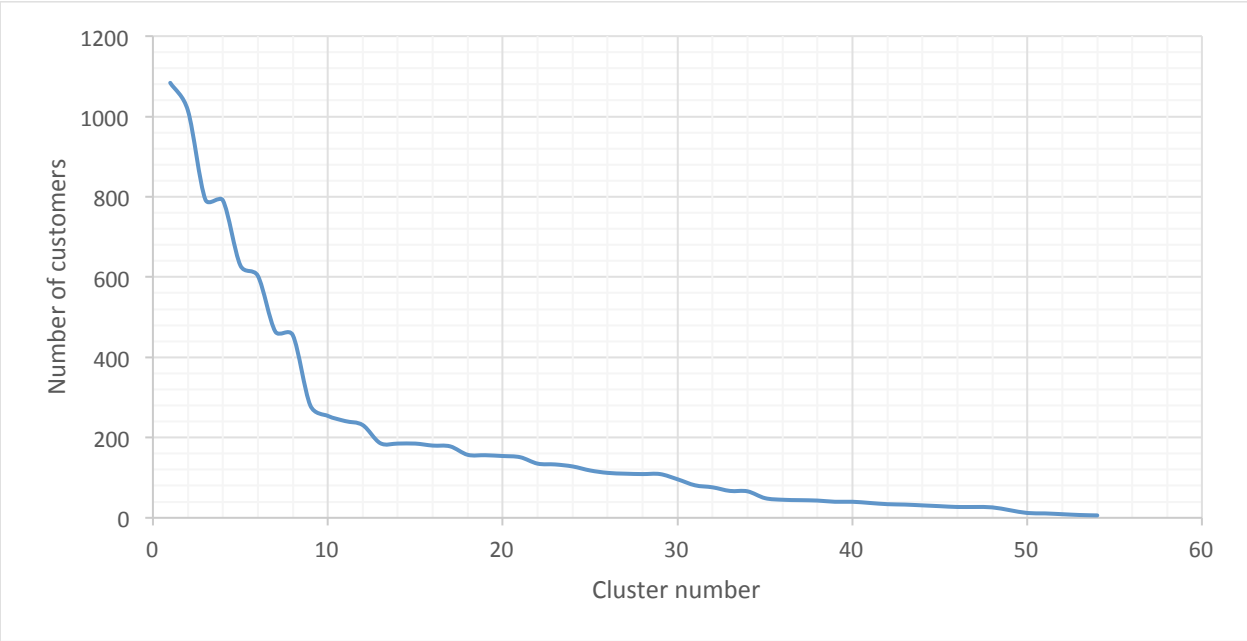

Figure 4-3 Number of customers per cluster

Products' Data

Similar to customers, the dataset contains information about products. Products are associated with six categorical attributes, i.e. product family, department, category, subcategory and brand, as well as other attributes, such as price, weight, and more. The categorical attributes form a hierarchy of 103 classes under which all 1560 products fall. A single product would fall under one and only one class as given in Figure 4 below. The product family attribute divides products into food, drink, and non-consumable products. The product department further divides products into 22 departments, such as Produce, Beverages, and Households. Products are then further divided into 45 categories, such as Meats, Juice, and Hardware and then into 103

subcategories, such as Beagles, Orange Juice, and Candles. At this level each of the 1560 products exists as branded products.

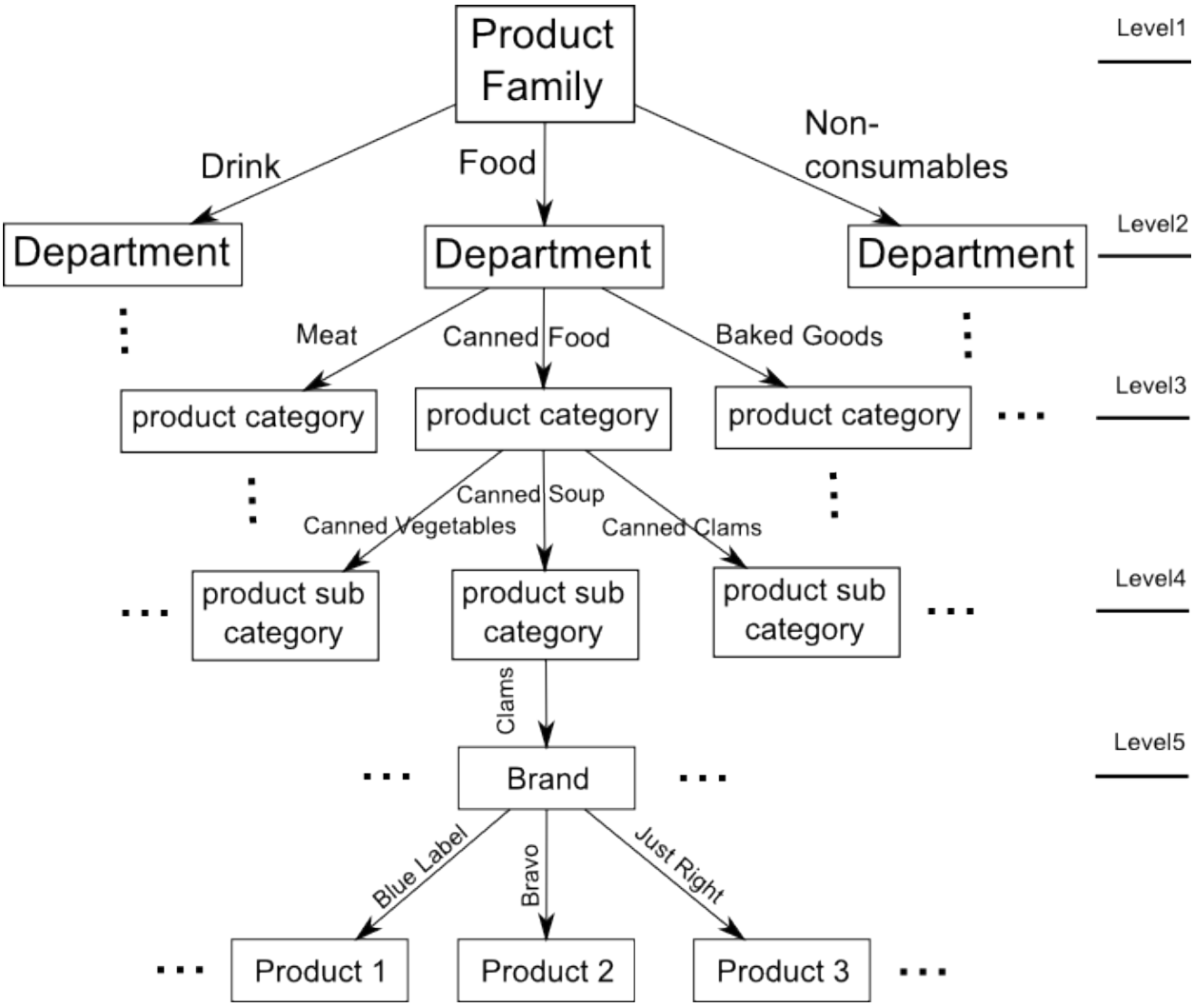

Figure 4-4 Five-level product taxonomy

When considering the fourth level attribute, namely the subcategory attribute, all products with the same subcategory are considered to be in the same class. The number of products in each of these classes differs. Figure 5 below was obtained by listing all 103 classes and the number of products that fall under them. From this figure, we immediately see that the

distribution of number products to number classes is skewed. 51% of the classes contain 5 – 10 products while all the others contain any value from 12 to 120 products.

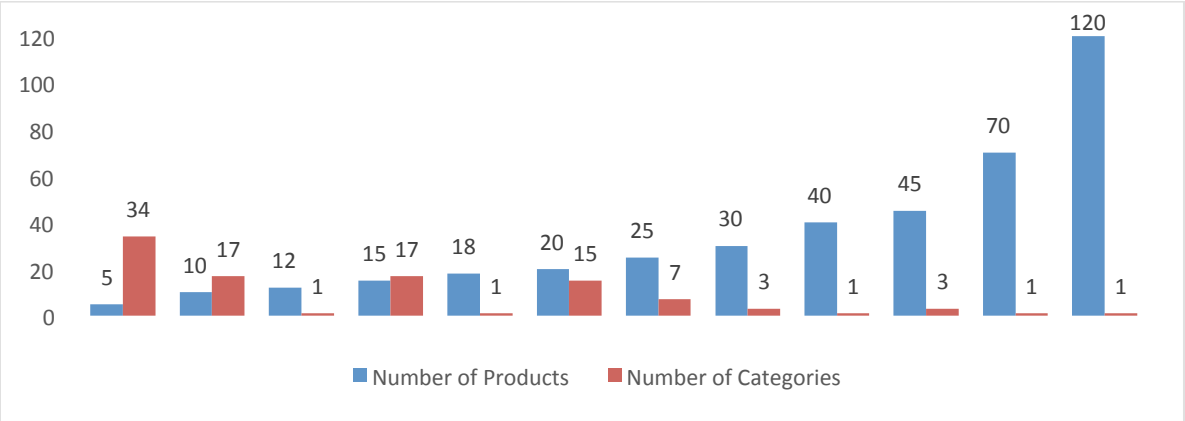

Figure 4-5 Number of Categories Containing a Number of Products

Transactions' Data

The FoodMart dataset contains the transactions for two years from 1997 to 1998 with a total of 58,308 transactions to purchase a total of 269,720 products. Each transaction includes information about purchased product(s), the quantity of the purchased product(s), date and time, location and more.

Table 4-2: Products purchased per quarter

Quarter	*Number of products*
1997 Q1	21588
1997 Q2	20368
1997 Q3	21453
1997 Q4	23428
1998 Q1	44252
1998 Q2	43849
1998 Q3	44993
1998 Q4	49789

In Table 2, we see the total number of products purchased per quarter. The interesting fact is the number of products purchased in 1998 is more than double the transactions in 1997.

Therefore, we may need to consider the first five quarters as the minimum training set of our recommender since it would contain nearly 50% of customers purchased products.

Another interesting fact is in number of transactions, which contains the same product. The graph shows the number of products in Figure 6, which are purchased for certain number of transaction, plotted at x-axis. As shown in the graph, the maximum number of times a product appears on different transaction is 54 which is comparatively a small number compared to 269,720 sales records. Therefore, we have very limited data to find to the purchasing patterns of a particular product.

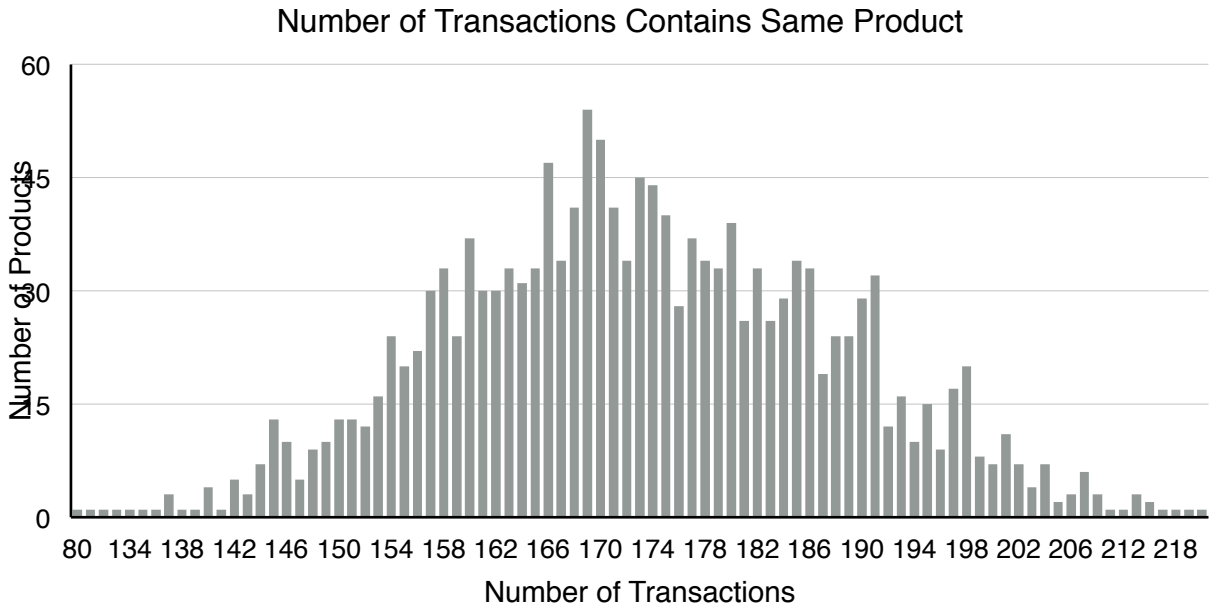

Figure 4-6 Transactions containing the same product

4.3.2 PROBLEMS IN THE DATASET

This dataset has a number of problems, which pose challenges to the recommender task. In the following sections, we discussed these challenges in details.

Branded vs. Non-Branded Products

We discovered an interesting fact about products, which we believe will cause problems in recommender systems. This fact is regarding the branding of products. We have 1560 branded products in this dataset. But the actual number of products is far less. A recommender system that recommends products out of these 1560 will be also be recommending brands. We believe this is too specific for the amount of information we have about products.

This is especially evident in Table 6 where we see the tendency of a customer to buy the same product is very low. On the other hand, if we decided to recommend a product subcategory instead, we will run into another problem. If we recommend product subcategory, we will have to deal with the large deviation of products distribution over these subcategories. From the distribution of products in subcategories Figure 5 above, we see that by recommending few product subcategories to customers, we may be recommending hundreds of products at the same time. Also, these subcategories do not make sense from a customer perspective. Subcategories such as Spices, Pasta, and Deli Meat are too vague and will produce recommendations that are vague. We need to group products into meaningful groups.

It would not be helpful to recommend customers an entire subcategory since this is over generalization and would render recommender system useless. We would most certainly like to recommend a lower level detail than subcategory.

This can pose a serious challenge to a recommender system. This is because the recommender will have to recommend 1560 products at the product level. Or recommend at the subcategory level with 103 subcategories of products. Both choices are bad since in the first

recommender system needs to recommend products along with their specific brand, which is very detailed for a recommender system in this setting. Alternatively, recommender system needs to recommend on the product subcategory level. The problem with this level is that recommending one subcategory could mean to recommend any number of products from 5 to 120, which is quite a distribution.

As a solution to this, we grouped products into a new subcategory and we referred it as product type. Product type is a fifth level division of products where product names were stripped from brand and then grouped. This produced a much healthier distribution of products. Table 3 below shows that we have 306 groups of products each containing 5 products and only 5 other classes, which contains six products.

Table 4-3: Number of Products in Each Category/Class

Number of Products	*Number of classes*
5	306
6	5

Cold Start Problem

First problem we encountered in this dataset is cold start problem. Cold start is defined as the recommendations for products that no one has yet rated. In our dataset, we have two manifestation of the cold start problem. The first is in customers who never made any purchases and the second is in customers who made purchases in the later quarters of the dataset. Table 4 below demonstrates both manifestations. This table shows the number of new customers per year and quarter. New customers in this context are customers who made their first transaction in the given year-quarter and have not made any transactions before that. Customers who made their first transaction in 1997 Q2, for example, are counted as new customers in that year-quarter and

are not counted in the next quarters or the previous ones. This table demonstrates two important facts: First, the total number of customers who did not make any purchases in this dataset is 1439 customers, which is nearly 14% of the entire customer population. Secondly, not all customers in this dataset started purchasing products at the start date of this dataset. Nearly 36% of customers did not make any purchases until the second year of this dataset.

Table 4-4: New Customers per quarter

Quarter	*Number of Customers*
1997 Q1	2981
1997 Q2	1276
1997 Q3	789
1997 Q4	535
1998 Q1	1891
1998 Q2	667
1998 Q3	388
1998 Q4	315
Total	**8842**

We handled the first manifestation of cold start problem by eliminating all customers who did not make any purchases. This is because we are not attempting to predict whether a customer will make a purchase or not. Rather a recommender system merely produces a set of products such that if the customer was to purchase, (s) he is more likely to buy from this recommended set.

The second manifestation of this problem is handled by our approach of recommender system. This is because it is expected from recommender system to be able to recommend to customers even if we lack some aspects of his preferences. We integrate demographic properties of customers to handle this cold start problem as stated in methodology.

