TOPIC MODEL BASED RECOMMENDATION SYSTEMS FOR RETAILERS

SATICILAR İÇİN KONU MODELLEME YÖNTEMİNE DAYALI ÖNERİ SİSTEMİ

RİMA AL WASHAHİ

YRD. DOÇ. GÖNENÇ ERCAN
Supervisor

Submitted to Graduate School of Science and Engineering of Hacettepe University
as a Partial Fulfillment to the Requirements
for the Award of the Degree of Master of Science
in Computer Engineering

2016

This work named **"Topic Model Based Recommendation Systems For Retailers"** by **Rima AL WASHAHI** has been approved as a thesis for the Degree of **MASTER OF SCIENCE IN COMPUTER ENGINEERING** by the below mentioned Examining Committee Members.

Prof. Dr. Nihan KESİM ÇİÇEKLİ

Head

N. Çiçekli

Asst. Prof. Dr. Gönenç ERCAN

Supervisor

Assoc. Prof. Dr. Sevil ŞEN

Member

Sevil Şen

This thesis has been approved as a thesis for the Degree of MASTER OF SCIENCE IN COMPUTER ENGINEERING by Board of Directions of the Institute for Graduate School of Science and Engineering.

Prof. Dr. Salih Bülent Alten

Director of the Institute of

Graduate School of Science and Engineering

ABSTRACT

TOPIC MODEL BASED RECOMMENDATION SYSTEMS FOR RETAILERS

Rima AL WASHAHİ
Master of Science, Computer Engineering
Supervisor: Asst. Prof. Dr. Gönenç Ercan
November 2016, 58 pages

Nowadays, sellers need very good strategy to keep their customers' loyalty and to attract new customers to their shops. One of the important ways to accomplish this task is to present new and interesting items to their customers. In this thesis, we propose a new recommender system (RS) which recommends new items to sellers that they did not sell previously in their shop. Most of the RSs, recommend items to customers; unlike traditional RSs, proposed model is designed to suggest new items to sellers. In order to build the model, we adopted generative models that are used in text mining domain. Specifically, the probabilistic latent semantic analysis (pLSA) technique is extended to build the proposed RS. Several experiments are conducted using a real world dataset to validate the model. Furthermore, Collaborative Filtering (CF) method is used as a baseline algorithm to compare the performance of the proposed algorithm to state-of-the-art. Our experiments suggest that the proposed recommender system is more efficient than the pure CF algorithm for this task.

ÖZET

SATICILAR İÇİN KONU MODELLEME YÖNTEMİNE DAYALI ÖNERİ SİSTEMİ

Rima AL WASHAHİ
Yüksek Lisans, Bilgisayar Mühendisliği
Tez Danışmanı: Yrd. Doç. Dr. Gönenç Ercan

Kasım 2016, 58 sayfa

Günümüzde, satıcıların daha fazla müşteri kazanmak, var olan müşterilerinin sadakatını sürdürmek ve artırabilmek için iyi bir stratejiye ihtiyaçları vardır. Bu görevi gerçekleştimek için en iyi yollardan birisi müşteri profiline uygun ve müşterinin ihtiyacını karşılayabilecek yeni ürünleri (daha önce satışı bulunmayan) müşterilere sunmaktır. Bu tezde, satıcılara yardımcı olabilmek için yeni ürün önerme yeteneğine sahip bir öneri sistemi geliştirilmiştir. Klasik öneri sistemleri müşteriye ürün önermek için geliştirilmiştir, ancak bu çalışmada geliştirilen sistemin en büyük farklarından birisi müşteriye değil satıcıya yönelik yeni ürünleri önerebilen bir öneri sistemi geliştirmektir. Söz konusu sistemi geliştirebilmek için “Olasalıksal Örtük Anlam Analizi” metodu genişletilmiştir. Genişletilen metodun asıl amacı müşterileri alışveriş verilerini kullanarak satıcıların satma olasılığı yüksek olan yeni ürünleri tespit etmektir. Bu amaç doğrultusunda üç temel veri kaynağı dikkate alınmıştır; müşteri, müşterinin ziyaret ettiği veya alışveriş yaptığı satıcı ve müşterinin aldığı ürün. Bahsedilen veri kaynakları kullanılarak ilgili değişken olasılıkları hesaplayabilmek için olasılıksal model geliştirilmiştir. Bu modelin doğrulanabilmesi için gerçek müşteri veri setlerini kullanarak deneyler yapılmıştır. Ayrıca deney sonuçlarının karşılaştırılması ve daha uygun değerlendirme yapılabilmesi için “İşbirliğine Dayalı Filtreleme” methodu bazal algoritma olarak kullanılmıştır. Aynı deney koşulları sağlanarak iki algoritma aynı verilerle test edilmiş ve sonuçlar karşılaştırılmıştır. Test neticelerine göre bu tez doğrultusunda tasarlanan modelden daha doğru ve gerçeğe yakın sonuçlar alındığı gözlemlenmiştir.

ACKNOWLEDGEMENTS

I would like to thank my first supervisor Asst. Prof. Dr. Gönenç Ercan (Hacettepe University) for his help, advice, and patience during this thesis. He has always been very supportive, informative and I have learned a lot from him. Without his supervision the achievements in this thesis would not be possible.

I would also like to thank my second supervisor Asst. Prof. Dr. Joschka Boedecker (University of Freiburg) and Anas Alzoghbi (University of Freiburg) for their guidance and regular face to face meetings. My life in Germany was much easier with their help and advice.

Table of Contents

ABSTRACT i

ÖZET ii

ACKNOWLEDGEMENTS iv

ABBREVIATIONS vii

INTRODUCTION 1

1.1. Overview and Motivation 1

1.2. Contribution of the Thesis 2

1.3. Thesis Structure 2

RECOMMENDER SYSTEMS 4

2.1. The Utility Matrix 5

2.2. The Long Tail 6

2.3. Content Based Recommender Systems 6

2.4. Collaborative Filtering 7

2.4.1. User Based Collaborative Filtering 7

2.4.2. Item Based Collaborative Filtering 7

2.4.3. Computing Similarity 8

2.5. Singular Value Decomposition (SVD) 9

2.6. Probabilistic Methods for Recommender Systems 10

2.7. Recommender Systems Major Challenges 10

2.8. Related Work 12

3. PROBABILISTIC TOPIC MODELS 15

3.1. Generative Models: 16

3.2. Latent Dirichlet Allocation: 16

3.3. Probabilistic Latent Semantic Analysis 19

4. PROPOSED MODEL 27

5. EXPERIMENTS AND EVALUATION 34

5.1. Calculating Ground Truth 34

5.2. Defining number of latent variables 34

5.3. Item Association Test 35

5.4. Main Experiments 37

5.4.1. Data Set 37

5.4.2. Experimental Steps 37

5.4.3. Evaluation Metrics 38

5.4.4. Experimental Results 40

5.4.5. Discussion of the Main Experimental Results 41

6. CONCLUSION AND FUTURE WORK 43

7. APPENDICES 45

REFERENCES 49

ABBREVIATIONS

RS	Recommender System
CB	Content Based
CF	Collaborative Filtering
SVD	Singular Value Decomposition
LDA	Latent Dirichlet Allocation
PLSA	Probabilistic Latent Semantic Analysis
EM	Expectation Maximization
CNAM	Customer Need Aspect Model

1. INTRODUCTION

1.1. Overview and Motivation

In recent years fast growth of online platforms made products available to masses, therefore, it became necessary to use software tools such as Recommender Systems (RS) to help users. The main purpose of RSs is to assist users to make accurate decisions without spending too much on searching this vast amount of information. RS are widely used in many e-commerce web sites, such as Amazon and eBay.

Traditional RSs are designed to recommend meaningful items to their users. Those items depend on the purpose of the RS, for example Google recommends news to people while Facebook recommends people (friends) to people. In this thesis work, we propose to build a RS which aims to recommend "new items" to sellers. In our experiments we will target a domain where items are the consumer products sold in a supermarket chain and the sellers are the supermarket branches. The term "new item/s", in this context is used to refer to the products that were not previously in sale in that certain shop. For example, eBay, one of the very popular e-commerce web sites, has millions of sellers that try to sell products each day. One effective strategy for sellers to survive in this competition is to introduce new items to their users. A new item introduced in accordance with the customers' needs, can increase the revenue of the shop and also can keep and raise the customer loyalty [1]. However, for sellers it is almost difficult to analyze customers purchase behavior and suggest new items to their users. Therefore we design a model to help sellers decide which items to introduce to their customers and those items are expected to be good fit to customers' need.

In order to build the proposed model, 3 different observable variables are used from the real world transactional dataset; customer, shop and item. One challenge for working with this data is the privacy concerns as it is possible to acquire too much personal knowledge about a person by examining these data, even if the identity of the person is concealed. For these reasons in practice both customer and product information are anonymized to circumvent privacy issues. Our model can work with anonymized data without relying on product descriptions or customer profiles.

In order to reveal the underlying relations between customer – shop and customer – item, we introduce a latent variable. We use the latent variable for two main reasons;

1) Latent variable helps to calculate probability of an event which is not previously observed. Without a latent variable the probability of a new item being sold in a certain shop would always be zero as this shop never sold that item before. With the introduction of the latent variable it is possible to assign a probability to this new product.

2) Latent variable discovers hidden relationships between observable variables, i.e. it can group shops that sell similar items, similar customers buying similar items from similar shops and products bought by similar customers.