Sparsity Problem

We have 3 manifestation of sparsity in this dataset:

1. The majority of customers do not buy often from this groceries store
2. Majority of customers do not buy the same products they have purchased previously
3. Purchased products are not highly associated with each other

Majority of customers do not buy often

This dataset contains 10,281 customers. Figure 7 below shows the percentage of customers who made a certain number of product purchases. That is, this figure shows the number of products purchased and the cumulative percentage of customers who bought the same number of products or less. We can see clearly that 14% of customers have made no transactions and nearly 85% of customers have bought less than 50 products in this dataset. This shows that the majority of customers in this dataset do not buy often from this store. Therefore, a collaborative filtering approach in this case would suffer lack of data.

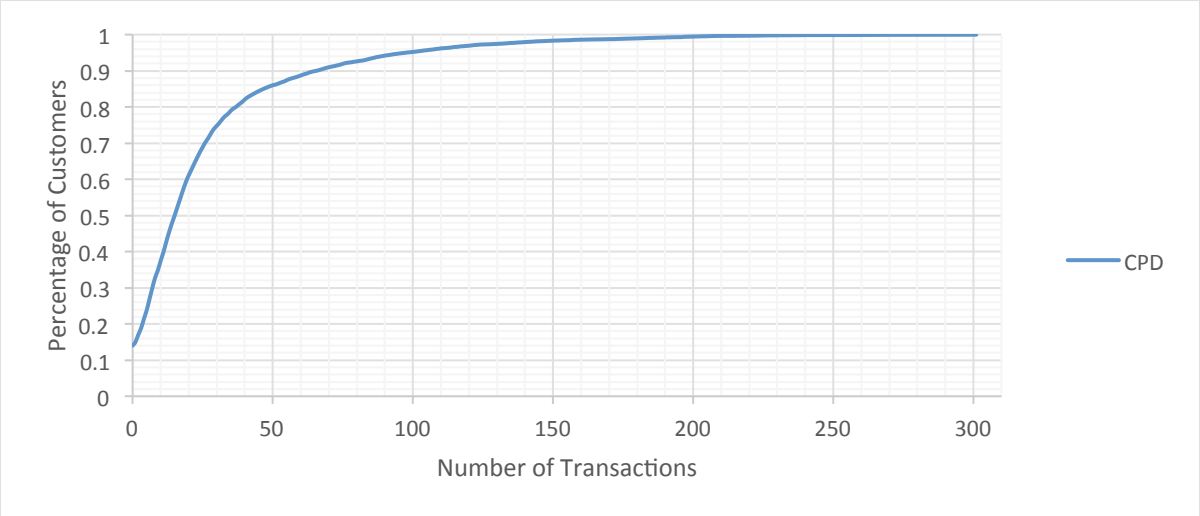

Figure 4-7 Number of Transactions per Customer

Figure 8 is the Lorenz curve, which shows the cumulative percentage of customers, and the cumulative percentage of products purchased those customers own. The straight line shows the expected ratio of product purchases ownership while the curved line the actual observed ownership. This figure also reaffirms our belief. From this figure, you can see that the first 22% of customers own only 1% of the purchased products and the last 20% of customers own over 61% of all the purchased products. Both these figures confirm our statement that customers do not tend to buy often.

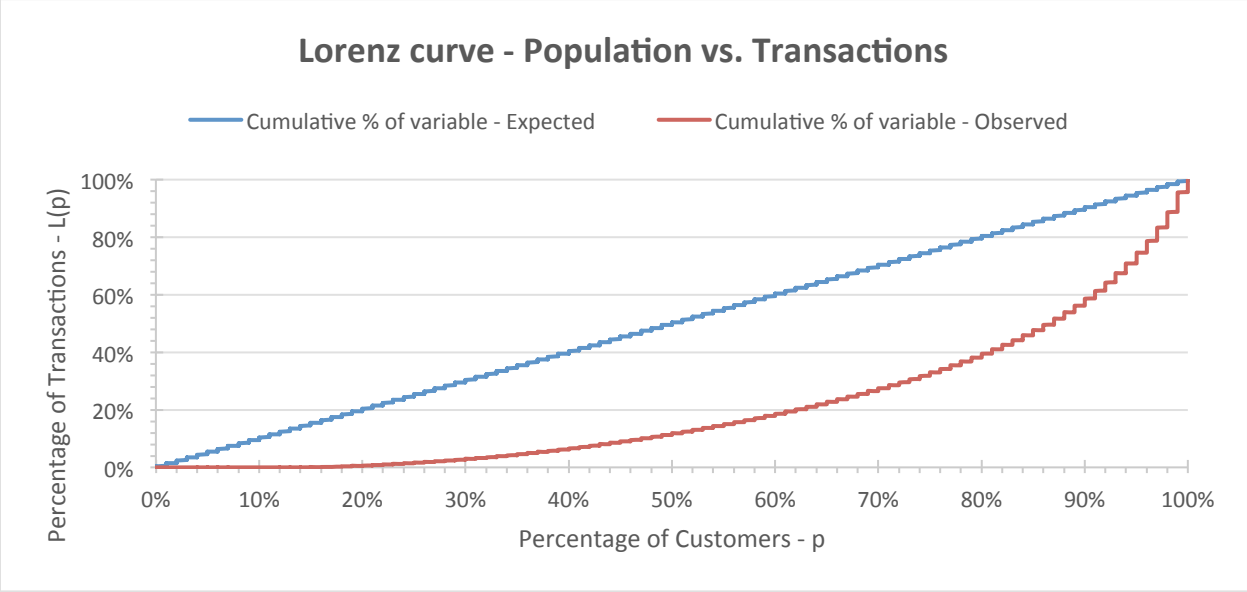

Figure 4-8 Customers vs. Transactions

Majority of customers do not buy the same products they have purchased previously

Table 5 below, shows the number of times any product was purchased by the same customer over the two years period of the dataset. It also shows the number of times this has occurred for all products and all customers. We see that the maximum number of times a branded product was purchased by the same customer is 4 times and this has occurred 7 times in

the database. We see slightly better results with the unbranded products in Table 6. This shows that in this dataset, customers tend to buy diverse products, which mean that we have highly sparse transactions.

Table 4-5: Repetitive Purchases at Product Name Level

Times a branded product is purchased by the same customer	*Times occurred*
4	7
3	299
2	7971
1	252853

Table 4-6: Repetitive Purchases at Product Type Level

Times an unbranded product is purchased by the same customer	*Times occurred*
6	7
5	38
4	275
3	2552
2	21959
1	214469

Another manifestation of the sparse data is in the number of unique products a customer purchased. Table 7 and 8 were constructed by obtaining the number of unique products (branded and unbranded) a customer has purchased divided by the total number of products purchased by the same customer. The result of this division is the percentage of unique products purchased by a customer given all his\her purchases. This percentage is then binned into ten groups, for ease of display, and the count of customers who fall in these categories is provided. This table shows that the probability of a customer going to purchase a unique product (which he has never bought before) is around 97%.

We see in Table 7 a slightly healthier trend of purchases than Table 6. It is evident from this that customers have a slightly higher tendency to purchase the same unbranded product. If

customer buy nuts, it is likely that customer will buy nuts again. However, if a customer buy company x's nuts, it is less likely that customer will buy company x's nuts again.

Table 4-7: Probability of Purchasing Branded Products (Product Name)

Percentage of branded product purchases that are unique	*# Customers who made a purchase*	*Probability*
[100%-90%]	8564	96.86%
[89%-80%]	236	2.67%
[79%-70%]	34	0.38%
[69%-60%]	6	0.07%
[59%-50%]	2	0.02%
[50>]	0	0.00%

Table 4-8: Probability of Purchasing Unbranded Products (Product Type)

Percentage of unbranded product purchases that are unique	*# Customers who made a purchase*	*Probability*
[100%-90%]	7194	81.36%
[89%-80%]	1348	15.25%
[79%-70%]	263	2.97%
[69%-60%]	35	0.40%
[59%-50%]	2	0.02%
[50>]	0	0.00%

The product dataset have 1560 unique products and the highest number of unique products bought by a customer is 282. As shown in the graph, 1648 customers usually buys from 1 to 150 unique products and only 75 bought more than 150 unique products. The median number of unique product purchased is 18. Therefore, the size of the recommendation list must be constraint to a specific number, which meets the customer preferences while not producing a long list of recommendations.

Purchased products are not highly associated with each other

Another issue that we noticed in this dataset is in the associations between products (branded or unbranded). The largest number of times any two branded products purchased together is 14 times and this occurred for only two combinations of products as shown in Table 9. We performed association rule mining on the transactions and extracted all associations of size

2 and support value >=2. We found that customers in this dataset tend to not buy the same products together.

When we ran the association rule mining on the unbranded products as presented in Table 10, we found a very different trend. The largest number of times any two unbranded products purchased together is 45 times which is much larger than that of the branded products. Generally, the associations between the unbranded products is demonstrates a much healthier trend than that of branded.

Table 4-9: Frequent Product-Sets of Branded Products (Product Name)

Support	***Combinations of Products***
14	2
12	4
11	14
10	49
9	138
8	356
7	919
6	2556
5	6455
4	15691
3	36076
2	82258

Table 4-10: Frequent Product-Sets of Branded Products (Product Name)

Sup	*Combinations of Products*	*Sup*	*Combinations of Products*	*Sup*	*Combinations of Products*	*Sup*	*Combinations of Products*
45	1	**30**	65	**20**	1086	**10**	2631
39	1	**29**	108	**19**	1283	**9**	2610
38	2	**28**	128	**18**	1448	**8**	2772
37	2	**27**	175	**17**	1625	**7**	2860
36	7	**26**	254	**16**	1796	**6**	3072
35	6	**25**	354	**15**	1961	**5**	3133
34	11	**24**	480	**14**	2192	**4**	2994
33	12	**23**	589	**13**	2349	**3**	2625
32	19	**22**	747	**12**	2419	**2**	1827
31	48	**21**	873	**11**	2492		

Chapter Five: **Experiment and Analysis**

The system is mainly implemented using the data integration tool, Kettle that is provided by Pentaho. The Kettle package provides the Spoon application, which allows us to establish the connection to MYSQL and integrate the batch script file to execute the algorithms code in jar files. In this chapter, we present experiments performed to evaluate our system. Firstly, we describe the evaluation matrix that is used to evaluate our system based on the accuracy of the results. Results are obtained using different combinations of variables setting. Finally, results are graphed and discussed.

5.1 EVALUATION MATRIX

Three evaluation measures are used to evaluate the performance of proposed recommendation model: precision, recall and F1-measure. As stated in the survey of accuracy evaluation metrics of recommendation systems in [40], the accuracy of predicting good recommendations not only depends on the number of good predictions but also on the number of bad recommendations as well. According to Powers in [42], Recall or Sensitivity is a proportion of real positive cases that are predicted positive. This measures the coverage of real positive cases. Its desirable feature is that it reflects the number of relevant cases the system picks up. Precision or Confidence denotes the proportion of predicted positive cases that are correctly real positives [42]. Precision is a true positive accuracy through being a measure of accuracy of predicted positives in contrast with the rate of discovery of real positives [42]. F1-measure or F-score references the true positives to the arithmetic mean of predicted positives and real positives. F1-score is being a constructed rate normalized to an idealized value, and expressed in this form known as a Proportion of Specific Agreement [42].

In our recommender system, True-Positive is the numbers of products that are bought by a customer and recommended to the customer as well. False-Positive is the number of products that are not bought by a customer but are recommended to the customer. True-Negative is the number of products preferred by a customer but not recommend to the customer. Based on the three different scenarios, precision shows the accuracy of producing correct recommendation in comparison to the total number of recommendations produced. Recall is a ratio of the number of correct recommendations to the number of products actually bought by a customer. Precision and recall are inversely related to each other in this case. The increase in total number of recommendations increases recall but decreases precision and vice versa. Therefore, F1-measure is used to evaluate the results based on both precision and recall to check the test accuracy.

$$\text{Precision} = N_{rs} / N_s$$

$$\text{Recall} = N_{rs} / N_r$$

Where N_{rs} is total number of products produced by recommendation system and bought by the customer, N_s is the total number of recommendation produced by recommendation system, and N_r is total number of products bought by the customer.

$$F1 = 2 * (Precision * Recall) / (Precision + Recall)$$

Precision, recall and f_score are calculated for each customer in the dataset. In order to compare the results of different models, we calculated the averages of precision, recall and f_score.

5.2 CROSS VALIDATION BASED EXPERIMENTS

We train our model on a specific date range and then test on remainder, which must happen before the prediction date range. We cannot predict the next time a customer will make a purchase in a store since we need far more data on the lives of customers to able to model their needs, moods, vacations, and all that may impact their decision.

Additionally, the recommendation task is to predict customers' future purchases for this given dataset. There was no recommendation system was in place when this data was collected. So, we cannot say with any reasonable amount of confidence that a customer will buy a product for a certain reason but shear necessity.

Before performing our evaluation, we split the dataset into a training set and a testing set. Since we have 2 years of data [8 quarters], we split our data into a number of sequential quarters for training and the remaining quarters for testing. We have seven possible configurations: [1-7, 2-6, 3-5, 4-4, 5-3, 6-2, 7-1], where the first number is the number of quarters in training set, and the second number is the number of quarters in the testing set. For rest of the chapters, we will refer to these configurations as case 1 to 7 as given in Table 1.

Table 5-1: Different Configuration Data Setting for the Experiments

Case Number	*Configuration Setting* *[[Quarters in training], [Quarters testing]]*
1	[[1997Q1], [1997Q2, 1997Q3, 1997Q4, 1998Q1, 1998 Q2, 1998 Q3, 1998 Q4]]
2	[[1997Q1, 1997Q2], [1997Q3, 1997Q4, 1998Q1, 1998 Q2, 1998 Q3, 1998 Q4]]
3	[[1997Q1, 1997Q2, 1997Q3], [1997Q4, 1998Q1, 1998 Q2, 1998 Q3, 1998 Q4]]
4	[[1997Q1, 1997Q2, 1997Q3, 1997Q4], [1998Q1, 1998 Q2, 1998 Q3, 1998 Q4]]
5	[[1997Q1, 1997Q2, 1997Q3, 1997Q4,

	1998Q1], [1998 Q2, 1998 Q3, 1998 Q4]]
6	[[1997Q1, 1997Q2, 1997Q3, 1997Q4, 1998Q1, 1998 Q2], [1998 Q3, 1998 Q4]]
7	[[1997Q1, 1997Q2, 1997Q3, 1997Q4, 1998Q1, 1998 Q2, 1998 Q3], [1998 Q4]]

5.3 EXPERIMENT ANALYSIS AT PRODUCT NAME LEVEL

In order to analyze our recommendation model, we divide our analysis into three steps.

1. In the first step, we analyze the impact of integrating different techniques into a basic recommendation model.
2. In the second step, we analyze the impact of using demographic information of customers in a recommendation model to handle the cold start problem.
3. In the third step, we analyze the impact of influence transfer from indirect connections using a graph-based recommendation model.

5.3.1 IMPACT OF DIFFERENT TECHNIQUES

Firstly, we want to see the impact of integrating the different techniques such as frequent set mining, content based, and collaborative filtering on a recommender model. Therefore, we created a base model and it recommends the previously purchased products. We also created models using content-based, collaborative filtering and frequent product set mining techniques.