As topic models are used in text mining tasks to model the relationships between words and documents with respect to the latent variable topic, we decided to use this framework in a similar fashion in customer-product transaction data. In this thesis we explore the adoption of probabilistic topic models to design the proposed RS model.

1.2. Contribution of the Thesis

Our contributions can be summarized in two folds.

- We extend the PLSA topic model for modeling customer shopping behavior,
- We demonstrate that by using topic models it is possible to recommend items to sellers.

1.3. Thesis Structure

Section 1 gives the abstract of the complete thesis. It summarizes this work by giving key points and findings of the thesis.

Section 2 defines the researched problem, presents the importance and motivation of this problem. Most importantly it presents the list of the contribution of this thesis work.

Section 3 gives general overview of the RSs. Also, this section presents RSs functionality and importance, including definition of major terms in RSs; “Utility Matrix” and “Long

Tail". In addition, different approaches in RSs are explained. Those approaches cover Content Based, Collaborative Filtering and Probabilistic Methods for RSs.

Generally RSs encounter some challenges; the most common of them are explained in this section. Furthermore, in the sub section *Relevant Work* similar studies to ours are given in order to have a better weigh the contributions of our work.

Section 4 demonstrates fundamentals of Probabilistic Topic Models and their advantages. Then it explains two important probabilistic topic models; Latent Dirichlet Allocation and Probabilistic Latent Semantic Analysis and their usage. Probabilistic Latent Semantic is explained in more detail as our model is based on this method.

Section 5 presents the proposed model and its derivation in detail.

Section 6 demonstrates several experiments conducted to evaluate the proposed model explained in Section 5. Datasets, evaluation metrics that are used in these experiments as well as experimental results are also given in this section.

Section 7 concludes and summarizes thesis work by emphasizing our findings. Furthermore, this section also presents possible improvements for future work.

2. RECOMMENDER SYSTEMS

Recommender Systems (RSs) are software tools that try to predict future user behavior by analyzing users' past preferences or purchase history. In other words, RSs help users to find useful objects depending on their needs. Growth of internet sources made huge information and products available on the web. Therefore, each day for a user it is getting harder and time consuming to find what products they are exactly looking for. In order to address this problem various RS techniques are used in different scenarios. There are many popular websites that are actively using RS every day. For example; Pandora recommends music and MovieLens recommends movies.

Ricci et al.[2] gives five reasons why using RSs is important:

1) Users might give up buying items if they cannot find the products that they are looking for in a certain amount of time. For example in e-commerce web sites such as eBay oodles of products are available. Sometimes for users it is very hard to find the exact items that they need. This process can be time consuming and that may lead customer to give up buying product(s). Another case is that sometimes users do not know what to buy exactly, they know their need but they do not know which exact product may fit their need. In both cases RS plays important role to help customer to find products in a personalized way therefore we can argue that RS "*Increase the number of items sold*".

2) Most of the popular web sites that are using RSs has a search bar in their web page. Users write what they are searching for, and then system generates the list of items according to written word(s). Since there are many products available, users do not just go and try to find different items (s)he might like because this could be time consuming. However, RS can find various items by analyzing the items bought or liked by similar users. Therefore, it can be argued that RSs help "*Sell more diverse items.*"

3) A successfully designed RS analyzes users’ preferences and predicts what user wants. Furthermore, system updates itself with users’ reaction to the recommended items therefore this helps to “*Better understand what the user wants”*.

4) Successfully designed RSs suggest items which supplies users’ needs perfectly and that can “*Increase the user satisfaction”*.

5) RS can generate recommendation that fulfills the users’ needs and that makes users to become more loyal to website.

2.1. The Utility Matrix

Most RSs are built utilizing the past data such as users and their movie ratings or users’ transactional information. It is possible to represent the dataset as a utility matrix. In the utility matrix there are two entities, users and items [3]. Here “users” and “items” are general terms, such as in our case users are sellers, while in an e-commerce web sites users are customers. For the proposed model, each cell of this matrix represents the amount of sold item in the related shop. Example of utility matrix is shown in Figure 2.1 and it can be inferred that item I_2 is sold at shop S_3 7 times and similar rule applies for all item-shop pairs. As data gets much larger, (i.e. many users and many items exist), utility matrix gets sparser. One common problem of RSs face today is data sparsity [2]. Different techniques are applied in order to solve this problem.

Items

Shops

	I_0	I_1	I_2	I_3
S_0			12	
S_1	10			3
S_2				
S_3			7	
S_4		8		
S_5				5

Figure 2.1 Example of Shop Item Utility Matrix

2.2. The Long Tail

The long tail term is used to describe the relations of popular and non-popular items in the online and offline shops. The classical graphic that expresses this relation is shown in Figure 2.2. The left side of the graph represents popular items and the right side represents non popular (low selling rate) items. Customers of offline shops can only see those popular items. However, in online stores customers can see all available products. This difference between online and offline stores based on their products (popular and non-popular) can be expressed as long tail phenomenon [3]. In online stores with the help of RS non popular items can be presented / recommended to the users. This is one of the very important advantages of the RSs that help all items to be seen by users and that leads to selling variety of items.

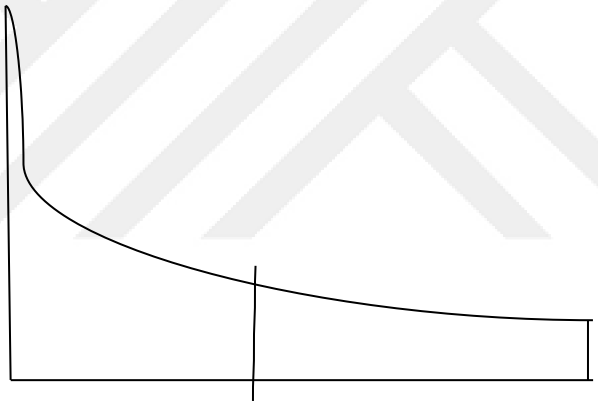

Figure 2.2 The Long Tail Phenomena

2.3. Content Based Recommender Systems

Content Based (CB) Recommender Systems recommend items to a user according to the content of user’s past preferences. In other words, system generates recommendations based on item features that match with the user profile. The basic process can be explained in two main steps;

1. System creates user profile using users past behavior, more precisely using item features that has been purchased or liked in the past by the user
2. Then, system generates recommendation by analyzing the attributes of these items and comparing them with the user profile

CB algorithm can be understood from its name that this technique mostly cares about item's content. CB method can be successfully used in item recommendation but it requires that the relevant attributes of the items can be extracted, or in other words it relies on the item's content. For example, if system recommends documents to its users then the CB algorithm analyzes documents' words (content). However, some items' features cannot be extracted easily such as movies and music, or they can be concealed due to privacy issues therefore applicability of these methods is limited depending on the nature of the items.

2.4. Collaborative Filtering

CF is considered as a state-of-the-art recommendation algorithm widely used in different domains [4]. CF is frequently used in popular web sites especially in social networks such as Facebook, LinkedIn, and Twitter. Recommendations only depend on similar users' preferences therefore unlike the Content based algorithm CF does not require knowledge about the content of the items.

CF algorithm can also be considered as a Nearest Neighborhood algorithm which relies on relationships between users or alternatively between items. More specifically, if CF algorithm depends on the relation of the user then it is called as User-based CF and similarly if CF algorithm considers item relations then it is called as Item-based CF.

2.4.1. User Based Collaborative Filtering

User-based CF applies same logic with the core CF algorithm. First, it finds similar users whose past behavior in the system is similar to the current user. Using the past preferences of these similar users, a prediction for the user in question is made.

2.4.2. Item Based Collaborative Filtering

In Item-based CF, again applies same logic with the core CF algorithm. First it computes current item's similarity with other items based on likes or purchases and if two items are liked or disliked by similar users, and then they can be considered as similar. Then, system generates a list of recommendations according to the most similar items to a user's already-rated items.

2.4.3. Computing Similarity

There are different algorithms that can be used to measure item or customer similarities. One common model is representing them as vectors and then measuring the cosine similarity between these vectors.

Cosine Similarity

Cosine similarity is used to find similarity between vectors. Similarity is measured using the angle between these vectors. For example, one way of finding similarity between customers is representing them as a vector and that vector is created by using the items that customers liked or disliked items in the past. Then using the following formula (2.1) the cosine-similarity between those vectors is calculated:

$$sim(i,j) = \cos(\vec{i},\vec{j}) = \frac{\vec{i}\,.\vec{j}}{\left|\left|\vec{i}\right|\right|^2 * \left|\left|\vec{j}\right|\right|^2} \tag{2.1}$$

Computing Predictions

After similarity computation the final step is to predict whether item should be recommended to a user or not (2.2):

$$r_{s,i} = \frac{1}{|S'|} \sum_{s \in S\prime} sim(s', s) * n(i, s') \tag{2.2}$$

$r_{s,i}$: Score value of the item i for the shop s

S' : Neighbor shops

i : Item i

$n(i,s')$: Number of item i sold by neighbor shop s’

$sim(s',s)$: Cosine similarity between shop s and neighbor shop s’

According to formula (2.2), algorithm first finds the most similar shops (has the highest similarity value) for each shop in the data set based on their item sales. And similar shops can be referred as neighbors (S'). For each item for the certain shop algorithm calculates score value ($r_{s,i}$) by using neighbors' information of the shop; the number of the item sold in the neighbor shop, similarity value of the neighbor shop and the number of neighbor shops.

2.5. Singular Value Decomposition (SVD)

Singular Value Decomposition has been used in many different areas such as signal processing[5], pattern recognition[6], latent semantic indexing [7].

The main purpose of SVD is to represent a utility matrix as a product of 3 different (also thinner) matrices as shown in the formula (2.3),

$$U \approx \mathrm{L\Sigma R} \tag{2.3}$$

In this example, we considered utility matrix U as a user-movie ratings matrix which represents the ratings of the users for the corresponding movie.

U: $u \times r$ Utility matrix such as user-movie matrix and called input matrix
L : $u \times z$ left singular matrix such as user-concept matrix
$\sum$: $z \times z$ singular diagonal matrix (presents the strength of the concepts)
R: $z \times$ r right singular matrix movie-concepts matrix

Note that, the value of concept z is significantly smaller than the user u and movie $r.$ In this context, "concept" refers to a theme of the movies.

In singular diagonal matrix singular values are ordered in descending order (the highest is the largest). The largest values affect the calculation of the utility matrix most and naturally the smallest values have a smaller effect. Eliminating the bottom values (the smallest values) reduces the dimensionality of the matrices, therefore, SVD is also known as a data dimensionality reduction technique. Reducing dimensionality also helps to filter noise and generate better predictions.

After utility matrix is decomposed into three lower dimensional matrices, recommendations can be generated by estimating the similarity (cosine similarity) between users or alternatively between items[8].

2.6. Probabilistic Methods for Recommender Systems

Probabilistic techniques are often successfully applied in RS problem. Grouping similar customers or items using probabilistic models is a technique previously explored in RS [2]. Bayesian Networks can be given as examples to probabilistic methods.

In the domain of RSs, probabilistic methods are designed to analyze users' past data (whole data) using probabilistic models and based on that model predicts users' future preferences/ behaviors with a probability [2]. Probabilistic models are designed to find user interest for an item by formulating it as the probability of an item being bought by a certain user $\boldsymbol{P}(\boldsymbol{i}|\boldsymbol{u})$**,** where i is an item and u is a user [4]. If the parameters of the probabilistic model can be estimated accurately, then the system can generate effective recommendations to its users. With the parameters of the model, RS can recommend the top 10 items with the highest $P(i|u)$ values.

Bayesian Clustering

In RSs discipline Bayesian Networks are used to classify/group users or items with using probabilistic framework [9]. The probabilistic model is built by using the complete data set. Variables are represented as circular and they are connected to each other with edges according to their dependencies. Calculations of the probability of the variables are based on Bayesian rule.

2.7. Recommender Systems Major Challenges

There are many challenges that recommender system researchers face today and those challenges can affect the performance of the algorithms. Li et al. [10] categorize those challenges as follows;

- **Data sparsity:** Nowadays millions of items are available especially in e-commerce web sites and each day this number is increasing. Therefore, finding similar users (that bought similar items) is getting harder. Most of the RS algorithms are using users/items similarity to generate recommenders. Thus, because of data sparsity algorithms may not perform accurately.
- **Scalability:** Especially, big web sites have millions of users and millions of items. Therefore, when designing a RS it should also consider the computational cost.
- **Cold start:** When new users or items enter the system, system cannot draw any information therefore it cannot generate recommendations either. One of the most naïve solutions for the cold start problem is recommending popular or trendy items to new customers. For example, in YouTube, when a user has no past video history it will recommend the most popular videos to this user. But once the user watches a video then system will have some idea about the user’s preference and then it will recommend similar videos to the past video that the user has watched.
- **Diversity and accuracy:** It is usually very effective to recommend popular items to users. However, users can also find those items by themselves without a recommender system. Recommender system should also find the less popular items but are likely to be preferred by the customers to recommend. One solution to this problem is using hybrid recommendation methods.
- **Vulnerability to attacks:** RSs can be target of several attacks trying to abuse the RS algorithms employed in the e-commerce web sites. Those attacks try to fool RS to wrongly suggest predetermined items for profit.
- **The value of time:** Customer needs/preferences tend to change in time. However, most RS algorithms do not consider time as a parameter.
- **Evaluation of recommendations**: There are several RS designed with different purposes and metrics proposed to evaluate the RS. However, how to choose the one that accurately evaluates the corresponding system is still not clear.

Those are the main challenges that most recommender systems are facing today but also they may face different challenges based on their design and user-item type. Also, as expected each system has different capabilities for dealing with these challenges.

2.8. Related Work

In this section we briefly overview some studies that are related to general RSs, CF based RSs, topic model based RSs and also studies that are relevant to this thesis.

Recommender systems are a sub-class of information retrieval systems and designed to predict users' future preferences by analyzing their past interaction with the system. Usage of RSs became more common in recent years. There are many different real world applications using RS in different domains such as eBay [11] for item recommendation to customers, Twitter[12] recommends friend/people to follow to its users, Levis[13] for cloth recommendation to its customers.

In the field of RS most known and common techniques are Content Based and Collaborative Filtering algorithms [14]. Text documents, online blog and similarly web pages can be successfully recommended based on a comparison between their content and a user profile using CB technique [15]. On the other hand, well known web sites use CF based RS algorithm. For example, Amazon.com uses traditional pure item-based CF to recommend same type of items (books, CDs) to similar groups of users. Likewise, MovieLens and Netflix use CF based RS for recommending movies to their users [4].

In the domain of Topic Models, they are mostly used to model unstructured texts and discover the hidden thematic structure in large archives of documents [16]. For example, Kong et al.[17] use pLSA to summarize the documents, Akita et al. [18] use pLSA for topic detection in meetings. However, topic models have been also adopted in many different domains for different types of data; they are utilized in information retrieval [3], multimedia retrieval [8] and many other related machine learning fields. Still, there are not many studies that try to build a RS using topic models but, there exist some studies trying to predict customer future purchase behavior. We consider this prediction task relevant to our work. Because the proposed model, first tries to model each customer purchase behavior for the certain sellers/shops and then generates recommendations for sellers/shops.

Since the following studies are more related to our work, they are presented in more details. Iwata et al.[19] propose a topic model to model user's purchase tendency by analyzing consumer purchase data with price information especially for marketing strategy. To model each user's purchase behavior they consider users, items and items' price information.

Iwata et al.[20] build a model using topic models that investigates the temporal effects on purchase behavior and also performs trend analysis. To build the model they use three different data sources; customers, items and time.
Sun et al. [21] analyze customer group's (people with similar taste) purchase behavior from group deal websites by modifying traditional LDA technique. The goal of the study is to predict potential customers who might join the group purchasing events. Ishigaki et al. [22] use topic modeling technique for modeling market responses for large-scale transaction data for finding the group of products that are likely to be purchased by certain users. Christidis et al. [23] try to model customer purchase behavior by using topic modeling techniques for recommending item(s) to customers.

Note that, studies Iwata et al.[19], Iwata et al.[20], Sun et al. [21] , Ishigaki et al. [22] and Christidis et al. [23] try to model customer behavior using topic models with different data sources such as customer item preferences, but none of them considered retailers' information as a data source. Generally related RS studies use a customer past data set which is obtained from a certain shop (only one shop information exists in the whole data set) and focus on predicting customer item preferences. According to our literature review, there is only one study Giering [24] that has the same purpose to our proposed model; "retailer sale prediction and based on that product recommendation to retailers". In this research, to construct the model the following steps are applied;
First, shops are clustered based on customer demographic information such as age and income. The clustering process is conducted using 3 different techniques; K-mean, Correlation and pLSA clustering. Secondly after clustering shops, for each shop group sales are modeled.

Finally, Singular Value Decomposition method is used to generate the recommendations to retailers.

The important difference between our method and Giering is that they combine different algorithms to build the RS while in our model we use only 1 algorithm; pLSA with an extended graphical model. Another important difference is Giering uses more data sources to build the model such as; customer demographic information (age, wealth and income), however we do not use any customer demographic information. Because, we aim to imitate the real world scenario while building the proposed model. In real life, mostly because of the privacy issues customer demographic information may not be feasible to acquire.

To sum up, with regard to our literature review there are considerable amount of RS studies and successfully implemented RS algorithms. Most of the studies focus on analyzing user behavior and based on that generates recommendations to its users. In addition, the other studies that have similar purpose use different methods and algorithms. Therefore, our proposed model can be considered as a unique study with respect to methodology that we improved to design a RS in this thesis.

3. PROBABILISTIC TOPIC MODELS

Today there are millions of articles, web pages, books and blogs available online. Furthermore, each day the amount of text documents are increasing with contributions from social networks and technological developments. Therefore, finding what we are exactly looking for is not an easy task as it used to be and it can be very time consuming. For example; for researchers, there are millions of scientific articles available online, to find the related ones is a challenge for scientists. It is not feasible to read each text and organize or categorize them. So, it is necessary to use software tools to organize (group/cluster) them [25]. As an another example, most journals archive their issues, storing every published article, and therefore, they must store a large amount of information. Without using computational tools organizing such a big unstructured text collection is impossible by only using human labour. Thus, researchers build different probabilistic models for theme discovery from a large unstructured text corpus and they called them probabilistic topic models [26].

Probabilistic topic models are algorithms designed to discover the hidden topic of the documents. In other words, they are statistical methods trying to discover the hidden theme of each document by analyzing the frequency of the words. The main idea behind topic models is an assumption that documents are mixtures of topics (normal distribution) and topics are normal distribution over words. Topic models are generative models which basically mean that generating a document is considered as a probabilistic process. This process can be explained in 3 main steps as follows;

- Choose a document to be generated
- Choose topic for each word of the document
- Draw a word based on the topic that has been chosen

Probabilistic topic model algorithms are unsupervised algorithms that do not need any prior labels or any other annotation. Successfully implemented algorithms can categorize, cluster or organize them with respect to their hidden topic.