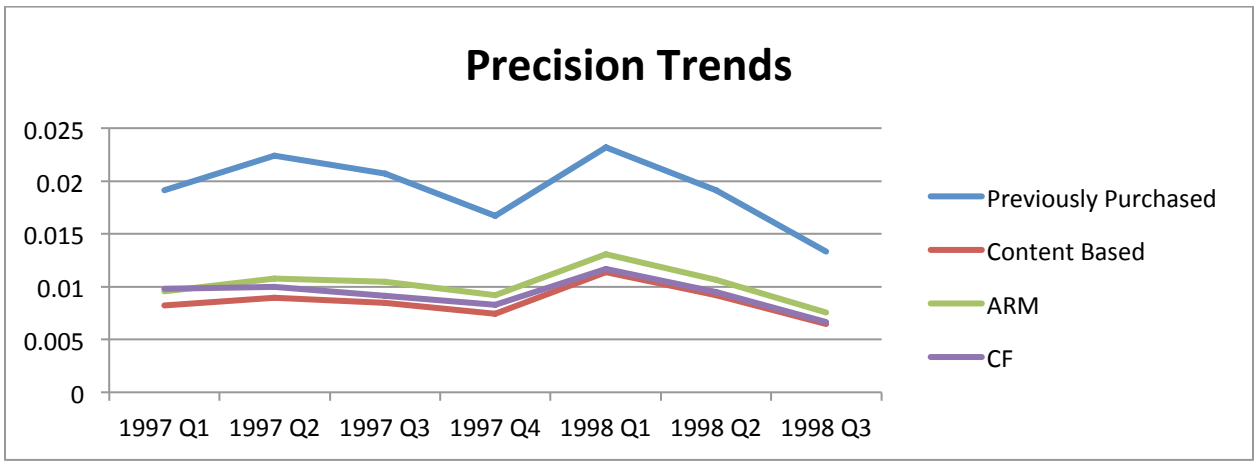

Figure 5-1 Precision Trends in Recommendations Models

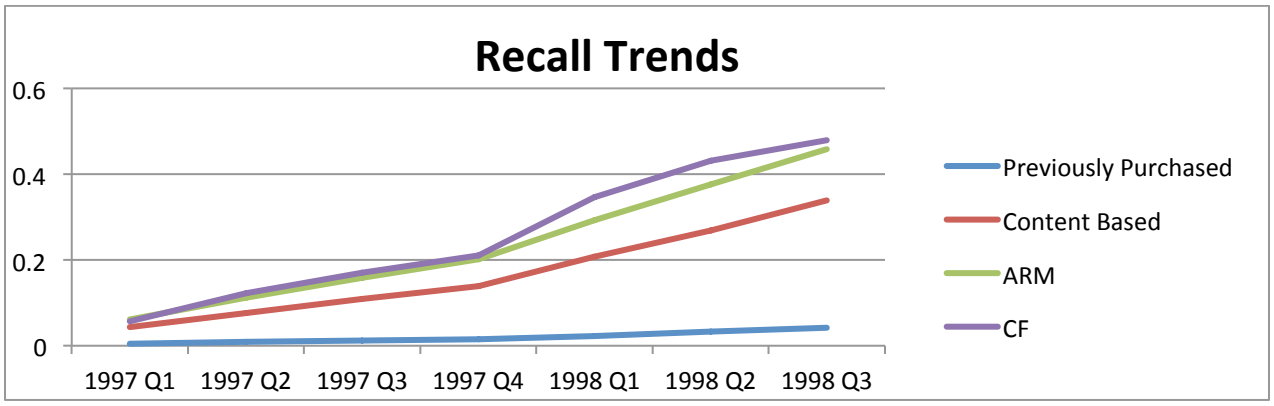

Figure 5-2 Recall Trends in Recommendations Models

There are some common trends in all models. The first trend decreases in precision, when the number of quarters in training set increases and the number of quarters in testing set

decreases. Since we have only 8 quarters for experiments, we increased the number of quarters in training sets and it leaves fewer numbers of quarters in the testing sets. The decrease in testing data lowers the number of transactions and the number of average products purchased by a customer. As shown in the Table 2, the average number of products purchased by a customer is decreasing from case 1 to case 7. Additionally, the increase in training data extracts more of the customers' preferences and increases the numbers of recommendations. As shown in the Table 3, the number of total recommendations in each model is increasing from case 1 to case 7.

Table 5-2 Average Number of Purchased Products by A Customer in Each Quarter

Time Period	***Number of Purchased Products***
1997 Q1	27.53
1997 Q2	25.73
1997 Q3	23.97
1997 Q4	22.13
1998 Q1	18.26
1998 Q2	14.24
1998 Q3	9.88

Table 5-3 Number of Correct and Total Recommendations Produced by Each Technique

Time Period	***Previously Purchased***		***CB***		***ARM***		***CF-Purchase***	
	Correct	**Total**	**Correct**	**Total**	**Correct**	**Total**	**Correct**	**Total**
1997 Q1	0.15	2.41	1.96	61.17	2.49	69.7	1.89	74.42
1997 Q2	0.25	4.69	2.91	110.07	3.83	126.99	2.86	154.45
1997 Q3	0.31	7.13	3.44	155.37	4.57	180.65	3.32	217.08
1997 Q4	0.34	9.84	3.54	199.56	4.72	233.68	3.62	270.68
1998 Q1	0.52	15.56	5.6	299.74	7.22	351.94	6.96	462.38
1998 Q2	0.58	22.23	5.57	391.12	7.2	462.23	7.45	585.21
1998 Q3	0.51	31.75	4.46	494.74	5.71	588.27	6.03	678.82

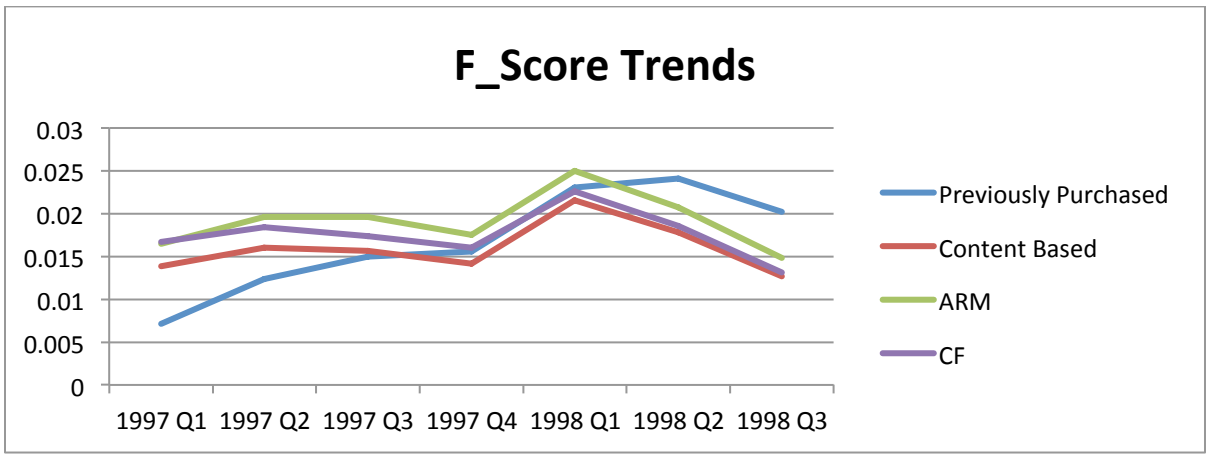

Figure 5-3 F_Score Trends in Recommendations Models

Precision should decrease since the increase in correct number of recommendations is relatively less in comparison to the increase in total number of recommendations. However, the trend of decreasing precision breaks when the first and second quarters of 1997 are in training set and rest are in testing set, and the case when all the 4 quarters of 1997 and first quarter of 1998 are in training set and rest are in testing set. Many new customers enter in 1st and 2nd quarter of 1997 and 1st quarter of 1998, and this lead to a lower number of new customers in testing data. Lower number of new customers increases precision since a recommendation system can predict more accurately for existing customers. Additionally, the number of transactions in 1st quarter of 1998 is 44252 is relatively higher in comparison to the total number of transaction in the quarters of 1997. A recommender system can extracts more of the customers' preferences if a large set of transactions is available. Therefore, these two quarters breaks the trend of diminishing precision.

Table 5-4 Precision of different techniques

	Previously Purchased	Content Based	ARM	CF
1997 Q1	**0.01912**	0.00825	0.00954	0.00978
1997 Q2	**0.02238**	0.00897	0.01077	0.00997
1997 Q3	**0.02070**	0.00844	0.01044	0.00916
1997 Q4	**0.01673**	0.00745	0.00917	0.00831
1998 Q1	**0.02321**	0.01136	0.01307	0.01167
1998 Q2	**0.01910**	0.00920	0.01065	0.00948
1998 Q3	**0.01332**	0.00646	0.00754	0.00665

Table 5-5 Recall of different techniques

	Previously Purchased	Content Based	ARM	CF
1997 Q1	0.00438	0.04282	**0.06133**	0.05691
1997 Q2	0.00855	0.07673	0.11207	**0.12185**
1997 Q3	0.01170	0.10852	0.15866	**0.16998**
1997 Q4	0.01457	0.13864	0.20197	**0.21138**
1998 Q1	0.02295	0.20733	0.29269	**0.34701**
1998 Q2	0.03259	0.26910	0.37588	**0.43169**
1998 Q3	0.04175	0.33894	0.45809	**0.47935**

Table 5-6 F_Score (F1) of different techniques

	Previously Purchased	Content Based	ARM	CF
1997 Q1	0.00713	0.01384	0.01651	**0.01669**
1997 Q2	0.01237	0.01605	**0.01965**	0.01843
1997 Q3	0.01495	0.01566	**0.01960**	0.01739
1997 Q4	0.01558	0.01414	**0.01754**	0.01599
1998 Q1	0.02308	0.02155	**0.02503**	0.02258
1998 Q2	**0.02408**	0.01780	0.02072	0.01856
1998 Q3	**0.02020**	0.01268	0.01483	0.01313

After observing two common trends, we compared models to each other. As we can see, the precision of previously purchased model is better for every case, but other models have higher recall for all cases. Therefore, the previously purchased model has higher ratio of the correct recommendations to the total number of recommendations, but this model have very lower number of total recommendations and only covers very limited customer preferences. However, the integration of other techniques in previously purchased model generates better recall values for all the cases and f_score for first five cases.

Recommending all products is not a feasible approach because it creates the problem of information overload. However, we have to generate a sufficient number of recommendations while generating correct recommendations. Additionally, f_score of content-based, collaborative filtering and frequent set mining for first five cases have shown better performance since these techniques discovered customers' preferences more efficiently in the sparse data. Therefore, the integration of other recommendation techniques explore many other types of relations such as content based, which shows if a customer buys same type of products, collaborative filtering, which shows if the customer have similar preferences to other customer, and frequent set mining, which shows if customers buys trending or popular products or set of products.

Furthermore, frequent product set mining based recommendation model has the highest f_score for most of the cases and this shows that the customers buy trending or popular products. The content-based recommender model has lower f_score in comparison to collaborative filtering and frequent set mining based models and this suggests that customers' preferences are more descriptive through trending or popular products and similar customers' preferences.

5.3.1.1 Impact of Support in the Frequent Product-Set Mining Based Model

A higher support creates less number of associations between products and vice-versa. There is a significant drop in recall from support count 2 to support count 4 and from support count 4 to count 6 because higher support count limits the number of connections between products. Lower number of connections means lower number of frequent sets and this leads to lower number of recommendations. The fall in precision from support count 4 to support count 2 is less significant compared to the fall in precision from support count 6 to support count 4. Therefore, support count 4 is balancing the precision and recall better than other supports. In other words, support

count 4 is able to produce better recommendations since support count 4 will have the correct coverage of customers' preferences while discovering the sufficient customers' preferences.

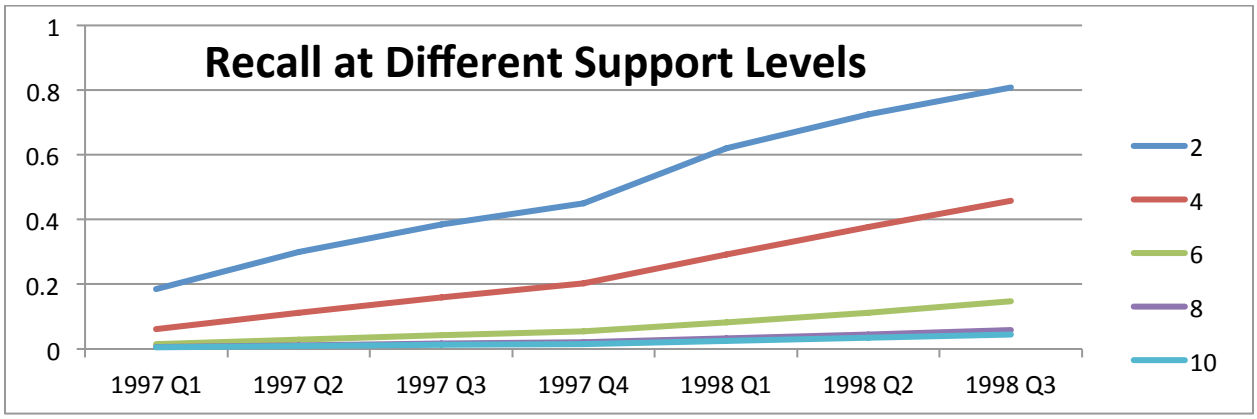

Figure 5-4 Recall at Different Support Counts

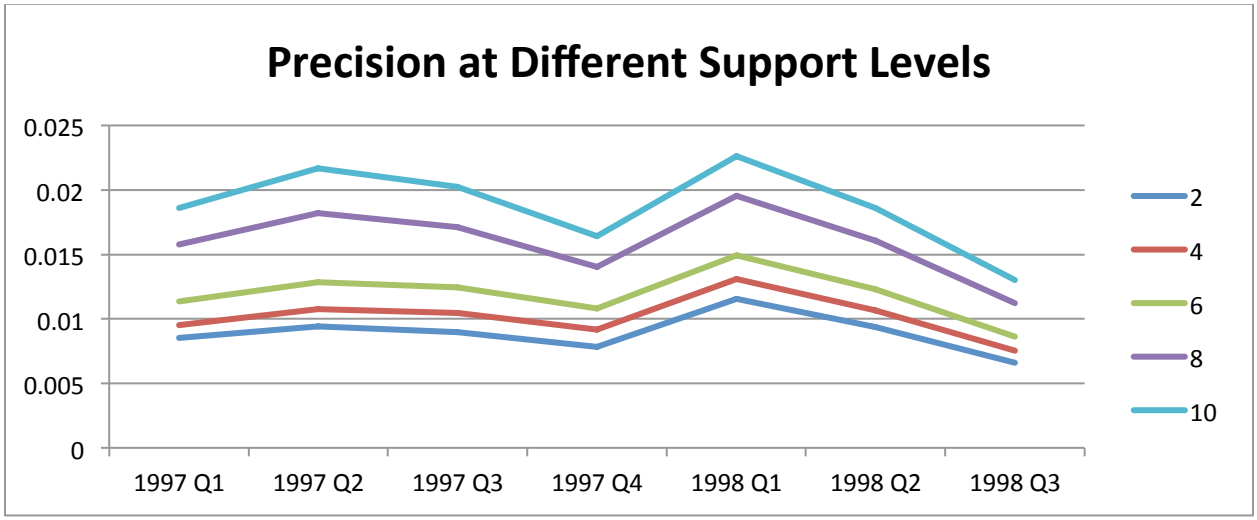

Figure 5-5 Precision at Different Support Counts

Support count 4 has highest f_score for first three cases, support count 6 has highest f_score for 4th and 5th case, and support count 10 has highest f_score for last two cases. Therefore, lower support count is better when dataset does not have sufficient number of transactions to establish association rules with higher support. On the other hand, if a dataset has sufficient

number of transactions to establish associations with higher support count, the usage of higher support count would produce better results.

Table 5-7 F_Score (F1) at different support counts

	2	*4*	*6*	*8*	*10*
1997 Q1	0.01626	**0.01651**	0.01288	0.00869	0.00738
1997 Q2	0.01824	**0.01965**	0.01785	0.01420	0.01274
1997 Q3	0.01751	**0.01960**	0.01926	0.01669	0.01541
1997 Q4	0.01536	0.01754	**0.01804**	0.01671	0.01596
1998 Q1	0.02267	0.02503	**0.02530**	0.02425	0.02335
1998 Q2	0.01854	0.02072	0.02212	0.02365	**0.02403**
1998 Q3	0.01309	0.01483	0.01626	0.01882	**0.02008**

5.3.1.2 Impact of Different Similarities in the Collaborative Filtering Model

Another interesting fact is the increase in precision and decrease in recall when we increase the similarity value. However, the increase in similarity value decreases recall more significantly and decreases f_score as well.