Even though topic models are originally designed to organize or find the hidden topic of unstructured documents [27] they have been adopted in many different domains with different types of data. For example, they are utilized in information retrieval[28], multimedia retrieval [29] and collaborative filtering [30].

Blei et al.[16] summarize advantages of topic models;

- Topic modeling algorithms can also be used to “analyze streaming collections, and consumed from a Web API”.

- Topic modeling algorithms are very flexible algorithms and can be adopted in many different machine learning fields

3.1. Generative Models:

Generative model for documents is a set of probabilistic rules that explains how each document is generated word-by-word using latent variable/s [16]. The main goal is to find the latent variables that explain how the observable variables are generated, thus, topic modeling can be considered as a statistical estimation modeling [16].

Generative models are based on probabilistic graph models representing the set of random variables and their conditional dependencies. Random variables can be observable or latent. For example, for modeling document generation process there are two observable variables, namely documents and words, with a latent variable representing the topic distribution of the documents.

3.2. Latent Dirichlet Allocation:

“Latent Dirichlet Allocation (LDA) is a generative probabilistic topic model for large unstructured text documents (bag of words)” [21]. This model can also be considered as one of the simplest topic model. The goal of the model is to use statistical methods to find documents’ hidden theme in huge document collections. This method is not only used to find topic of the documents but also to summarize or find the similarity between documents. To generate the corpus, the generative process explained under Section 3 is followed for each document in the corpus.

The graphical representation of LDA model is shown in Figure 3.1. As it can be seen from the graph that there are three different levels of variables and parameters;

- First level is corpus level parameters α and η they are sampled in the beginning i.e. before start generating the corpus
- Second level is document level variables θ_d and β_k they are sampled once for generating each document
- Third level variables are word level variables $Z_{d,n}$and $W_{d,n}$and they are generated for each word of all documents in the corpus

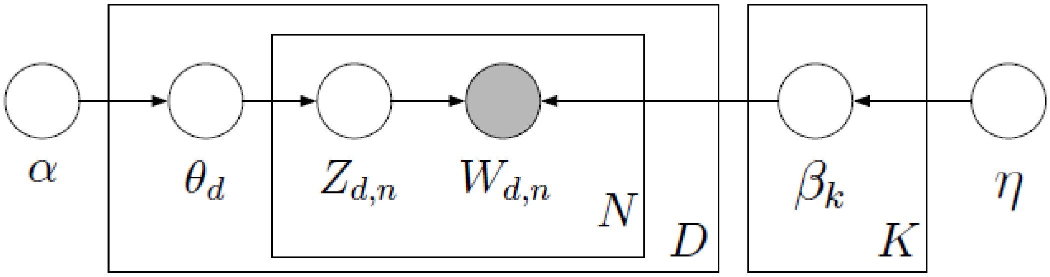

Figure 3.1: Graphical representation of LDA

Figure 3.1, represents the graphical appearance of LDA, the darker node means an observable variable while lighter nodes mean latent-unobservable random variables.

N	Number of words
D	Number of documents
K	Number of topics
α	K dimensional vector where each element α_k is prior weight of topic k in a document
θ	K dimensional vector denoting the topic distribution for all documents
θ_d	K dimensional vector denoting the topic distribution for d document
Z	N-dimension vector represents topic of all words in all documents
$Z_{d,n}$	The topic of word w in document d
W	N-dimension represents all words in all documents
$W_{d,n}$	The word w in document d
β_k	V-dimension vector indicates probability distribution of words in topic k
η	K dimensional vector each element such that η_k is prior weight of word in a topic k

Figure 3.2 shows the description of all the symbols that are used in the graphical model

For a predefined k number of topics Dirichlet density can be calculated using the following formula (3.1)[16]:

$$P(\theta|\alpha) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_i \theta_i^{\alpha_i - 1} \qquad (3.1)$$

The formula of joint probability of the complete LDA probabilistic graph as follows (3.2)(for a single document) [16];

$$P(\theta, z, w, \beta|\alpha, \eta) = \prod_{k} P(\beta_k, \eta) \prod_{d} [P(\theta_d, \alpha) \prod_{n} P(Z_{d,n}, \theta_d) p(W_{d,n}|Z_{d,n}, \beta)] \tag{3.2}$$

In order to find out θ,β and the topic assignment for each word z, Expectation Maximization (EM) algorithm or Gibbs sampling technique can be used [31].

3.3. Probabilistic Latent Semantic Analysis

Probabilistic latent semantic analysis (pLSA), also known as probabilistic latent semantic indexing (pLSI) is one of the commonly used probabilistic topic model algorithm. Although pLSA in the beginning was mostly used in information retrieval [27], recently it has also been used in different machine learning fields such as image classification [32], web usage mining [33].

pLSA is a generative model originally developed for organizing unstructured documents (bag of words) with possible latent topics by analyzing document's word co-occurrence. A document (distribution over topics) can be described with latent topics. Similarly, a latent topic is a distribution over a fixed size (unique number of words in whole corpus) vocabulary. This sentence can be explained in two steps as follows;

1. Documents consist of topics (fixed size) and each document (d) has a probability for a certain topic (z). The mathematical expression of this can be indicated as follows;

$$P(d|z)$$

$P(d|z)$ is also used to find similar documents or for summarization of documents depending on their topic(s). In this context one important point is that each document is a multinomial distribution over topics and the size of topics is fixed (specified in the beginning), the mathematical formula for this assumption is as follows (3.3);

$$\sum_{z}^{Z} p(d|z) = 1 \tag{3.3}$$

2. Similarly, topics consist of words (fixed size) and each word (w) has a probability for a certain topic (z). The mathematical expression of this can be indicated;

$$P(w|z)$$

Each word is associated with a topic. The size of words is fixed and this size is equal to number of all unique words in the corpus. In other words, topics are consisting of words and each of the topics are multinomial distributions over words. This can be formulized as follows (3.4);

$$\sum_{w}^{W} p(w|z) = 1 \tag{3.4}$$

The main idea behind the generative model can be explained for pLSA as follows;

> To generate a document from the corpus;
> 1. Pick a latent variable with a probability P(d|z),
> 2. Generate a word with a probability P(w|z), based on the latent variable from the first step

For each document from the corpus this process is repeated.

pLSA model can be described as a probabilistic graphical model as shown in Figure 3.3.

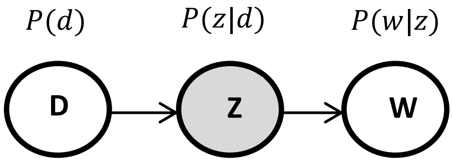

Figure 3.3 The probabilistic latent semantic analysis (pLSA) graphical representation

D: represents documents,

Z: represents topics, latent variable that is not observable,

W: represents words.

This is a probabilistic model with 3 variables. Here document and word are observable but topic is a latent variable.

pLSA as seen in the Figure 3.4 is a basic probabilistic graph model with 3 variables. Based on Bayes rule we can easily formulate the joint probability of documents and words as below (3.5);

$$\mathrm{P}(\mathrm{d_i}, \mathrm{w_j}) = \mathrm{P}(\mathrm{d_i}) \sum_{\mathrm{k}} \mathrm{P}(\mathrm{w_j}|\mathrm{z_k})\mathrm{P}(\mathrm{z_k}|\mathrm{d_i}) \tag{3.5}$$

By applying Bayesian rule we can have second formula (3.6) as shown below which leads to similar result but different inference process.

$$P(z|d) = \frac{P(d|z)P(z)}{P(d)}$$

$$P(z|d)P(d) = P(d|z)P(z)$$

$$P(d) = \frac{P(d|z)P(z)}{P(z|d)} \tag{3.6}$$

Applying Bayes rule to formula (3.7) we get the equivalent equation as below:

$$\mathrm{P}(\mathrm{d_i},\mathrm{w_j}) = \sum_{\mathrm{k}} \mathrm{P}(\mathrm{w_j}|\mathrm{z_k})\mathrm{P}(\mathrm{d_i}|\mathrm{z_k})\mathrm{P}(\mathrm{z_k}) \tag{3.7}$$

And whole data set can be formulized (3.8) as follow:

$$D = \prod_{d}\prod_{w} P(d,w)^{n(d,w)} \tag{3.8}$$

$\mathrm{n}(\mathrm{d},\mathrm{w})$: number of times term w occurs in document d

Here we use log likelihood function (formula (3.9)) to estimate the parameters of the model. The main goal is to learn model using statistical estimation from the given data set.

$$logD = \sum_{i=1}^{N}\sum_{j=1}^{M} \mathrm{n}(\mathrm{d_i},\mathrm{w_j})\mathrm{logP}(\mathrm{d_i},\mathrm{w_j}) \tag{3.9}$$

Since we have two different joint probability formulas (3.5) and (3.7), the total log likelihood can be estimated in two different ways by using formulas (3.10) or (3.11):

$$L_1 = \sum_{d}\sum_{w} n(d,w) \log\left[\mathrm{P(d)} \sum_{\mathrm{z}} \mathrm{P(w|z)P(z|d)}\right] \tag{3.10}$$

or

$$L_2 = \sum_{d}\sum_{w} n(d,w)log\left[\sum_{z} P(w|z)P(d|z)P(z)\right] \tag{3.11}$$

A common algorithm for learning models especially when log likelihood function has unobservable variables is Expectation Maximization algorithm [34].