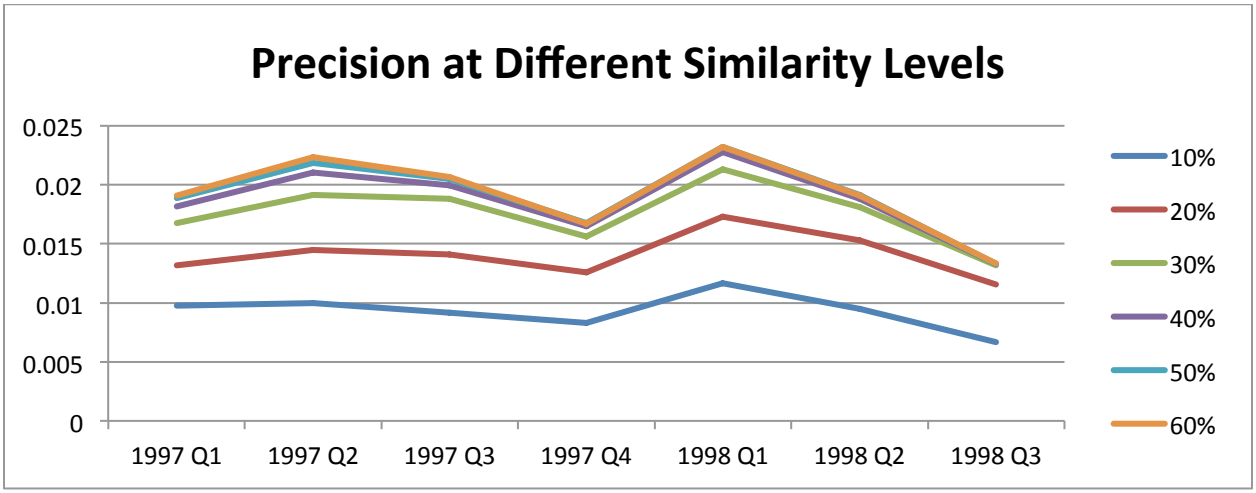

Figure 5-6 Precision at Different Similarity Levels

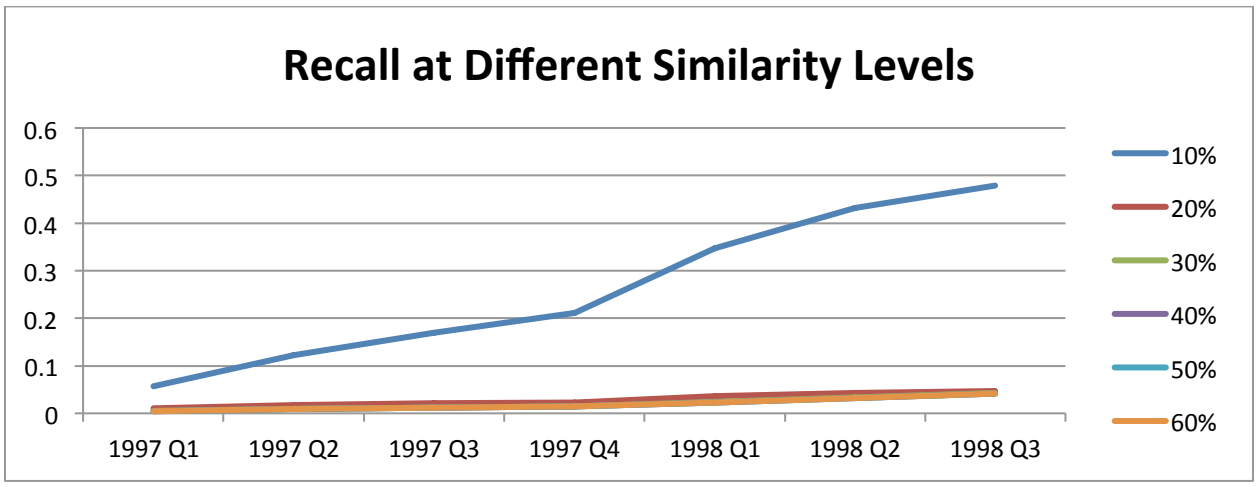

Figure 5-7 Recall at Different Similarity Levels

Similarity 10% has highest f_score for first three cases when data is very sparse. Similarity 20% performs better for case 4 and 5 since similarity 20% can establish sufficient number of stronger connections between customers in comparison to other similarity values. This shows the impact of similarity value on a recommendation model and a recommendation model can choose a similarity value that can establish sufficient number of stronger connections to produce accurate results.

Table 5-8 F_Score (F1) at different similarity level

	10%	*20%*	*30%*	*40%*	*50%*	*60%*
1997 Q1	**0.01669**	0.01171	0.00729	0.00717	0.00715	0.00715
1997 Q2	**0.01843**	0.01594	0.01246	0.01235	0.01236	0.01239
1997 Q3	**0.01739**	0.01696	0.0149	0.01488	0.01495	0.01495
1997 Q4	0.01599	**0.01617**	0.01529	0.01554	0.01561	0.01558
1998 Q1	0.02258	**0.02328**	0.02245	0.02305	0.02317	0.02313
1998 Q2	0.01856	0.02247	0.02352	0.0239	**0.02413**	0.02408
1998 Q3	0.01313	0.01858	0.02008	0.02019	0.02015	**0.0202**

5.3.1.3 Impact of including the Quantity based Customer Similarity in Collaborative Filtering Model

Table 5-9 Comparison of CF Model and CF Quantity based Similarity Model for Similarity 10% and 20%

	Precision			*Recall*			*F_Score*		
Similarity 1	**CF (1)**	**CF With Quantity (2)**	**(1 - 2)**	**CF (1)**	**CF With Quantity (2)**	**(1 - 2)**	**CF (1)**	**CF With Quantity (2)**	**(1 - 2)**
1997 Q1	0.05691	0.05666	0.00025	0.00978	0.00978	0	0.01669	0.01668	0.00002
1997 Q2	0.12185	0.1198	0.00205	0.00997	0.00999	-0.00002	0.01843	0.01844	-0.00001
1997 Q3	0.16998	0.16582	0.00417	0.00916	0.00921	-0.00005	0.01739	0.01745	-0.00006
1997 Q4	0.21138	0.20257	0.00881	0.00831	0.00834	-0.00003	0.01599	0.01602	-0.00003
1998 Q1	0.34701	0.33167	0.01535	0.01167	0.01174	-0.00007	0.02258	0.02268	-0.00010
1998 Q2	0.43169	0.40561	0.02608	0.00948	0.00953	-0.00004	0.01856	0.01862	-0.00006
1998 Q3	0.47935	0.44884	0.03051	0.00665	0.00672	-0.00006	0.01313	0.01324	-0.00011
Similarity 2									
1997 Q1	0.01055	0.01053	0.00002	0.01316	0.01316	0	0.01171	0.01170	0.00001
1997 Q2	0.01776	0.01767	0.00009	0.01447	0.01448	-0.00001	0.01594	0.01592	0.00003
1997 Q3	0.0213	0.02109	0.00021	0.01408	0.01414	-0.00006	0.01696	0.01693	0.00003
1997 Q4	0.02258	0.02226	0.00032	0.01259	0.01269	-0.00010	0.01617	0.01616	0.00001
1998 Q1	0.03553	0.03471	0.00082	0.01732	0.01734	-0.00003	0.02328	0.02313	0.00016
1998 Q2	0.04249	0.04198	0.00051	0.01527	0.01536	-0.00009	0.02247	0.02249	-0.00002
1998 Q3	0.04739	0.04707	0.00032	0.01156	0.01165	-0.00009	0.01858	0.01867	-0.00009

The integration of quantity based customer similarity increases precision of our collaborative filtering model since it generates more of the correct recommendation. However, the fall in recall shows that quantity based customer similarity decreases the coverage of customer preferences. Moreover, there is improvement in f_score for similarity 10% but f_score of similarity 20% decreases when we use the quantity based similarity. Since similarity 10% performing better compared to other similarities, we can further enhance similarity 10% performance with the integration of quantity based customer similarity.

The reason is no significant improvement with integration of quantity because customers

buy different products and quantity does not play a significant role to establish the connection strength. Therefore, the number of similar products is more important than quantity of those products when quantity does not deviate at significant level. As shown in data characteristics chapter, the highest number of times a customer buys a same product is 4.

5.3.2 IMPACT OF USING THE DEMOGRAPHIC INFORMATION

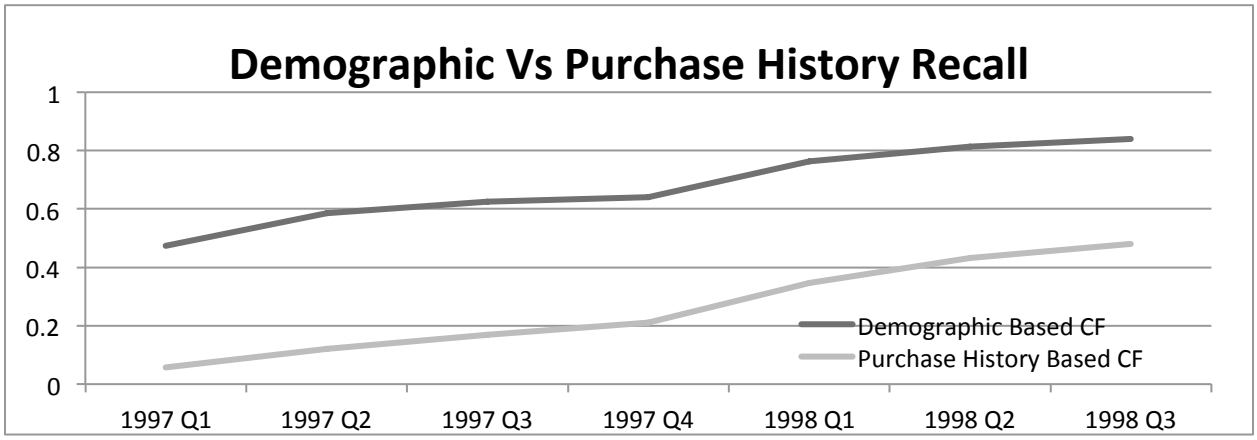

Figure 5-8 Demographic Vs Purchase History Recall

The usage of demographic properties to find similar customer in our collaborative filtering approach, leads to higher recall in comparison to our initial model. Therefore, a collaborative filtering recommendation model can accurately predict the purchasing behavior of customers using demographic properties of a customer. Although, the recommendation model based on demographic properties does not match the precision of recommendation model based on purchasing patterns of customers, this model is very useful to predict recommendation for new customers thus eliminating the cold start problem.

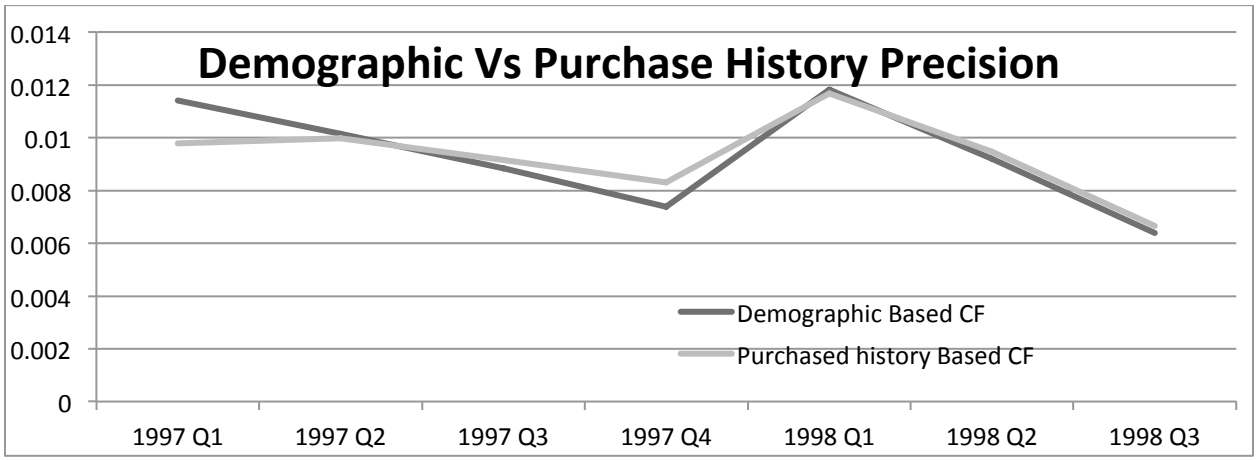

Figure 5-9 Demographic Vs. Purchase History Precision

Another interesting observation is the increase in precision of this model after first quarter of 1997 instead after 2nd quarter of 1997. The first quarter of 1998 introduces many new customers and this recommendation model is able to predict recommendations for new customers based on demographic properties of customers. This factor supports our argument that this model can predict for new customers and produce more accurate results as well.

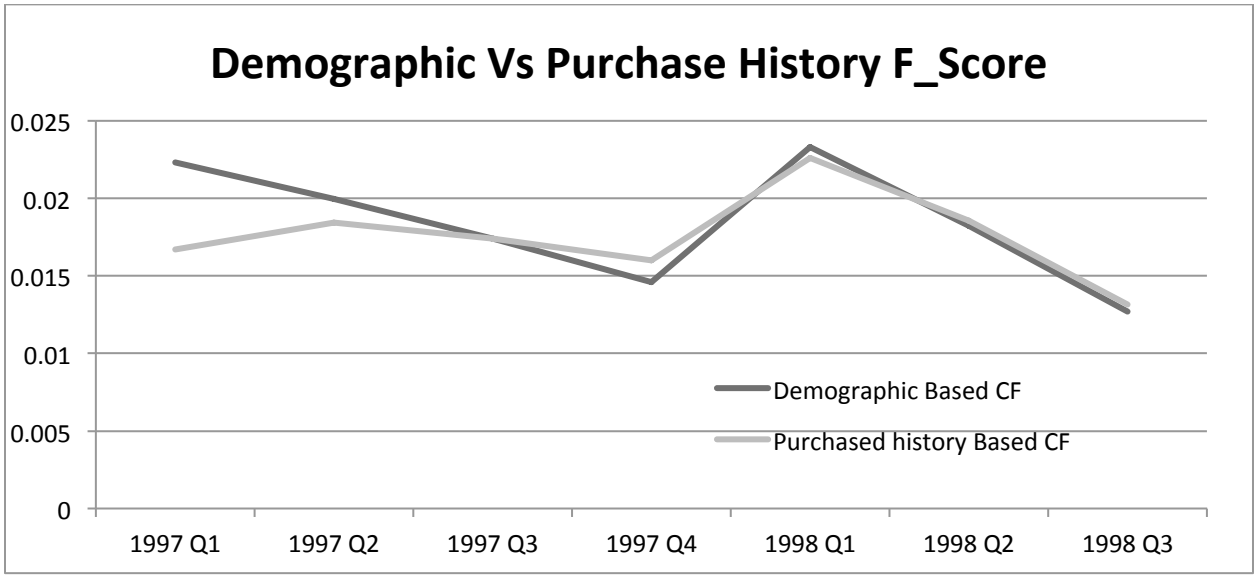

Figure 5-10 Demographic Vs. Purchase History F_Score

5.3.3 IMPACT OF INDIRECT CONNECTIONS

To analyze the impact of indirect influence, we compared the ARM model to the graph-based ARM model and CF model to graph-based CF model where graph-based model uses the indirect connections to generate a list of recommendations. The usage of indirect connections increases total number of recommendations and recall value as shown in Table 10 and 11. The increase in recall suggests that indirect connections can explains customers' preferences as well. However, precision and f_score of ARM model decreases since the total number of recommendations generated though graph-based ARM model have relatively higher number of products, which are not purchased by a targeted customer.

Table 5-10 Comparison of Graph-based ARM Model to ARM model for Support count 4

	Number of Recommendations				
Graph Based ARM Model	Correct	Total	Precision	Recall	F_Score
1997 Q1	11.79621	515.28769	0.00757	0.33009	0.01479
1997 Q2	12.78402	729.52589	0.0082	0.4676	0.01612
1997 Q3	12.12542	853.92501	0.00778	0.54745	0.01534
1997 Q4	10.65074	928.64777	0.00683	0.59541	0.01351
1998 Q1	16.50054	1,272.65	0.01058	0.81621	0.0209
1998 Q2	13.48434	1,392.44	0.00865	0.8931	0.01713
1998 Q3	9.50608	1,461.10	0.0061	0.93716	0.01212
ARM Model	Correct	Total	Precision	Recall	F_score
1997 Q1	2.49389	69.69917	0.00954	0.06133	0.01651
1997 Q2	3.83105	126.98688	0.01077	0.11207	0.01965
1997 Q3	4.56927	180.652	0.01044	0.15866	0.0196
1997 Q4	4.72382	233.68065	0.00917	0.20197	0.01754
1998 Q1	7.21935	351.94111	0.01307	0.29269	0.02503
1998 Q2	7.19875	462.22825	0.01065	0.37588	0.02072
1998 Q3	5.71394	588.2707	0.00754	0.45809	0.01483

Since the graph-based collaborative filtering based recommendation model produces more recommendations for similarity 10%, we can see the increase in the total number of recommendations in comparison to the collaborative filtering based recommendation model in Table 11. However, there is no significant improvement in recall. Therefore, the usage of indirect connections increases the coverage of customers' preferences. However, the usage of indirect connections also produces many incorrect recommendations as well.