EM algorithm for the first log likelihood $\boldsymbol{L_1}$:

The Q-function for the complete likelihood E[L_1] [35] formula (3.12):

$$E[L_1] = \sum_{i=1}^{N}\sum_{j=1}^{M} n(d_i, w_j) \sum_{k} P(z|w,d)\log\left[P(d_i)P(w_j|z_k)P(z_k|d_i)\right] \tag{3.12}$$

E-step, is calculated using the current value of the parameters as in formula (3.13):

$$\begin{aligned} P(z|w,d) &= \frac{P(w,z,d)}{P(w,d)} \\ P(z|w,d) &= \frac{P(w|z)P(z|d)P(d)}{\sum_z P(w|z)P(z|d)P(d)} \end{aligned} \tag{3.13}$$

For M-step, Q-function must be maximized to find maximum value of the parameters [35] as follows:

$$E[L_1] + \alpha\left[1 - \sum_{d} P(d)\right] + \beta\sum_{z}\left[1 - \sum_{w} P(w|z)\right] + \gamma\sum_{d}\left[1 - \sum_{z} P(z|d)\right] \tag{3.14}$$

α, β, γ are Lagrange Multipliers.

To maximize Q function derivatives are taken as follows:

$$\begin{aligned} \frac{\partial H}{\partial P(d)} &= \sum_{w}\sum_{z} n(d,w)\frac{P(z|w,d)}{P(d)} - \alpha = 0 \\ \frac{\partial H}{\partial P(d)} &= \sum_{w}\sum_{z} n(d,w)p(z|w,d) - \alpha P(d) = 0 \end{aligned} \tag{3.15}$$

$$\frac{\partial \mathrm{H}}{\partial \mathrm{P}(\mathrm{w}|\mathrm{z})} = \sum_{\mathrm{d}} \mathrm{n}(\mathrm{d},\mathrm{w}) \frac{\mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}{\mathrm{P}(\mathrm{w}|\mathrm{z})} - \beta = 0$$

$$\frac{\partial \mathrm{H}}{\partial \mathrm{P}(\mathrm{w}|\mathrm{z})} = \sum_{\mathrm{d}} \mathrm{n}(\mathrm{d},\mathrm{w})\mathrm{p}(\mathrm{z}|\mathrm{w},\mathrm{d}) - \beta\mathrm{P}(\mathrm{w}|\mathrm{z}) = 0 \qquad (3.16)$$

$$\frac{\partial \mathrm{H}}{\partial \mathrm{P}(\mathrm{z}|\mathrm{d})} = \sum_{\mathrm{w}} \mathrm{n}(\mathrm{d},\mathrm{w}) \frac{\mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}{\mathrm{P}(\mathrm{z}|\mathrm{d})} - \gamma = 0$$

$$\frac{\partial \mathrm{H}}{\partial \mathrm{P}(\mathrm{z}|\mathrm{d})} == \sum_{\mathrm{w}} \mathrm{n}(\mathrm{d},\mathrm{w})\mathrm{p}(\mathrm{z}|\mathrm{w},\mathrm{d}) - \gamma\mathrm{P}(\mathrm{z}|\mathrm{d}) = 0 \qquad (3.17)$$

For the M-step total data log-likelihood is maximized and after taking the derivatives M step equations are obtained as below;

$$\mathrm{P}(\mathrm{d}) = \frac{\sum_{\mathrm{w}}\sum_{\mathrm{z}} \mathrm{n}(\mathrm{d},\mathrm{w})\mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}{\sum_{\mathrm{d}}\sum_{\mathrm{w}}\sum_{\mathrm{z}} \mathrm{n}(\mathrm{d},\mathrm{w})\mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}$$

$$\mathrm{P}(\mathrm{d}) = \frac{\mathrm{n}(\mathrm{d})}{\sum_{\mathrm{d}} \mathrm{n}(\mathrm{d})} \qquad (3.18)$$

$$\mathrm{P}(\mathrm{w}|\mathrm{z}) = \frac{\sum_{\mathrm{d}} \mathrm{n}(\mathrm{d},\mathrm{w})\mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}{\sum_{\mathrm{w}}\sum_{\mathrm{d}} \mathrm{n}(\mathrm{d},\mathrm{w})\mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})} \qquad (3.19)$$

$$\mathrm{P}(\mathrm{z}|\mathrm{d}) = \frac{\sum_{\mathrm{w}} \mathrm{n}(\mathrm{d},\mathrm{w})\mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}{\sum_{\mathrm{z}}\sum_{\mathrm{w}} \mathrm{n}(\mathrm{d},\mathrm{w})\mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}$$

$$\mathrm{P}(\mathrm{z}|\mathrm{d}) = \frac{\sum_{\mathrm{w}} \mathrm{n}(\mathrm{d},\mathrm{w})\mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}{\mathrm{n}(\mathrm{d})} \qquad (3.20)$$

EM algorithm for the second log likelihood L_2 :

For the both log likelihood formulas L_1 and L_2 E- step is calculated in the similar way;

$$\mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d}) = \frac{\mathrm{P}(\mathrm{w},\mathrm{z},\mathrm{d})}{\mathrm{P}(\mathrm{w},\mathrm{d})} \tag{3.21}$$

$$\mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d}) \; = \frac{\mathrm{P}(\mathrm{w}|\mathrm{z})\mathrm{P}(\mathrm{z}|\mathrm{d})\mathrm{P}(\mathrm{d})}{\sum_{\mathrm{z}} \mathrm{P}(\mathrm{w}|\mathrm{z})\mathrm{P}(\mathrm{z}|\mathrm{d})\mathrm{P}(\mathrm{d})}$$

Q-function as follows:

$$\mathrm{E}[\mathrm{L}_2] + \alpha\left[1 - \sum_{\mathrm{z}} \mathrm{P}(\mathrm{z})\right] + \beta \sum_{\mathrm{z}} \left[1 - \sum_{\mathrm{w}} \mathrm{P}(\mathrm{w}|\mathrm{z})\right] + \gamma \sum_{\mathrm{d}} \left[1 - \sum_{\mathrm{z}} \mathrm{P}(\mathrm{d}|\mathrm{z})\right] \tag{3.22}$$

α, β, γ are Lagrange Multipliers.

Derivatives for the E step are as follows:

$$\frac{\partial \mathrm{H}}{\partial \mathrm{P}(\mathrm{z})} = \sum_{\mathrm{d}} \sum_{\mathrm{w}} \mathrm{n}(\mathrm{d},\mathrm{w}) \frac{\mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}{\mathrm{P}(\mathrm{z})} - \alpha = 0$$

$$\frac{\partial \mathrm{H}}{\partial \mathrm{P}(\mathrm{z})} = \sum_{\mathrm{d}} \sum_{\mathrm{w}} \mathrm{n}(\mathrm{d},\mathrm{w}) \mathrm{p}(\mathrm{z}|\mathrm{w},\mathrm{d}) - \alpha \mathrm{P}(\mathrm{z}) = 0 \tag{3.23}$$

$$\frac{\partial \mathrm{H}}{\partial \mathrm{P}(\mathrm{d}|\mathrm{z})} = \sum_{\mathrm{w}} \mathrm{n}(\mathrm{d},\mathrm{w}) \frac{\mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}{\mathrm{P}(\mathrm{d}|\mathrm{z})} - \gamma = 0$$

$$\frac{\partial \mathrm{H}}{\partial \mathrm{P}(\mathrm{d}|\mathrm{z})} = \sum_{\mathrm{w}} \mathrm{n}(\mathrm{d},\mathrm{w}) \mathrm{p}(\mathrm{z}|\mathrm{w},\mathrm{d}) - \gamma \mathrm{P}(\mathrm{d}|\mathrm{z}) = 0 \tag{3.24}$$

$$\frac{\partial \mathrm{H}}{\partial \mathrm{P}(\mathrm{w}|\mathrm{z})} = \sum_{\mathrm{d}} \mathrm{n}(\mathrm{d},\mathrm{w}) \frac{\mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}{\mathrm{P}(\mathrm{z}|\mathrm{w})} - \beta = 0$$

$$\frac{\partial \mathrm{H}}{\partial \mathrm{P}(\mathrm{w}|\mathrm{z})} = \sum_{\mathrm{d}} \mathrm{n}(\mathrm{d},\mathrm{w}) \mathrm{p}(\mathrm{z}|\mathrm{w},\mathrm{d}) - \beta \mathrm{P}(\mathrm{w}|\mathrm{z}) = 0 \qquad (3.25)$$

After taking the derivatives M-step equations as follows (3.26), (3.27), (3.28):

$$\mathrm{P}(\mathrm{d}) = \frac{\sum_{\mathrm{w}} \sum_{\mathrm{z}} \mathrm{n}(\mathrm{d},\mathrm{w}) \mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}{\sum_{\mathrm{d}} \sum_{\mathrm{w}} \sum_{\mathrm{z}} \mathrm{n}(\mathrm{d},\mathrm{w}) \mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}$$

$$\mathrm{P}(\mathrm{d}) = \frac{\mathrm{n}(\mathrm{d})}{\sum_{\mathrm{d}} \mathrm{n}(\mathrm{d})} \qquad (3.26)$$

$$\mathrm{P}(\mathrm{w}|\mathrm{z}) = \frac{\sum_{\mathrm{d}} \mathrm{n}(\mathrm{d},\mathrm{w}) \mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}{\sum_{\mathrm{w}} \sum_{\mathrm{d}} \mathrm{n}(\mathrm{d},\mathrm{w}) \mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})} \qquad (3.27)$$

$$\mathrm{P}(\mathrm{z}|\mathrm{d}) = \frac{\sum_{\mathrm{w}} \mathrm{n}(\mathrm{d},\mathrm{w}) \mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}{\sum_{\mathrm{z}} \sum_{\mathrm{w}} \mathrm{n}(\mathrm{d},\mathrm{w}) \mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}$$

$$\mathrm{P}(\mathrm{z}|\mathrm{d}) = \frac{\sum_{\mathrm{w}} \mathrm{n}(\mathrm{d},\mathrm{w}) \mathrm{P}(\mathrm{z}|\mathrm{w},\mathrm{d})}{\mathrm{n}(\mathrm{d})} \qquad (3.28)$$

Using formulas obtained from the E-step formula (3.21) and M-step formulas (3.26), (3.27) and (3.28) pLSA algorithm can be implemented easily by repeatedly executing E-step and M-step. Algorithm should run iteratively until it reaches a local maxima. In other words, after finding the maximum value of the log-likelihood, joint probability of the pLSA model can be calculated.