Table 5-11 Comparison of Graph-based CF Model to CF model for Similarity 10%

	Number of Recommendations				
Graph Based CF Model	Correct	Total	Precision	Recall	F_Score
1997 Q1	1.88513	75.08630	0.00978	0.05691	0.01669
1997 Q2	2.86345	154.97968	0.00997	0.12185	0.01843
1997 Q3	3.31831	217.53064	0.00916	0.16998	0.01739
1997 Q4	3.61960	271.08325	0.00831	0.21138	0.01599
1998 Q1	6.96257	462.56265	0.01167	0.34701	0.02258
1998 Q2	7.45301	585.31645	0.00948	0.43169	0.01856
1998 Q3	6.03212	678.88290	0.00665	0.47935	0.01313
CF Model	Correct	Total	Precision	Recall	F_Score
1997 Q1	1.88513	74.41650	0.00978	0.05691	0.01669
1997 Q2	2.86345	154.44740	0.00997	0.12185	0.01843
1997 Q3	3.31831	217.07821	0.00916	0.16998	0.01739
1997 Q4	3.61960	270.67878	0.00831	0.21138	0.01599
1998 Q1	6.96257	462.37899	0.01167	0.34701	0.02258
1998 Q2	7.45301	585.20970	0.00948	0.43169	0.01856
1998 Q3	6.03212	678.82027	0.00665	0.47935	0.01313

5.4 EXPERIMENT ANALYSIS AT PRODUCT TYPE LEVEL

We performed another type of analysis and used product types in recommendation models. Product type is a category of products such as bread. Product name description has a brand name of a product and a product type. We discard the brand name from the description of a product and the remaining part of the description is the product type of that product, such as bread, milk, nuts, etc. Since a brand name should not significantly impact the preferences of a customer, we should consider product types instead. For example, it is unlikely that a customer buys a same product with same brand name. Moreover, different customers buy products with different brand names. Whereas, one customer buys a product that has low price, another customer buys different product that has high quality. Due to many variations in the preferences of customers, recommendation techniques become unable to discover the purchasing patterns of customers using product names. However, customers tend to buy same types of products such as milk, eggs, bread etc. The product taxonomy given in chapter 4 shows the different levels of products' categories. We used the last level of product taxonomy to see the impact of grouping products. Similarly, we can perform an analysis using another level of product taxonomy as well. However, one of the main focuses of our research is to find the impact of grouping products.

Firstly, we show the improvements in the recommendation models due to the usage of product types instead of product names. Therefore, we divide our analysis into five steps to analyze our recommendation model, where three steps are similar to last section's analysis steps.

1. In the first step, we analyze the impact of using product type instead of using product name.

2. In the second step, we analyze the impact of integrating different techniques on a recommendation model.
3. In the third step, we analyze the impact of using the demographic information of customers in a recommendation model to handle the cold start problem.
4. In the fourth step, we analyze the impact of influence transfer from indirect connections on a graph-based recommendation model.
5. In the fifth step, we analyze the impact of using PageRank to rank products in recommendation models.

5.4.1 COMPARISON OF THE PRODUCT NAME AND PRODUCT TYPE BASED RECOMMENDATION SYSTEM MODELS

Precision, recall and f_score increase significantly when product types are used in the recommendation system instead of product names as shown in the Tables 12 to 14 below.

Table 5-12 Comparison of Name Based Previously Purchased Model and Type Based Previously Purchased Model

	Precision		***Recall***		***F_Score***	
	Name	**Type**	**Name**	**Type**	**Name**	**Type**
1997 Q1	0.01912	0.04586	0.00438	0.01042	0.00713	0.01538
1997 Q2	0.02238	0.05232	0.00855	0.02050	0.01237	0.02640
1997 Q3	0.02070	0.04949	0.01170	0.03004	0.01495	0.03327
1997 Q4	0.01673	0.04341	0.01457	0.03988	0.01558	0.03665
1998 Q1	0.02321	0.06344	0.02295	0.06220	0.02308	0.05187
1998 Q2	0.01910	0.05296	0.03259	0.08641	0.02408	0.05678
1998 Q3	0.01332	0.03818	0.04175	0.11798	0.02020	0.05185

Table 5-13 Comparison of Name Based Content-Based Model and Type Based Content-Based Model

	Precision		***Recall***		***F_Score***	
	Name	**Type**	**Name**	**Type**	**Name**	**Type**
1997 Q1	0.00825	0.03778	0.04282	0.04269	0.01384	0.03431
1997 Q2	0.00897	0.04191	0.07673	0.07662	0.01605	0.04768
1997 Q3	0.00844	0.03982	0.10852	0.10839	0.01566	0.05239
1997 Q4	0.00745	0.03552	0.13864	0.13854	0.01414	0.05150
1998 Q1	0.01136	0.05344	0.20733	0.20731	0.02155	0.07391
1998 Q2	0.00920	0.04409	0.26910	0.26923	0.01780	0.06991
1998 Q3	0.00646	0.03155	0.33894	0.33899	0.01268	0.05520

Table 5-14 Comparison of Name Based Frequent Product Set Based Model and Type Based Frequent Product Set Based Model

	Precision		*Recall*		*F_Score*	
	Name	**Type**	**Name**	**Type**	**Name**	**Type**
1997 Q1	0.00954	0.03548	0.06133	0.24691	0.01651	0.05826
1997 Q2	0.01077	0.03861	0.11207	0.43901	0.01965	0.06730
1997 Q3	0.01044	0.03684	0.15866	0.53680	0.01960	0.06570
1997 Q4	0.00917	0.03263	0.20197	0.59171	0.01754	0.05926
1998 Q1	0.01307	0.04975	0.29269	0.81451	0.02503	0.08826
1998 Q2	0.01065	0.04139	0.37588	0.89274	0.02072	0.07559
1998 Q3	0.00754	0.02974	0.45809	0.93699	0.01483	0.05620

Table 5-15 Comparison of Name Based Collaborative-Filtering Model and Type Based Collaborative-Filtering Model

	Precision		*Recall*		*F_Score*	
	Name	**Type**	**Name**	**Type**	**Name**	**Type**
1997 Q1	0.00978	0.03544	0.05691	0.30274	0.01669	0.05783
1997 Q2	0.00997	0.03828	0.12185	0.44892	0.01843	0.06608
1997 Q3	0.00916	0.03683	0.16998	0.53355	0.01739	0.06500
1997 Q4	0.00831	0.03281	0.21138	0.58490	0.01599	0.05885
1998 Q1	0.01167	0.04967	0.34701	0.80302	0.02258	0.08764
1998 Q2	0.00948	0.04145	0.43169	0.88474	0.01856	0.07537
1998 Q3	0.00665	0.02970	0.47935	0.92994	0.01313	0.05610

As shown in Table 16, the average number of product types purchased by a customer is almost similar to the average number of products (name). Therefore, customers do not buy many products from a same category of products. Moreover, customers have different preferences for the brand name of products. However, a significant improvement in precision, recall and f_score shows that customers' preferences depend more upon product types instead of product names. Therefore, the preferences of a customer are more correlated to product types, such as milk, egg and many others. However, customers' preferences do not show much correlation with product names. This shows the impact of using a category level, or grouping similar entities in a sparse dataset.

5.4.2 IMPACT OF DIFFERENT TECHNIQUES

As shown in the Table 17, the numbers of correct recommendations are increasing as the numbers of total recommendations are increasing. However, precision is decreasing for all models since the total number of recommendations is increasing relatively higher compared to the total number of correct recommendations. As the number of training data increases, it establishes more relations in data entities regarding customer preferences and increases the number of total recommendations based on discovered preferences. Another common trend that was also observed in previous analysis is the increase in precision for second and fourth case for the same reason discussed in previous analysis.

Table 5-16 Average Number of Purchased Products by A Customer in Each Quarter

Time Period	*Number of Purchased Products (Name)*	*Number of Purchased Products (Type)*
1997 Q1	27.53	25.34
1997 Q2	25.73	23.75
1997 Q3	23.97	22.18
1997 Q4	22.13	20.52
1998 Q1	18.26	17.19
1998 Q2	14.24	13.62
1998 Q3	9.88	9.62

Table 5-17 Number of Correct and Total Recommendations Produced by Each Technique

Time Period	*Previously Purchased*		*CB*		*ARM*		*CF-Purchase*	
	Correct	**Total**	**Correct**	**Total**	**Correct**	**Total**	**Correct**	**Total**
1997 Q1	0.41	2.38	1.76	12.21	8.71	74.14	10.20	93.90
1997 Q2	0.69	4.58	2.65	21.97	11.44	134.64	11.60	139.36
1997 Q3	0.89	6.90	3.17	31.01	11.28	166.07	11.22	165.68
1997 Q4	1.00	9.43	3.31	39.83	10.09	183.53	10.03	181.79
1998 Q1	1.57	14.79	5.17	59.82	15.45	253.21	15.30	249.56
1998 Q2	1.77	20.82	5.25	78.04	12.87	277.60	12.81	275.09
1998 Q3	1.61	29.12	4.31	98.71	9.25	291.41	9.21	289.14

The precision of hybrid models is lower compared to that of the previously purchased model. However, the previously purchased model does not produce sufficient recommendations and recall rate is very low. Content-based, frequent product set mining and collaborative filtering models have much higher recall. Hybrid models produce higher number of recommendations and higher number of incorrect recommendations as well since customers do not purchase all products related to their preferences. Therefore, precision of hybrid models is lower compared to the previously purchased based model, but recall of hybrid models is better since hybrid models tend to satisfy more of customers' preferences.

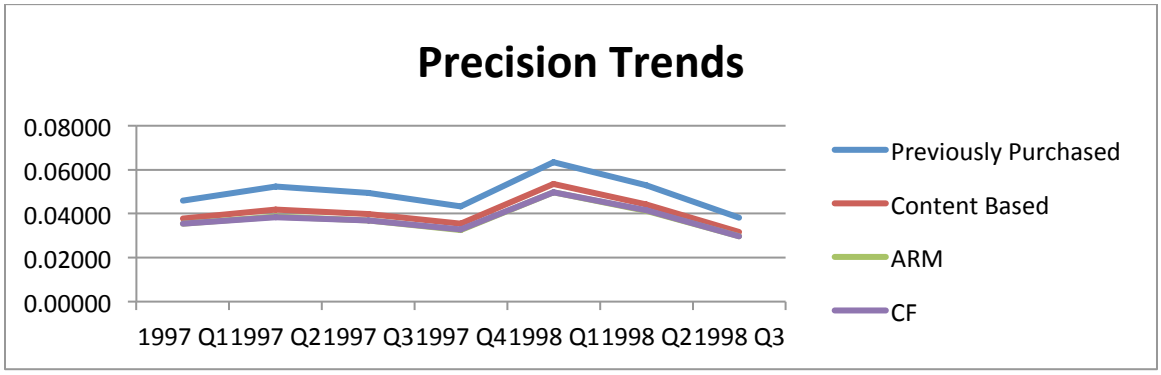

Figure 5-11 Precision Trends in Recommendations Models

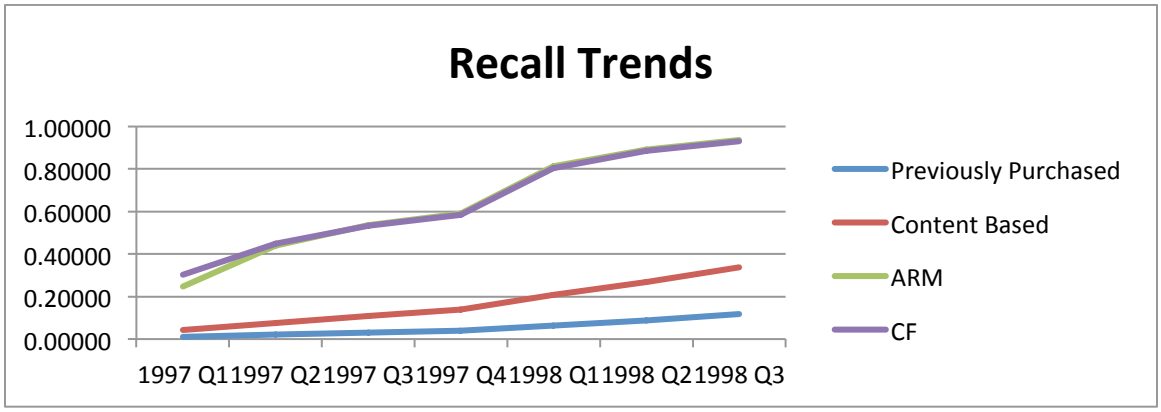

Figure 5-12 Recall Trends in Recommendations Models

Since recommending all products is not a feasible approach, we have to produce a limited set of recommendations related to customer preferences. On the other side, our base model only recommends the previously purchased products and this is also not a feasible approach either. Therefore, hybrid models produce more reasonable recommendations and it is evident through the increase in f_score.

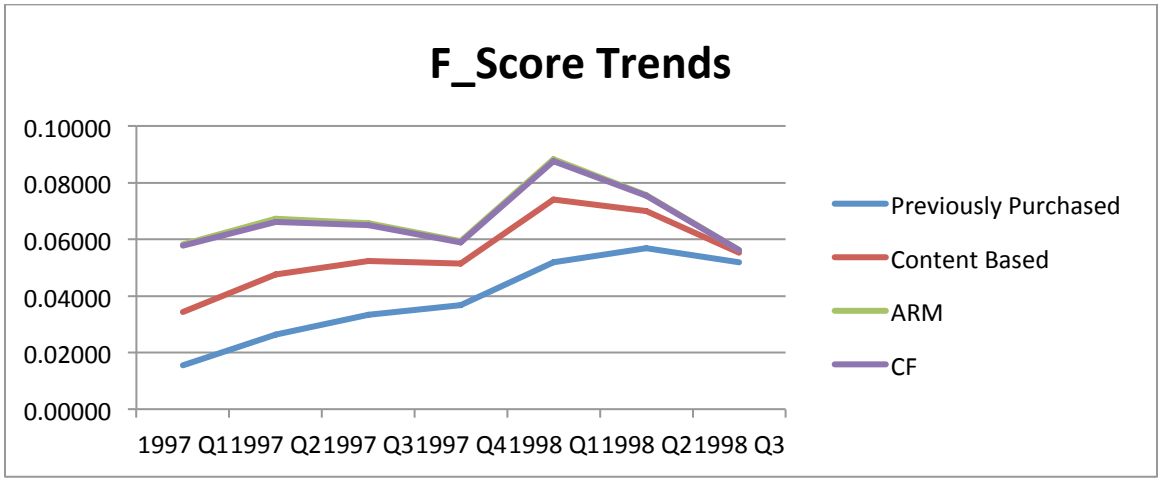

Figure 5-13 F_Score Trends in Recommendations Models

Furthermore, frequent product set mining and collaborative filtering have very high recall compared to other models. This shows that customers tend to buy different products and these products can be explored though association rule mining or collaborative filtering. The f_score of content-based is lower than frequent product set mining and collaborative filtering based models and this suggests that customers buy lower number of similar products in comparison to trending products or products bought by similar customers.

Table 5-18 Precision of different techniques

	Previously Purchased	*Content Based*	*ARM*	*CF*
1997 Q1	**0.04586**	0.03778	0.03548	0.03544
1997 Q2	**0.05232**	0.04191	0.03861	0.03828
1997 Q3	**0.04949**	0.03982	0.03684	0.03683
1997 Q4	**0.04341**	0.03552	0.03263	0.03281
1998 Q1	**0.06344**	0.05344	0.04975	0.04967
1998 Q2	**0.05296**	0.04409	0.04139	0.04145
1998 Q3	**0.03818**	0.03155	0.02974	0.02970

Table 5-19 Recall of different techniques

	Previously Purchased	*Content Based*	*ARM*	*CF*
1997 Q1	0.01042	0.04269	0.24691	**0.30274**
1997 Q2	0.02050	0.07662	0.43901	**0.44892**
1997 Q3	0.03004	0.10839	**0.53680**	0.53355
1997 Q4	0.03988	0.13854	**0.59171**	0.58490
1998 Q1	0.06220	0.20731	**0.81451**	0.80302
1998 Q2	0.08641	0.26923	**0.89274**	0.88474
1998 Q3	0.11798	0.33899	**0.93699**	0.92994

Table 5-20 F_Score (F1) of different techniques

	Previously Purchased	*Content Based*	*ARM*	*CF*
1997 Q1	0.01538	0.03431	**0.05826**	0.05783
1997 Q2	0.02640	0.04768	**0.06730**	0.06608
1997 Q3	0.03327	0.05239	**0.06570**	0.06500
1997 Q4	0.03665	0.05150	**0.05926**	0.05885
1998 Q1	0.05187	0.07391	**0.08826**	0.08764
1998 Q2	0.05678	0.06991	**0.07559**	0.07537
1998 Q3	0.05185	0.05520	**0.05620**	0.05610

5.4.2.1 Impact of Support In the Frequent Product-Set Mining Based Model

We investigated the impact of different support count values on the accuracy of a recommendation model. As explained in data characteristics, the support count values of frequent product type set vary from 2 to 45. Approximately 65% of frequent product type sets have support value larger than 8. Precision at different support count values is shown in Figure 14 below. Initially, precision increases when support count value increases. However, the difference between the precision decreases when data in training increases and occurrences of frequent product typesets increases as well. Therefore, most of the association rules with support count 2 also have support count 8 as well and the recommendation model produces almost the same number of recommendations at each support values. For same reason, recall is high at lower supports count and the difference between the recall values at different support count values keeps decreasing.

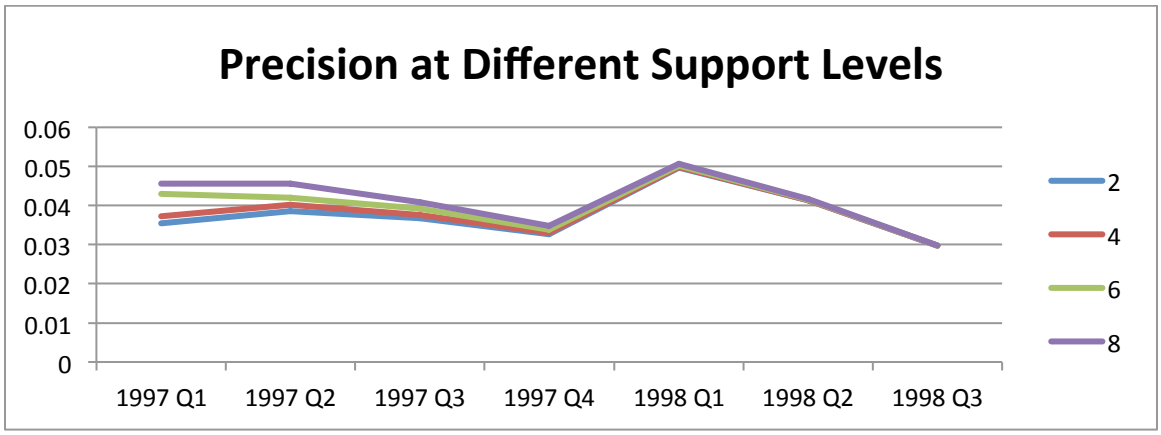

Figure 5-14 Precision at Different Support Counts

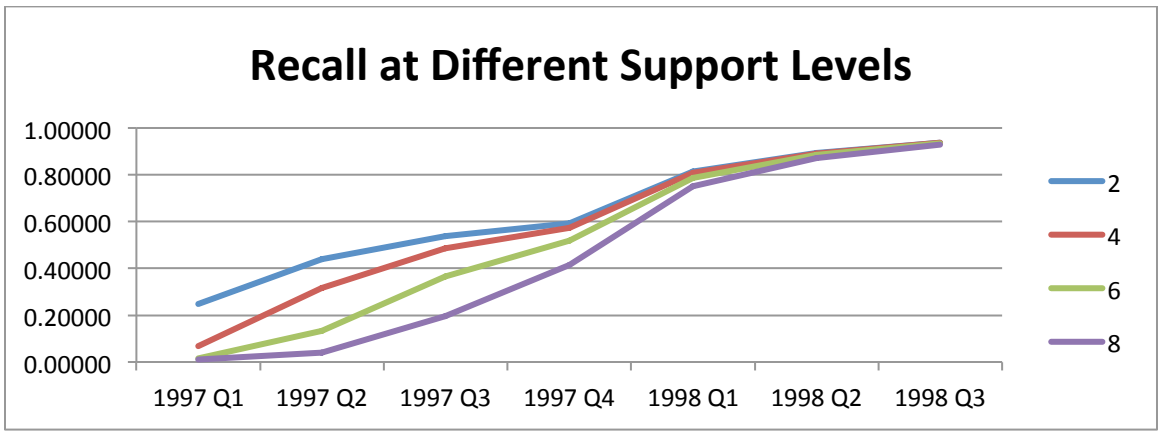

Figure 5-15 Recall at Different Support Counts

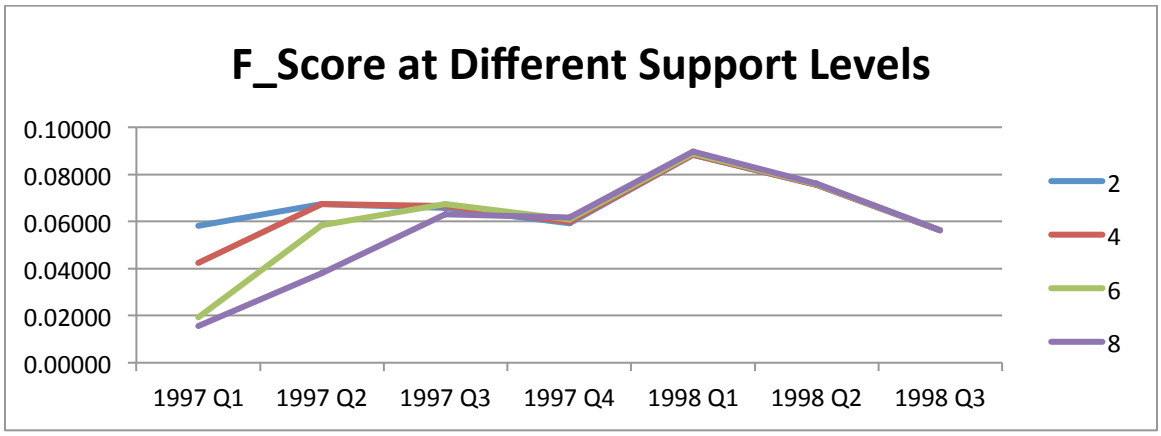

Figure 5-16 F_Score at Different Support Counts

There is another trend to notice is that support 2 has the highest f_score for first case, support 4 has the highest f_score for second case, support 6 has the highest f_score for third case as shown in table 21. Therefore, when we increase the size of training dataset, it extracts more number of association rules with higher support values. Therefore, higher support count is performing better for denser data. Similarly, lower support performs better for sparse data. Therefore, the support value can significantly improve the accuracy of recommendation models if the correct support count value is used to establish association rules.

Table 5-21 F_Score at different Support Counts

Time Period	***2***	***4***	***6***	***8***
1997 Q1	**0.05826**	0.04229	0.01927	0.01558
1997 Q2	0.06730	**0.06752**	0.05843	0.03814
1997 Q3	0.06570	0.06669	**0.06746**	0.06302
1997 Q4	0.05926	0.05976	0.06089	**0.06172**
1998 Q1	0.08826	0.08841	0.08878	**0.08955**
1998 Q2	0.07559	0.07563	0.07574	**0.07603**
1998 Q3	0.05620	0.05621	0.05624	**0.05628**

5.4.2.2 Impact of Different Similarities in Collaborative Filtering Model

Similar to the analysis of similarity at product name level, precision of the recommendation model increases with increase in similarity value. Additionally, there is a significant fall in the recall value from similarity 20% to similarity 30%. Therefore, similarity 20% has highest f_score for most of the cases except first two cases. The similarity 10% has the highest f_score for the first two cases.

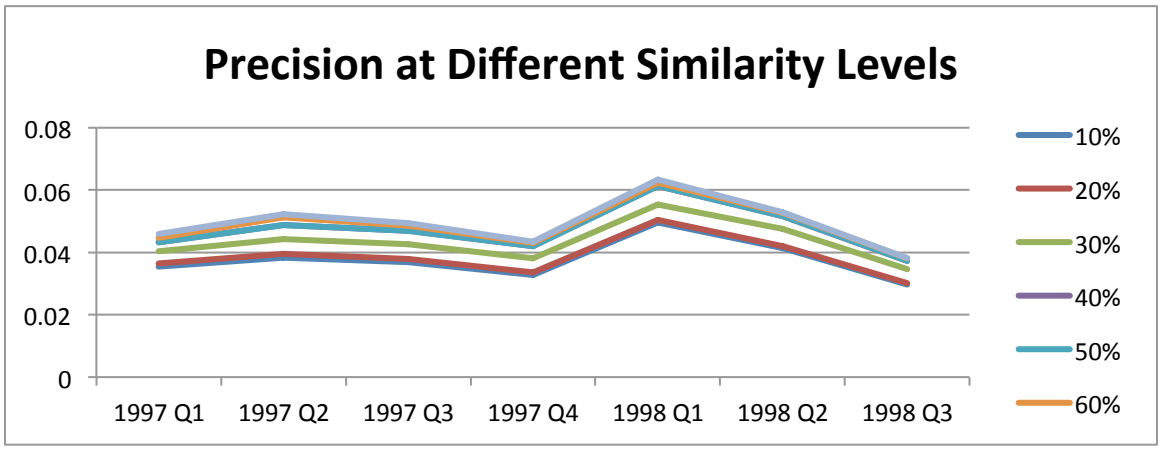

Figure 5-17 Precision at Different Similarity Levels

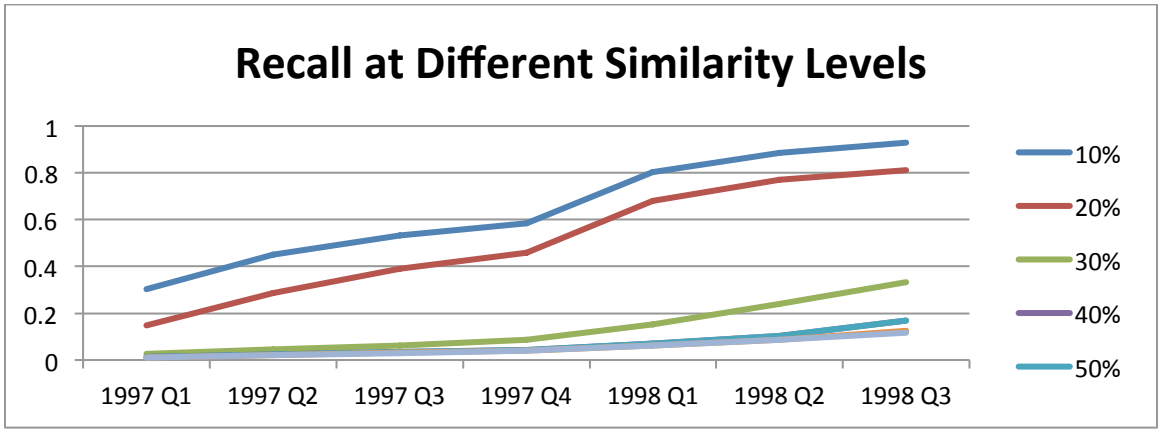

Figure 5-18 Recall at Different Similarity Levels

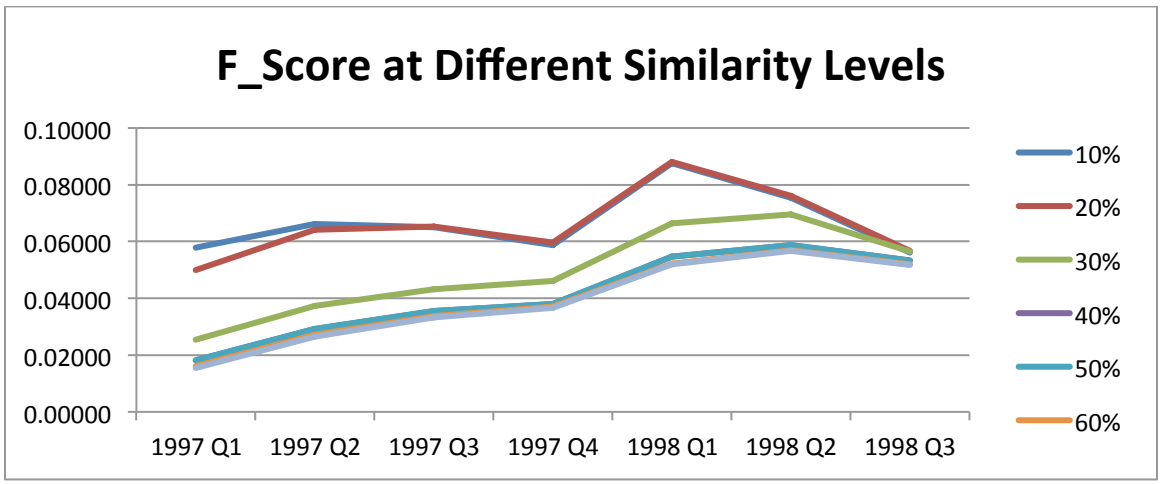

Figure 5-19 F_Score at Different Similarity Levels

The results show that similarity impacts the accuracy of recommendation systems. The level of similarity to find similar customers depends on data. If customers tend to buy many similar products on average, higher similarity performs better to establish relations. However, if customers buy few similar products, we still want to establish relations between customers to discover their personal preferences in a sparse dataset using a lower similarity level.

Table 5-22 F_Score (F1) at Different Similarity Levels

Time Period	*10%*	*20%*	*30%*	*40%*	*50%*	*60%*	*70%*
1997 Q1	**0.05783**	0.04992	0.02542	0.01807	0.01807	0.01602	0.01539
1997 Q2	**0.06608**	0.06424	0.03735	0.02914	0.02914	0.02710	0.02640
1997 Q3	0.06500	**0.06529**	0.04329	0.03551	0.03551	0.03380	0.03327
1997 Q4	0.05885	**0.05958**	0.04607	0.03798	0.03798	0.03699	0.03665
1998 Q1	0.08764	**0.08794**	0.06639	0.05475	0.05475	0.05218	0.05187
1998 Q2	0.07537	**0.07601**	0.06959	0.05878	0.05878	0.05702	0.05678
1998 Q3	0.05610	**0.05666**	0.05654	0.05323	0.05323	0.05206	0.05185

5.4.2.3 Impact of including the Quantity based Customer Similarity in Collaborative Filtering Model

Quantity does not significantly impact the accuracy of the recommendation models as shown in table 23 for similarity 10% and 20%. The quantity based recommendation model has better precision since it establish stronger relations between customers. However, recall decreases with the integration of quantity based similarity. For similarity 10%, the usage of quantity based similarity decreases f_score. However, the integration of quantity based similarity increases f_score for similarity 20%.

Table 5-23 Comparison of CF Model and CF Quantity based Similarity Model for Similarity 10% and 20%

	Recall			*Precision*			*F_Score*		
Similarity 1	**CF (1)**	**CF With Quantity (2)**	**(1 - 2)**	**CF (1)**	**CF With Quantity (2)**	**(1 - 2)**	**CF (1)**	**CF With Quantity (2)**	**(1 - 2)**
1997 Q1	0.30274	0.29072	0.01202	0.03544	0.03547	-0.00003	0.05783	0.05763	0.00020
1997 Q2	0.44892	0.44053	0.00839	0.03828	0.03834	-0.00006	0.06608	0.06610	**-0.00002**
1997 Q3	0.53355	0.52694	0.00661	0.03683	0.03686	-0.00003	0.06500	0.06502	**-0.00002**
1997 Q4	0.58490	0.57979	0.00511	0.03281	0.03284	-0.00003	0.05885	0.05889	**-0.00004**
1998 Q1	0.80302	0.79787	0.00515	0.04967	0.04968	-0.00001	0.08764	0.08765	**-0.00001**
1998 Q2	0.88474	0.87882	0.00592	0.04145	0.04144	**0.00001**	0.07537	0.07535	0.00002
1998 Q3	0.92994	0.92265	0.00729	0.02970	0.02971	-0.00001	0.05610	0.05612	**-0.00002**
Similarity 2									
1997 Q1	0.14798	0.09392	0.05406	0.03654	0.03723	-0.00069	0.04992	0.04312	0.00680
1997 Q2	0.28680	0.17551	0.11129	0.03957	0.04027	-0.00070	0.06424	0.05733	0.00691
1997 Q3	0.39094	0.25053	0.14041	0.03786	0.03884	-0.00098	0.06529	0.06234	0.00295
1997 Q4	0.45959	0.30791	0.15168	0.03362	0.03440	-0.00078	0.05958	0.05841	0.00117
1998 Q1	0.68109	0.49993	0.18116	0.05036	0.05114	-0.00078	0.08794	0.08624	0.00170
1998 Q2	0.77058	0.59562	0.17496	0.04202	0.04297	-0.00095	0.07601	0.07629	**-0.00028**
1998 Q3	0.81136	0.64623	0.16513	0.03015	0.03065	-0.00050	0.05666	0.05683	**-0.00017**

Customers buy different products and the quantity of products does not play a significant role to establish connections between two customers. If two customers buying similar products, the number of similar products is more important than the quantity of those products when

quantity does not deviate at significant level. As shown in data characteristics chapter, the highest number of times a customer buys a same product type is 4. However, the integration of quantity can significantly improve the accuracy of recommendation system if customers buy many similar products and the strength of a relation based on quantity can distinguish customers that are more similar to a target customer.