4. PROPOSED MODEL

We propose a recommender system capable of recommending new items to shops. In order to build the proposed RS we decided to design a model capable of finding good (high probability of being sold) items for the shops. The mathematical interpretation of the main idea is to find the probability of an item for the given shop $P(i|s)$. Based on this probability, recommendations are generated.

There are 2 main observations considered to find $P(i|s)$;

- A customer and an item s/he bought
- A customer and a shop s/he visited

We aimed to group customers who visit similar shops and buy similar items. Therefore, the hidden variable is used to reveal the underlying relation of observable variables; customers, items and shops. In this thesis, we explore the adoption of probabilistic latent semantic analysis (pLSA) technique to build the proposed RS.

However traditional pLSA model uses three variables; 2 observables and 1 unobservable variable. As we mentioned in this section we have 3 observable variables; customers, items, shops and 1 unobservable variable. We called the unobservable variable *need* to represent the idea of customers are grouped based on their needs. Since we have one more observable variable we extended traditional pLSA as seen Figure 4.1.

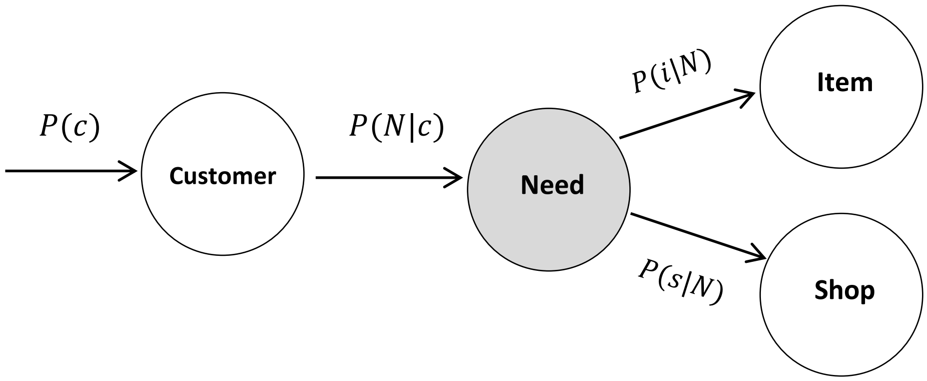

Figure 4.1: Graphical representation of proposed model. Model shows the intermediate layer of latent variable that links the customer item and the shops.

We decided to call the proposed model as *Customer Need Aspect Model* (CNAM). As seen from the Figure 4.1 one more branch is added to standard pLSA model. The reason is instead of 2 we consider 3 observable variables for customer behavior prediction. Before analyzing the model in detail there are some assumptions that need to be explained for better understanding the logic behind the model.

First we assume that customers have a multinomial distribution over fixed size of Need N. In other words customers have multiple Needs and this assumption can be formulated as follows (4.1);

$$\sum_{k=1}^{K} P(n_k|c) = 1 \tag{4.1}$$

Similarly, a need N is a multinomial distribution over fixed size of Shop and Item, (4.2) and (4.3);

$$\sum_{m=1}^{M} P(s_m|n) = 1 \tag{4.2}$$

And

$$\sum_{p=1}^{P} P(i_p|n) = 1 \tag{4.3}$$

The model only uses the basic transaction information of the customers to generate relevant recommendations. This transaction information should only include customer, namely only the events, item being bought by customer and shop/seller being visited by the customer. The latent variable “Need” is associated with each observation. The observation is an item being purchased and the shop being visited by the customer. Each customer can be represented as a mixture of needs weighted by the probability $\mathrm{P(n|c)}$. Shop and item expresses a need with probability $\mathrm{P(s|n)}$, $\mathrm{P(i|n)}$ respectively.

This generative model implicitly discovers similar purchase behavior and groups them to one or more latent classes as shown in the Figure 4.2. Similar purchase behavior means customers that are interested in similar items and visits similar shops. An example for a need could be a “camping trip”, where outdoor camping gear like tents and sleeping bags will be associated through this need. Furthermore, the shops that sell outdoor camping gear will be associated with each other through only transactional information.

As depicted in Figure 4.2, observed variables are Customers, Items and Shops and they are being associated with the latent variable ‘Need’. Our generative model simulates a generative process as follows:

- For each customer c
    - Choose a need
    - Choose an item and a shop according to the need that has been already chosen (item and shop are analyzed independently)

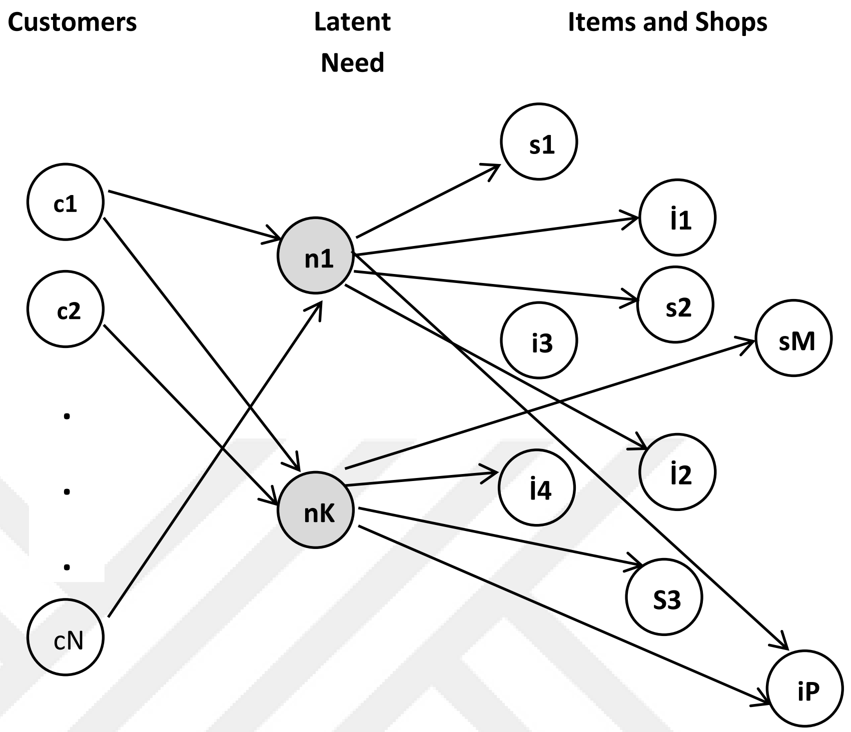

Figure 4.2 Proposed model illustrations

Customer: c ∈ C = {c1, · · ·, cN}

Shop: s ∈ S = {s1, · · · , sM}

Item: i ∈ I = {i1, · · · , iP}

Need: n ∈ N = {n1, · · · , nK}

The probabilities in the model can be formulated (4.4) using the joint distribution of the variables. We can obtain joint probability $P(c, s, i)$ using the product rule as it is previously done for the pLSA model:

$$P\left(c_k, i_{j,} s_l\right) = P(c_k) \sum_{z} P(n_z|c_k) P(i_{j,}|n_z) P(s_l|n_z) \tag{4.4}$$

Also, we have another equation from Bayes rule:

$$\mathrm{P(n|c)P(c) = P(c|n)P(n)} \tag{4.5}$$

So, we can modify our main joint probability equation as follows (4.6):

$$\mathrm{P}\big(\mathrm{c_k, i_{j,}}s_l\big) = \sum_{\mathrm{z}} \mathrm{P(n_z)P(c_k|n_z)P(i_{j,}|n_z)P(}s_l\mathrm{|n_z)} \tag{4.6}$$

Both joint probability equations will produce the same results for the same dataset. Note that the variables shop and item are assumed to be independent from each other, allowing to estimate the probability of buying an item from a shop that never sold the item before. If the two variables are dependent this probability would be equal to zero, even if the shop is used by customers to fulfill similar needs.