5.4.3 IMPACT OF USING DEMOGRAPHIC INFORMATION

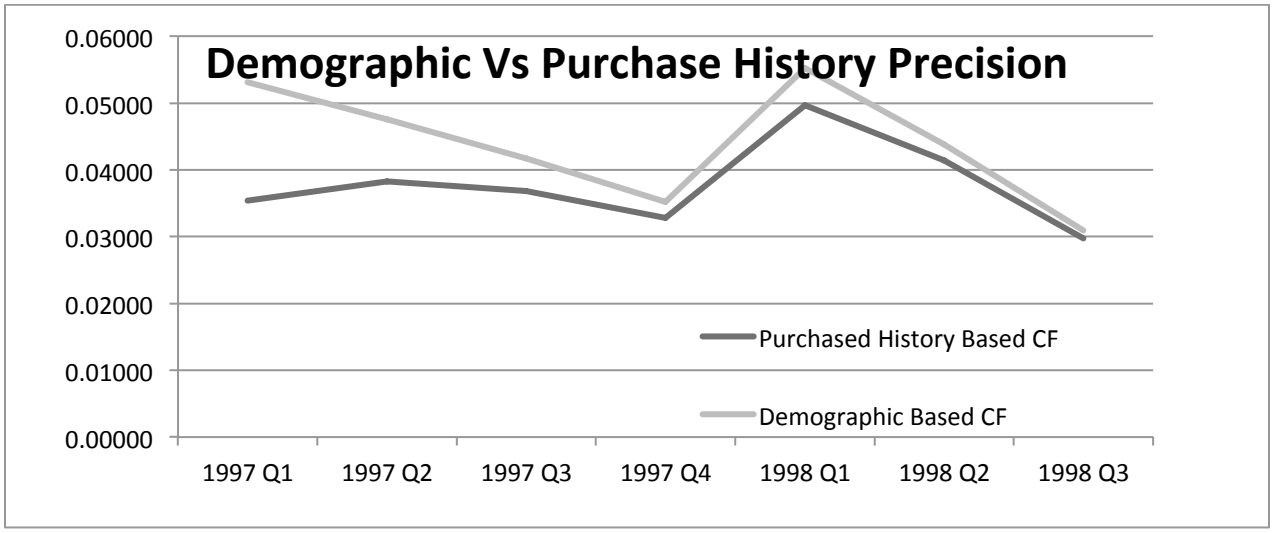

Figure 5-20 Demographic Vs. Purchase History Precision

The demographic properties based collaborative filtering model produced better precision and recall in comparison to our collaborative filtering model. Therefore, a recommendation model predicts purchasing patterns of a customer based on demographic properties of that customer. The demographic based collaborative filtering model have better precision for all the cases and it shows that demographic properties can also distinguish the existing customers into the groups with similar purchasing patterns.

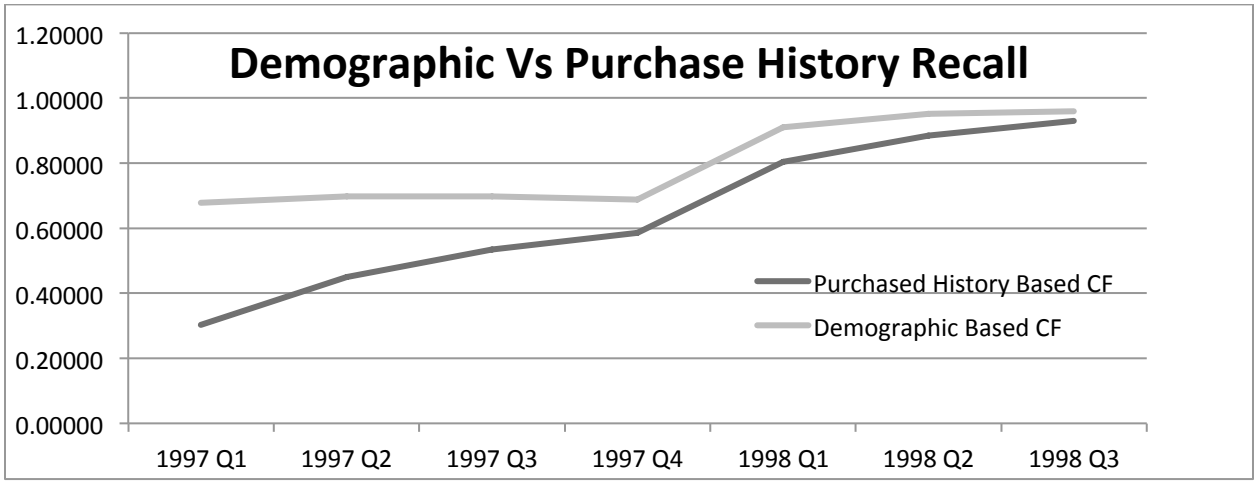

Figure 5-21 Demographic Vs. Purchase History Recall

Unlike other models, precision of demographic recommender model decreases from very first case. First quarter of the 1997 introduces many new customers and our recommendation model able to make the connections between customers based on demographic properties. This factor supports our argument that our collaborative filtering model can predict for new customers using demographic properties and the usage of demographic properties increases the accuracy of collaborative recommendation systems.

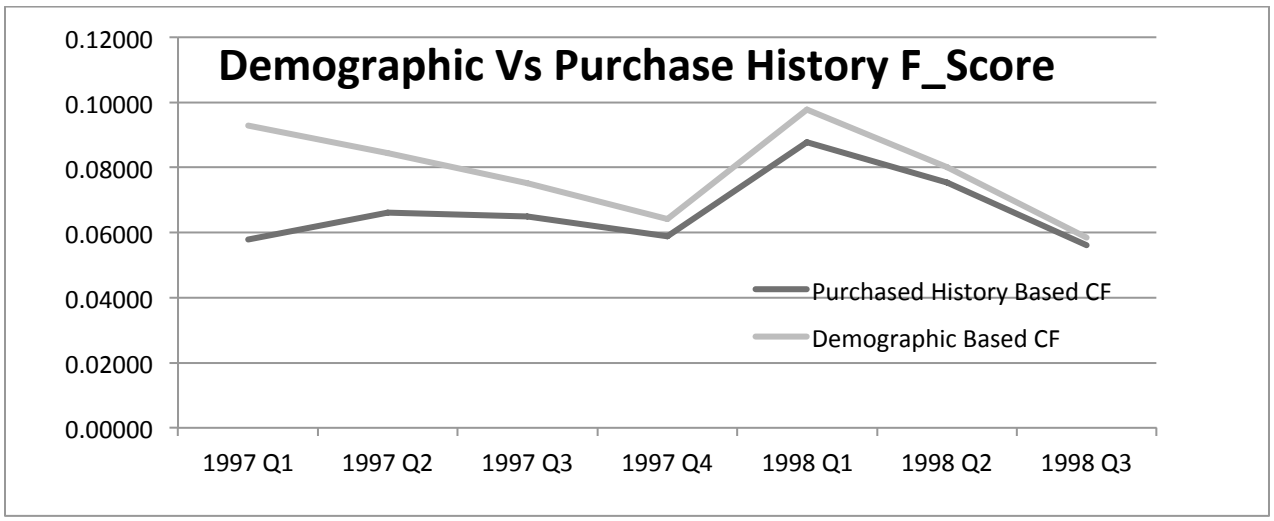

Figure 5-22 Demographic Vs. Purchase History F_Score

5.4.4 IMPACT OF INDIRECT CONNECTIONS

To analysis the impact of indirect influence, we compared the ARM model to the graph-based ARM model and CF model to graph-based CF model where graph-based model also uses indirect connections to generate a list of recommendations. There are no significant improvements in the accuracy of recommendation models as we can see in the Tables 24, 25, 26, and 27. The usage of indirect connections increases the coverage of customers' preferences because it increases the total number of recommendations. However, customers do not necessarily purchase all the recommendations generated through indirect connections since customers usually buy required and a limited set of products.

Table 5-24 Comparison of correct and total number of recommendations in Graph based CF and CF at Similarity 60%

Time Period	*# Of Correct Recommendations*		*# Of Total Recommendations*	
	Graph Based CF	CF	Graph Based CF	CF
1997 Q1	0.4101	0.42371	3.05509	2.54703
1997 Q2	0.69136	0.70437	5.12355	4.78867
1997 Q3	0.89473	0.90367	7.36266	7.08941
1997 Q4	1.00509	1.00993	9.8438	9.57519
1998 Q1	1.56909	1.5754	14.98055	14.97719
1998 Q2	1.77099	1.81782	20.92974	21.16286
1998 Q3	1.61899	1.84241	29.22821	31.52563

The usage of indirect connections in our collaborative filtering model increases the total number of recommendations and recall value as shown in the Table 24 and 25. However, the precision and f_score of our collaborative filtering model decreases since the increase in total number of recommendation is higher in comparison to the increase in correct number of recommendations.

Table 5-25 Comparison of recommendations in Graph based CF and CF Similarity 60%

Time Period	*Precision*		*Recall*		*F_Score*	
	Graph Based CF	CF	Graph Based CF	CF	Graph Based CF	CF
1997 Q1	0.04576	0.04468	0.01046	0.01104	0.01542	0.01602
1997 Q2	0.05226	0.05116	0.02054	0.02131	0.02643	0.0271
1997 Q3	0.04943	0.04858	0.03008	0.03075	0.03329	0.0338
1997 Q4	0.04343	0.04299	0.0399	0.04032	0.03667	0.03699
1998 Q1	0.06337	0.0624	0.06229	0.06286	0.05189	0.05218
1998 Q2	0.05293	0.05269	0.08641	0.08767	0.05677	0.05702
1998 Q3	0.03818	0.03805	0.11814	0.12537	0.05186	0.05206

The usage of indirect connections in our frequent product set mining model increases total number of recommendations and recall as shown in the Table 26 and 27. However, the precision and f_score are decreasing through integrating the indirect connections.

Table 5-26 Comparison of correct and total number of recommendations in Graph based ARM and ARM at Support Count 2

Time Period	*# Of Correct Recommendations*		*# Of Total Recommendations*	
	Graph Based ARM	ARM	Graph Based ARM	ARM
1997 Q1	10.79701	8.70579	103.32929	74.14242
1997 Q2	11.89526	11.44194	145.95692	134.63551
1997 Q3	11.41214	11.27933	170.70899	166.06819
1997 Q4	10.13002	10.09342	185.5768	183.53176
1998 Q1	15.46673	15.44701	254.02375	253.2066
1998 Q2	12.87029	12.8674	277.85964	277.59945
1998 Q3	9.24915	9.24815	291.52045	291.40973

Table 5-27 Comparison of Graph based ARM and ARM at Support Count 2

Time Period	*Precision*		*Recall*		*F_Score*	
	Graph Based ARM	ARM	Graph Based ARM	ARM	Graph Based ARM	ARM
1997 Q1	0.03472	0.03548	0.33009	0.24691	0.05919	0.05826
1997 Q2	0.03825	0.03861	0.4676	0.43901	0.06694	0.0673
1997 Q3	0.03669	0.03684	0.54745	0.5368	0.0655	0.0657
1997 Q4	0.03257	0.03263	0.59541	0.59171	0.05917	0.05926
1998 Q1	0.04973	0.04975	0.81621	0.81451	0.08824	0.08826
1998 Q2	0.04138	0.04139	0.8931	0.89274	0.07558	0.07559
1998 Q3	0.02974	0.02974	0.93716	0.93699	0.0562	0.0562

5.4.5 ENSEMBLE MODEL ANALYSIS

We compared our ensemble model to other models. As given in table 28, the ensemble approach recommends all products' types. The ensemble approach has highest recall and f_score. The precision of ensemble model is lower than previously purchased model. However, the previously purchased model has the lowest recall and f_score. The demographic similarity based model has shown the performance to be very close to that of the ensemble model. Although, the ensemble approach has shown the better performance compared to other models, recommending all products to a customer is not a feasible option.

Table 5-28 Comparison of ensemble approach to other models

Model	*# Of Correct Recommendation*	*# Of Total Recommendation*	*Precision*	*Recall*	*F_Score*
Purchased	1.56882	14.78776	0.06344	0.06220	0.05187
CB	5.17346	59.81701	0.05344	0.20731	0.07391
ARM	15.44701	253.20660	0.04975	0.81451	0.08826
CF - PH	15.29836	249.55554	0.04967	0.80302	0.08764
CF - Demo	16.06010	283.15468	0.05526	0.90995	0.09786
Graph based ARM	15.46673	254.02375	0.04973	0.81621	0.08824
Graph Based CF	15.29836	249.73920	0.04967	0.80302	0.08764
Ensemble	17.18728	311.00000	0.05526	1.00000	0.09881

However, the ensemble approach has a major benefit compared to recommending all products, the ensemble approach produces recommendations based on discovered customers' preferences. Therefore, the ensemble approach is able to rank products based on the relevance of a product to the preferences of a customer. Ranking products based on demographic properties in our demographic based model is not possible. Demographic based model does not require indirect connection and there is no criterion to rank products. Similarly, previously purchased model can't rank products since customers usually do not buy same products. Moreover, the

content-based model did not show better results and it uncovers fewer customers' preferences compared to other models. Moreover, the frequent product set mining model only produces popular and trending products. Therefore, everyone will get similar recommendation list using the frequent product set mining recommendation technique. Furthermore, the collaborative filtering model based on purchase history can rank products but this will result in ignoring the other factors such as trending or popular products, products from similar customer based on demographic properties and similar products to the previously purchased products. Therefore, we analyze the impact of PageRank based ranking on our ensemble approach model.

5.4.6 PAGERANK BASED RANKING ANALYSIS

Since recommending all products creates the information overload problem, we need a ranking procedure to limit the number of recommendations. PageRank based ranking uses the influence transfer from indirect connections. We recommended a certain percentage of total number of recommendations and compared the results. As shown in table 59, the total number of products is 311 when we recommend all, and 5% of total number of products is 15 for first case. As we can see, percentage change in total number of correct products is correlated to the percentage change in the total number of recommendations. Therefore, if we limit the total number of recommendations, it will limit the number of correct recommendation and precision will stay almost same as shown in table 30. Therefore, we can limit the number of total recommendations according to our requirements regarding the total coverage of customers' preferences.

Table 5-29 Change in the correct and total number of recommendations in ranked recommendations list

Percentage of the Total Recommendations	*# Of Correct Recommendation*	*% Change in the # of Correct Recommendations*	*# Of Total Recommendation*	*% Change in the # of Total Recommendations*
5%	0.87805		15.00000	
10%	1.76402	101%	31.00000	107%
15%	2.60343	48%	46.00000	48%
20%	3.49852	34%	62.00000	35%
25%	4.31138	23%	77.00000	24%
30%	5.20258	21%	93.00000	21%
35%	6.03461	16%	108.00000	16%
40%	6.92407	15%	124.00000	15%
45%	7.76778	12%	139.00000	12%
50%	8.64690	11%	155.00000	12%
55%	9.51704	10%	171.00000	10%
60%	10.34720	9%	186.00000	9%
65%	11.22176	8%	202.00000	9%
70%	12.04320	7%	217.00000	7%
75%	12.93561	7%	233.00000	7%

80%	13.74376	6%	248.00000	6%
85%	14.61645	6%	264.00000	6%
90%	15.43413	6%	279.00000	6%
95%	16.31473	6%	295.00000	6%
100%	17.18728	5%	311.00000	5%

As shown in Table 29 and 30, the increase in correct number of recommendations is 101%, in recall is 97% and in f_score is 35.8% when we change the recommendations from 5% to 10% and it increased the total number of recommendations by 107%. We can preform a simple cost benefit analysis in which percentage change in total number of recommendations is the cost and percentage change in correct number of recommendation, precision, recall or f_score is the benefit. 5% turns out be the best case. Therefore, we are not getting same percentage change in correct number of recommendation, recall, and f_score and this indicates the marginal benefit of increasing total number of recommendation is less in comparison to the marginal cost which is the percentage change in total number of recommendations. Similarly, the increase in the total number of recommendations does not produce the same marginal benefit in the other cases, such as from 10% to 15%, 15% to 20% and so on. Additionally, the change in the precision is not as significant as the change in the recall and f_score. Precision is decreasing when we increase the total number of recommendations, which shows that the increase in the number of correct recommendations are relatively less in comparison to the increase in total number of recommendations. Moreover, we can customize the cost benefit function according to our own requirements. Therefore, constraining the number of recommendation based on ranked list produce better results.