The whole data set can be generated by formula (4.7):

$$D = \prod_{k=1}^{K}\prod_{j=1}^{J}\prod_{l=1}^{L} \mathrm{P}\big(\mathrm{c_k, i_j, s_l}\big)^{\mathrm{n}\left(\mathrm{c_k, i_{j,}}s_l\right)} \tag{4.7}$$

Where $\mathrm{n}\big(\mathrm{c_k, i_j,} s_l\big)$ denotes the number of times customer $\mathrm{c_k}$ bought item $\mathrm{i_j}$ from the shop s_l

Log likelihood function can be calculated using formula (4.8);

$$logD = \sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{l=1}^{L} \mathrm{n}\big(\mathrm{c_k, i_{j,}}s_l\big)\, \mathrm{logP}\big(\mathrm{c_k, i_j, s_l}\big) \tag{4.8}$$

In the E-step, the posterior probabilities of the latent variables are computed based on the current estimates of the parameters as in (4.9):

$$\mathrm{P}(\mathrm{n_z}|\mathrm{c,s,i}) = \frac{\mathrm{P}(\mathrm{n,c,s,i})}{\mathrm{P}(\mathrm{c,s,i})}$$

$$\mathrm{P}(\mathrm{n_z}|\mathrm{c,s,i}) = \frac{\mathrm{P}(n_z)\mathrm{P}(c|\mathrm{n_z})\mathrm{P}(\mathrm{i}|\mathrm{n_z})\mathrm{P}(\mathrm{s}|\mathrm{n_z})}{\sum_{\mathrm{z}} \mathrm{P}(n_z)\mathrm{P}(c|\mathrm{n_z})\mathrm{P}(\mathrm{i}|\mathrm{n_z})\mathrm{P}(\mathrm{s}|\mathrm{n_z})} \tag{4.9}$$

In the M-step, the parameters are updated to maximize the posterior probability of the latent variable from the E-step (4.10), (4.11), (4.12), (4.13):

$$\mathrm{P}(\mathrm{n})\ \alpha \sum_{c,s,i} n(c,i,s)\ \mathrm{P}(\mathrm{n}|\mathrm{c,s,i}) \tag{4.10}$$

$$\mathrm{P}(\mathrm{i}|\mathrm{n})\ \alpha \sum_{c,s} n(c,i,s)\ \mathrm{P}(\mathrm{n}|\mathrm{c,s,i}) \tag{4.11}$$

$$\mathrm{P}(\mathrm{s}|\mathrm{n})\ \alpha \sum_{c,i} n(c,i,s)\ \mathrm{P}(\mathrm{n}|\mathrm{c,s,i}) \tag{4.12}$$

$$\mathrm{P}(\mathrm{c}|\mathrm{n})\ \alpha \sum_{s,i} n(c,i,s)\ \mathrm{P}(\mathrm{n}|\mathrm{c,s,i}) \tag{4.13}$$

The E and M steps are iterated until the log likelihood *logD* converges to a local maximum. After log likelihood converges to a local maximum, then we can calculate the probability of the desired variables.

In our case, our goal is to find the relevant items from the system to recommend to sellers. Using formula (4.14) items are ranked for the target shop according to probability of $P(i|s)$.

$$p(i|s) \propto \sum_{c=1} p(c,s,i) \tag{4.14}$$

5. EXPERIMENTS AND EVALUATION

The proposed CNAM and pure CF described in Section 4 and Section 2 are implemented. As it is mentioned in Section 2 CF is a state-of-the-art RS method [36]; therefore, we decided to use it as a baseline algorithm. The same tests with the same dataset are conducted for both algorithms to compare the effectiveness and analyze the properties of the proposed model.

5.1. Calculating Ground Truth

The term ground truth is used to express real probabilities of items being sold in a certain shop - $P(i|s)$. Those values are calculated using the actual number of sales in the original data set.

5.2. Defining number of latent variables

Before designing the experimental setups for the proposed model the number of latent variables should be determined. We set this parameter, namely the number of needs, empirically. For the sake of being fair, a different data set is used to test different number of latent variables. In this experiment, the dataset used is acquired from Microsoft foodmart 2000 database [37]. *Foodmart* is a small sparse real world dataset. Data set consists of 8736 customers, 1559 items and 24 shops.

We conveyed our experiments with different number of latent variables separately. Experimental steps are as follows:

For each latent number in 8, 15, 25, 35, 50, 75 experiments are repeated:

- Train proposed model with the complete data set; in other words, run the proposed algorithm until it reaches the maximum Log likelihood
- Calculate $p(i|s)$ for each shop and item independently
- Compute Root Mean Square Error using proposed model and ground truth based probabilities

Since there are 5 different number of latent variable, we have 5 different experimental results plot of the results is shown in Figure 5.1.

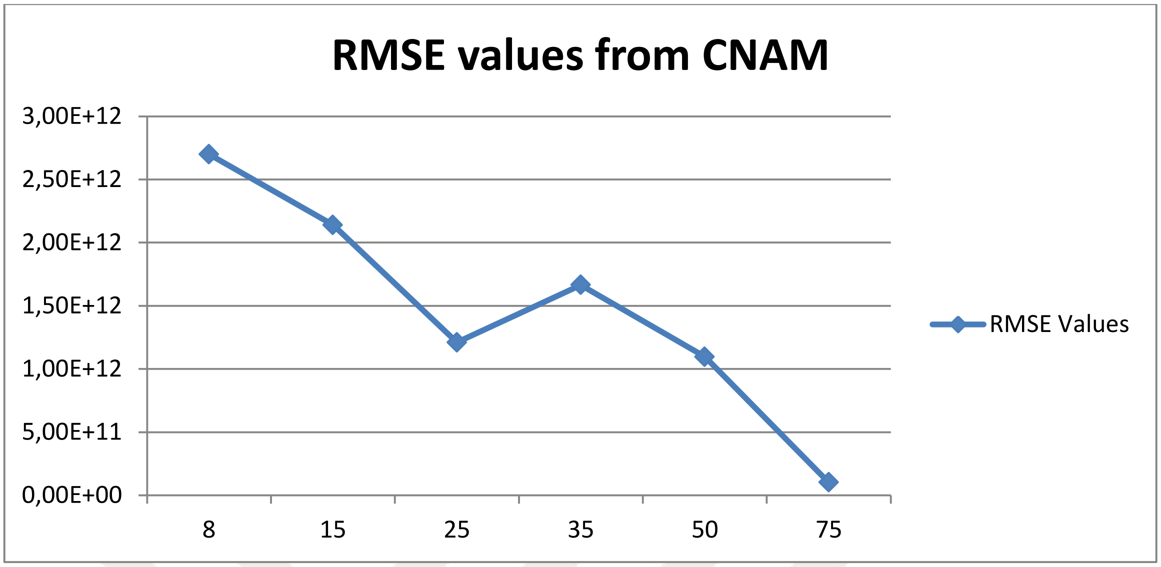

Figure 5.1: İllustration of the RMSE values for different latent variables count

As it is seen from Figure 5.1 CNAM gives the most accurate results when the latent variable number is set to 75. However, for the main experiments due computational constraints the number of hidden variable is fixed to 25.

5.3. Item Association Test

Under Section 4, we argue that using latent variables in CNAM, it is possible to cluster similar customers, similar items and shops. In order to show examples for this claim we manually analyzed items that are grouped using CNAM. Note that, items are associated through their relationships with similar needs. In this experiment, we used the same dataset acquired from Microsoft foodmart 2000 database [37]. Furthermore, in this experiment the value of latent variables is set as 25. Some examples of items that are related to the same latent variable are given in the tables (Need 15, Need 16, Need 2, and Need 18) below. As it is seen in “Need 16” Potato Chips, Beer and Salsa Dip are relevant items and they are clustered together using CNAM. Similarly, “Need 15” associated relevant items; Wine, Cheese, Cracker, and the same logic applies with other needs as well.

Need 15
Good Chablis Wine
Carlson Havarti Cheese
Good White Zinfandel Wine
Best Choice Sesame Crackers
Best Choice Beef Jerky

Need 16
Nationeel Potato Chips
Portsmouth Light Beer
Fast Salsa Dip
Walrus Light Beer
Atomic Bubble Gum

Need 2
Just Right Canned String Beans
High Top Shitake Mushrooms
Fort West Fondue Mix'
'Landslide Hot Chocolate'
'Thresher Mint Chocolate Bar'

Need 18
Excel Monthly Auto Magazine
Bird Call Silky Smooth Hair Conditioner
Red Wing Toilet Paper
Dollar Monthly Computer Magazine
Red Wing Plastic Knives

5.4. Main Experiments

5.4.1. Data Set

In our experiment the data set used is provided by KAGGLE 1. The whole data set is 22.1 GB, consisting of 350M lines of transaction, with 311,541 customers, 836 items and 134 chains. In the dataset, loyal customers have label one, otherwise, the value of the label is zero. As it was done for the original Kaggle contest, we conducted our experiments using only the loyal customers. Our subset includes the most active 40.331 customers, 826 items and 109 chains. For the main experiments, the data set is split into 4 equal parts such that first part is training and second third as well as fourth parts are test sets as shown in the Figure 5.2.

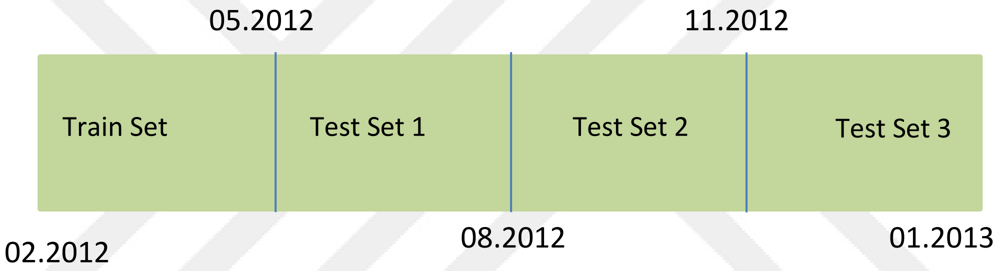

Figure 5.2: Visual descriptions of data splitting procedure

The reason for splitting the data set into 4 parts is to evaluate the accuracy of the predictions CNAM makes about near future and far future by training only with the first part of the dataset, the first three months of transactional data.