Table 5-30 Change in the Precision, Recall and F_score

Percentage of the Total Recommendations	*Precision*	*% Change in Precision*	*Recall*	*% Change in Recall*	*F_Score*	*% Change in F_Score*
5%	0.05854		0.05129		0.04394	
10%	0.05690	-2.80%	0.10117	97%	0.05967	35.8%
15%	0.05660	-0.53%	0.14931	48%	0.06852	14.8%
20%	0.05643	-0.30%	0.20195	35%	0.07499	9.4%
25%	0.05599	-0.78%	0.24877	23%	0.07889	5.2%
30%	0.05594	-0.09%	0.29906	20%	0.08252	4.6%
35%	0.05588	-0.11%	0.34617	16%	0.08515	3.2%
40%	0.05584	-0.07%	0.39854	15%	0.08751	2.8%
45%	0.05588	0.07%	0.44732	12%	0.08949	2.3%
50%	0.05579	-0.16%	0.50136	12%	0.09105	1.7%
55%	0.05566	-0.23%	0.55159	10%	0.09228	1.4%
60%	0.05563	-0.05%	0.60011	9%	0.09343	1.2%
65%	0.05555	-0.14%	0.65092	8%	0.09441	1.0%
70%	0.05550	-0.09%	0.69906	7%	0.09521	0.8%
75%	0.05552	0.04%	0.75131	7%	0.09611	0.9%
80%	0.05542	-0.18%	0.79810	6%	0.09667	0.6%
85%	0.05537	-0.09%	0.84918	6%	0.09728	0.6%
90%	0.05532	-0.09%	0.89669	6%	0.09778	0.5%
95%	0.05530	-0.04%	0.94922	6%	0.09834	0.6%
100%	0.05526	-0.07%	1.00000	5%	0.09881	0.5%

Chapter Six: **Conclusion and Future Work**

6.1 CONCLUSION AND SUMMARY

Two main challenges faced by recommender systems are the integration of different types of data sources and the integration of recommendation techniques. Since customer preferences are available in various sources, combining data sources and establishing relations between data entities needs a structural approach. We used a graph based recommender system to combine three types of available data: product categorical information, customer demographic information and transactional information. Another challenge is combining the various types of recommendation techniques to derive relations between customers and their preferences. We used a comprehensive approach to integrate content-based, collaborative filtering and frequent product set mining techniques in our recommender model. Additionally, we integrated the indirect influence transfer to rank entities in our recommender model. Recommendation system predicts recommendations based on the influential value of entities. In order to analyze our recommender model, we analyzed three main aspects of our recommender system.

Firstly, we found the impact of using different type of data sources through using the different recommendation techniques such as content-based using the product categorical information, collaborative filtering based on purchase history using customers information and transactions, and frequent product mining using transactions. Moreover, we analysis the impact of using product type instead of using product name in order to find the impact of grouping products or by using different level in a product taxonomy. Another analysis was performed to see the impact of using demographic information of customers in a collaborative filtering approach. The three main discoveries from our analysis are summarized below.

The first trend in our dataset is the highest accuracy of frequent product set mining technique. Therefore, the preferences of customers are more explained through trending or popular products. On the other side, customers do not tend to buy similar products since the accuracy of content based is less compared to the collaborative filtering and frequent product set mining approach. This shows the importance of different data sources to predict customer preferences.

The second trend is the increase in accuracy of our recommender system when we establish customers' preferences using product types instead of product names. Product type is a group of product names. Our results showed that there are not many patterns using brand names and the customer preferences are not apparent. However, the preferences of customers are more apparent using product type since customers tends to buy a particular product type such as milk, bread, or egg instead of a particular brand named product.

The third trend to show the impact of different types of data source is the usage of demographic properties of customers in a collaborative filtering approach. The recommender model based on demographic properties showed much higher accuracy since there are many new customers entering in each time periods. Moreover, the increase in accuracy of integrating the demographic information for new customers and purchase history information for existing customers showed the importance of using the different types of data sources.

Secondly, we found the impact of using various recommendation techniques to predict customers' preferences patterns. For example, the content-based approach shows if the customer tends to buy similar types of products, collaborative filtering shows if similar customers buy similar products, and frequent product set mining shows if customers buy trending or popular products. We compared the accuracy of different techniques and the ensemble approach.

Moreover, we analyzed the impact of different support count values in the frequent product set mining techniques to show the importance of the support count values using dense or sparse data. Similarly, we showed the impact of using different similarities in a collaborative filtering model based on purchase history. Furthermore, we compared the quantity based similarity measure to the basic similarity measurement to show the impact of different types of similarity measurements on a recommender model. The main four discoveries from our analysis are summarized below.

The first observation is the highest accuracy of ensemble approach compared to other recommendation models. This showed that the preferences of a customer depend on all the three techniques. As explained above, the frequent product set mining showed the best performance. However, the frequent product set mining does not cover all customer preferences since some of customer preferences depend on collaborative filtering or content based models.

The second observation is the better accuracy of higher support count for dense data and the better accuracy of lower support for sparse data. Our analysis showed that the initial periods with few transactions to establish associations between products, shows better correlation with lower value of support count on average. The sparse data have less number of transactions and it leads to lower value of support count. On the other side, higher number of transactions led to higher support count value for the association rules on average.

Similar to support analysis, the different similarities showed the different accuracy. The 10% similarity showed better accuracy for initial periods when data is very sparse. However, the similarity 20% showed the best results for rest of the time. However, the 20% similarity at the initial periods is too high to show similarities between the customer's purchasing patterns.

The fourth observation is the impact of using the quantity of similar products to find similar customers. Where general similarity function only considers the number of similar products purchased by two customers, the quantity-based similarity considers the quantity of similar products as well. The integration of quantity does not lead to significant improvement in the accuracy of the recommendation model. However, if a dataset has customers who buys very similar products, the integration of quantity leads to better presentation of similar customer since it can distinguish customers more discretely.

Thirdly, we analyzed the impact of indirect influence on the recommender model. Firstly, we compared the frequent product set mining model to graph based frequent product set mining model, which considers the indirect connections. Similarly, we compared the collaborative filtering model to graph based collaborative model. The graph based model uses the indirect connections. Thirdly, we applied the PageRank on our comprehensive graph based ensemble recommender model to include the indirect influences. Three main observations from integrating indirect influence are summarized below.

First observation is increase in recall of the graph based frequent product set model compared to the frequent product set based model. However, the indirect influence decreases the precision at higher rate, and it decreases f_score value as well. Therefore, the indirect influence increases the coverage of customer preferences but leads to a large number of recommendations that decreases the precision since customers do not buy all products related to their preferences.

Similar to the first observation, the recall rate of graph based collaborative filtering model is higher compared to collaborative filtering model. Similar to the impact on the frequent product set model, the indirect influence decreases precision at higher rate compared to recall and decreases f_score value as well. Therefore, the indirect influence discovers more of the customer

preferences compared to general model but also leads to higher number of recommendations and it generate the information overload problem.

To handle the information overload problem in a graph-based solution, we used PageRank to rank. The recommendation sets are produced based on the first 5, 10, 15 … % of total number of recommendations. The increase in size of a set of recommendations decreases the comparative rate of producing correct recommendations. The results suggest the correctness of integrating the PageRank algorithm to rank list of recommendations according to customer preferences.

6.2 FUTURE WORK

The scope of graph based recommender system can be expanded in many ways. Our proposed model used only three different types of data and three recommendation techniques. Since our graph based model is very flexible, the recommender model can improve through integrating other data sources or recommendation techniques as well. Moreover, many other types of influence transfer algorithm can be used to compare the performance against PageRank. PageRank algorithm can be used on the customer sub graph too, which will further modify the influence value of customers and products according to their relativity to a target customer.

Additionally, we used very simple similarity measurement for customers and products, and it can be improved in many other ways. As we establish the fact that the usage of higher-level product taxonomy can perform better for the sparse data, this can be further enhanced to define a similarity value based on the level of product taxonomy. Defining a similarity measurement based on the level of product taxonomy, will give better weight to links between products. Moreover, the demographic and purchase history based similarity can be combined for customers who do not have sufficient number of transaction to establish their preferences.

Therefore, the demographic properties of customers not only are able to define the preferences of new customers but can also be used for existing customers.

Another way of modify the existing recommender model is to define the significance of each recommendation techniques in a recommendation mode. The recommender model should able to learn which technique performs better compared to other techniques. Each technique should get the weight according to the accuracy of that technique. This addition will make recommender system model self-learning and it improves the model according to the changes in data sources.

Bibliography

1. Kangning Wei, Jinghua Huang, Shaohong Fu. A survey of e-commerce recommender systems. In: IEEE international conference on service systems and service management, Chengdu, China, 2007

2. Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. Recommender Systems Handbook (1st ed.). Springer-Verlag New York, Inc., New York, NY, USA. 2010.

3. Guibing Guo. Improving the performance of recommender systems by alleviating data sparsity and cold start problems. In Proceedings of the Twenty-Third international joint conference on Artificial Intelligence (IJCAI '13), Francesca Rossi (Ed.). AAAI Press 3217-3218. 2013.

4. Miquel Montaner, Beatriz Lopez, and Josep Lluis De La Rosa. A taxonomy of recommender agents on the internet. Artificial Intelligence Review, 19, 285–330. 2003.

5. Michael J. Pazzani. A Framework for Collaborative, Content-Based and Demographic Filtering. Artificial Intelligence Review, 13, 393-408. 1999.

6. Ian M. Soboroff and Charles K. Nicholas, "Combining Content and Collaboration in Text Filtering," Proc. Int'l Joint Conf. Artificial Intelligence Workshop: Machine Learning for Information Filtering, Aug. 1999.

7. Daniel Billsus and Micheal Pazzani, "User Modeling for Adaptive News Access," User Modeling and User-Adapted Interaction, vol. 10, pp. 147-180, 2000.

8. Byeong Man Kim, Qing Li, Chang Seok Park, Si Gwan Kim, and Ju Yeon Kim. 2006. A new approach for combining content-based and collaborative filters. J. Intell. Inf. Syst. 27, 79-91, 2006.

9. J. Ben Schafer, Dan Frankowski, Jon Herlocker, Shilad Sen. Collaborative filtering recommender systems. In: The Adaptive Web, pp. 291–324. Springer Berlin / Heidelberg. 2007.

10. Gediminas Adomavicius and Alexander Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. on Knowl. and Data Eng. 17, 734-749. 2005.

11. Joseph Pine. Mass Customization: The New Frontier in Business Competition. : Harvard Business School Press, 1993.

12. J. Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In Proceedings of the 1st ACM conference on Electronic commerce (EC '99). ACM, New

York, NY, USA, 158-166. 1999.

13. John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98), pp 43-52. 1998.

14. Lingling Zhang; Xiaojie Zhang; Quan Chen; Zhengxiang Zhu; Yong Shi, "Domain-Knowledge Driven Recommendation Method and Its Application," Computational Sciences and Optimization (CSO), 2011 Fourth International Joint Conference on, pp 21-25, 2011.

15. Liu Guo-rong; Zhang Xi-zheng, "Collaborative Filtering Based Recommendation System for Product Bundling," Management Science and Engineering, ICMSE '06. 2006 International Conference on, pp 251-254, 2006.

16. Long-Sheng Chen, Fei-Hao Hsu, Mu-Chen Chen, Yuan-Chia Hsu, Developing recommender systems with the consideration of product profitability for sellers, Information Sciences, Volume 178, Issue 4, 1032-1048,15 February 2008.

17. Cheng-Lung Huang, Wei-Liang Huang, Handling sequential pattern decay: Developing a two-stage collaborative recommender system, Electronic Commerce Research and Applications, Volume 8, Issue 3, 117-129, May–June 2009.

18. Jebrin AL-SHARAWNEH, Mary-Anne WILLIAMS, Credibility-aware Web-based Social Network Recommender: Follow the Leader, ACM RecSys 2010 Workshop on Recommender systems and Social Web. 2010.

19. Jianming He. A Social Network-Based Recommender System. Ph.D. Dissertation. University of California at Los Angeles, Los Angeles, CA, USA. Advisor(s) Wesley W. Chu. 2010.

20. Zan Huang, Wingyan Chung, Thian-Huat Ong, and Hsinchun Chen. A graph-based recommender system for digital library. In Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries (JCDL '02). ACM, New York, NY, USA, 65-73. 2002.

21. Jiajun Bu, Shulong Tan, Chun Chen, Can Wang, Hao Wu, Lijun Zhang, and Xiaofei He. Music recommendation by unified hypergraph: combining social media information and music content. In Proceedings of the international conference on Multimedia (MM '10). ACM, New York, NY, USA, 391-400. 2010.

22. Batul J. Mirza, Benjamin J. Keller, and Naren Ramakrishnan. Studying Recommendation Algorithms by Graph Analysis. J. Intell. Inf. Syst. 20, 2 (March 2003), 131-160. 2003.

23. Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases, in: Proc. 1993 ACM-SIGMOD Int. Conf. on Management of Data, Washington, D.C., 207-216. 1993.

24. Rakesh Agrawal and Ramakrishan Srikant, Fast algorithms for mining association rules, in: Proc. 20th Int. Conf. on Very Large Data Bases, Santiago, Chile, 487-499. 1994.

25. Thomas Tran and Robin Cohen. Hybrid Recommender Systems for Electronic Commerce, Proc. Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, Technical Report WS-00-04, AAAI Press, 2000.

26. Sumedh Sawant. Collaborative filtering using weighted bipartite graph projection: a recommendation system for yelp, in Proceedings of the CS224W: Social and Information Network Analysis Conference, December 2013.

27. D. Usha Nandini, Ezil Sam Leni, M. MariaNimmy. Mining of High Utility Itemsets from Transactional Databases, International Journal of Engineering and Advanced Technology (IJEAT), Volume-3, Issue-4, April 2014

28. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International World Wide Web Conference, 285–295, May 2001.

29. Badrul M. Sarwar, George Karypis, Joseph A. Konstan, John T. Riedl. Application of dimensionality reduction in recommender systems: A case study. In Proceedings of the WebKDD Workshop at the ACM SIGKKD. ACM, New York, 2000.

30. Liyan Zhang, Kai Zhang, and Chunping Li. A topical PageRank based algorithm for recommender systems. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '08). ACM, New York, NY, USA, 713-714. 2008

31. Marco Gori and Augusto Pucci. Research paper recommender systems: A random-walk based approach. In Web Intelligence, pages 778-781, 2006.

32. Tian Chen and Liang He. Collaborative Filtering Based on Demographic Attribute Vector, In proceedings of International conference on future Computer and Communication, IEEE, 2009.

33. Laila Safoury and Akram Salah, "Exploiting User Demographic Attributes for Solving Cold-Start Problem in Recommender System," Lecture Notes on Software Engineering, vol. 1, no. 3, pp. 303-307, 2013.

34. Yibo Chen, Chanle Wu, Ming Xie, Xiaojun Guo. 'Solving the Sparsity Problem in Recommender Systems Using Association Retrieval.', Journal of Computers, Vol 6, No 9, 1896-1902, September 2011.

35. Yuanyuan Wang; Chan, S.C.-F.; Ngai, G., "Applicability of Demographic Recommender System to Tourist Attractions: A Case Study on Trip Advisor," Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International

Conferences on , vol.3, pp.97-101, 4-7 Dec. 2012.

36. Lawrence Page, Sergey Brin, Rajeev Motwani and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report. Stanford InfoLab. 1999.

37. Jing Wang, Zhijing Liu, and Hui Zhao. Group Recommendation Based on the Page-Rank. Journal of Networks, VOL. 7, NO. 12, DECEMBER 2012

38. http://pentaho.dlpage.phi-integration.com/mondrian/mysql-foodmart-database

39. Jiawei Han and Yongjian Fu. Mining Multiple-Level Association Rules in Large Databases, IEEE Transactions on Knowledge and Data Engineering, vol. 11, no. 5, pp. 798–805. 1999.

40. Asela Gunawardana and Guy Shani. A Survey of Accuracy Evaluation Metrics of Recommendations Tasks. Journal of Machine Learning Research 10, 2935-2965. 2009.

41. Easley David and Kleinberg Jon. 2010. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, New York, NY, USA.

42. D. M W Powers, (2007/2011) Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation, School of Informatics and Engineering, Flinders University, Adelaide, Australia, TR SIE-07-001, Journal of Machine Learning Technologies 2:1 37--63.