5.4.2. Experimental Steps

To validate CNAM, 10-fold cross validation technique is used. The main purpose of using cross validation method is to assess how accurately model can make predictions in practice.

Since 10 cross validation technique is used, first the data set is divided into 10 equal parts and for each part of the data set the following experimental steps are repeated;

- From the corresponding part of the dataset, choose randomly %10 of total existing item-shop (non-zero cells from the main item-shop matrix) pairs times as a test set
- Mask chosen pairs (test set) for the main item-shop matrix as shown in Figure 5.3. In other words, assign them to zero as they were never available in the corresponding shop for sale (like a new item)

In each iteration, 90% of the dataset is considered the training set, and 10% of the existing item-shop pairs are the test set. The following example Figure 5.3 illustrates the masking stage.

shop

item

0	0	0	0	0	7	0
0	5	5	6	0	5	0
1	3	0	0	1	0	1
0	0	8	5	0	1	0
0	7	0	0	4	0	1
2	0	1	1	2	0	2

Example of item-shop matrix original

shop

item

0	0	0	0	0	7	0
0	5	5	6	0	5	0
1	3	0	0	1	0	1
0	0	**0**	5	0	1	0
0	7	0	0	**0**	0	1
2	0	1	1	2	0	2

Masked item-shop matrix

Figure 5.3: Example of data masking technique

In Figure 5.3, there are 20 item-shop pairs %10 of pair number is equal to 2, therefore randomly masked pair number is 2 as in the Figure 5.3.

The masked data set is used as a training set for the CNAM and CF algorithms and randomly chosen item-shop pairs are used to assess both algorithms. Note that, because we have 3 different test sets we repeat experiments for each test set separately. After calculating predicted results for CNAM and CF algorithms, they are compared with the ground truth and *Spearman's rho* and RMSE scores are reported.

5.4.3. Evaluation Metrics

Results are measured using two different metrics; Spearman's rank correlation coefficient and Root Mean Square Error (RMSE).

1) Spearman's rank correlation coefficient:

For RS one of the most important key point is the system should be able to predict the order of items based on their preferableness. RSs order items with respect to the estimated scores or probabilities. To evaluate how well an algorithm orders the item recommendations, Spearman's rank correlation (Spearman's rho) coefficient is used. As shown in the Figure 5.4 the correlation between two variables will be high when two orderings are similar or it is 1 when orderings are perfectly same. Alternatively, correlation value 0 means when there is no relation between two orderings and -1 value interpreted if the two lists are in reverse order.

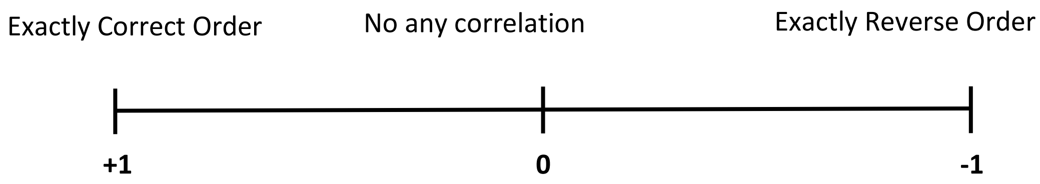

Figure 5.4: Spearman's rank correlation coefficient

Spearman correlation can be estimated as below (5.1):

$$\rho_{r_x,r_y}=\frac{cov(r_x,r_y)}{\sigma_{r_x}\sigma_{r_y}} \tag{5.1}$$

$cov(r_x, r_y)$: is the covariance of the rank variables.

$\sigma_{r_x}\sigma_{r_y}$: are the standard deviations of the rank variables.

2) Root Mean Square Error (RMSE):

CNAM and the ground truth are probability values. In order to evaluate accuracy of the forecasted probability values RMSE can be used. RMSE is a commonly used evaluation metric when the predicted and actual values are in the same unit. RMSE can be calculated as follows (5.2):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i}(p_i - \widehat{p_\imath})^2} \tag{5.2}$$

n: Number of observations

p_i: observed value for the ith observation

$\widehat{p_i}$: predicted value for the ith observation

5.4.4. Experimental Results

There are 3 different test sets, each consisting of 3 months of transactional data, as shown in Table 5.2. With the same training data CNAM and CF algorithms are tested with 3 different test sets as explained Section 5.4.1. In the Table 5.2, Average of 10-fold cross validation for 3 different tests set results with using Spearman rho metric are shown. The more detailed experimental results are presented in appendix A.

Test Set	**Average Spearman rho for CNAM**	**Average Spearman rho CF**
First	0.79732446641537325	0.697730968218773
Second	0.7388295990291394	0.6769041290869182
Third	0.7510716925351071	0.6563192904656319

Table 5.1: Comparison of CNAM with CF based algorithm with using Spearman’s rho metric

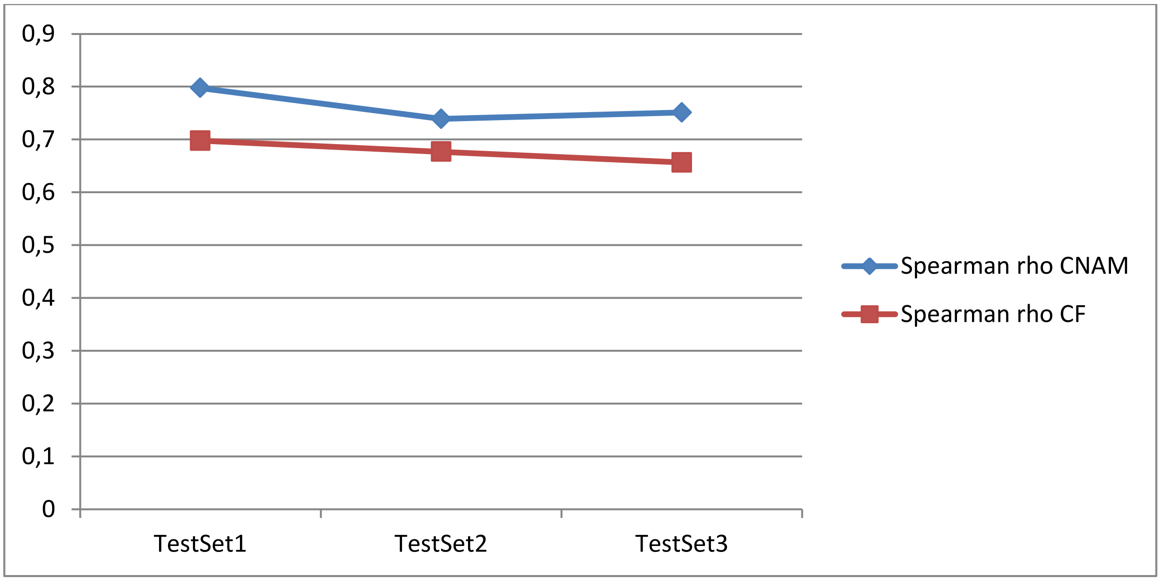

Figure 5.5: Graphical illustration of test results that are shown in the Table 5.2

CF algorithm gives only score value of the items for the corresponding shops and the ground truth based on probability therefore we could not calculate RMSE for CF algorithm. However, CNAM gives probability values so RMSE is calculated for each test using the ground truth values. Averages of RMSE values for three different test set are given in the Table 5.2. Detailed results for each iteration are indicated in the Appendix A.

Test Set	Average RMSE CNAM
First	0.003790666452304488
Second	0.003381344499353804
Third	0.0030966738148859433

Table 5.2 CNAM based RMSE values

In addition, as indicated in the Table 5.3 CNAM can estimate probabilities of items accurately with a low error rate. It can be concluded from the RMSE results that we validated our CNAM one more time with using different metric.

5.4.5. Discussion of the Main Experimental Results

To better examine the experimental results an important point should be clarified first. To evaluate the CNAM we used 2 different evaluation metrics; Spearman rho and RMSE. Each metric assesses the CNAM generated results in a different way, such that; while Spearman rho tries to evaluate how well we order the items based on their sale probability, RMSE considers how accurate CNAM calculates the probabilities.

From the conducted experiments the most important findings can be specified as follows;

- Spearman rho results in Table 5.2 can be interpreted as CNAM showing reasonable performance for the function of ordering items based on their tendency of being preferred which is the core idea of RSs
- As it is seen from Table 5.2 and Figure 5.5, CNAM outperforms the CF algorithm for this task.
- By observing the results shown in Table 5.2, the best forecasting of item preferences is made for the near future as expected.

- CNAM can make reasonable predictions about customer behavior with a very low RMSE rate
- It can be seen from Table 5.3. The error rates are getting lower for the more distant future predictions. However, the difference of RMSE for each test set is very small (less then 10^{-3} , can be tolerable for RS tasks). Therefore, we can interpret those RMSE results as CNAM can make predictions for new item sale rate with a probability.
- Both evaluation metrics gives similar results for the 3 test sets. This can be interpreted as for the similar tasks training with only 3 months of the transactional data we can predict sales rate of approximately rest of the year.

From the overall results, we can easily conclude that CNAM is an appropriate model for new item recommendation system, generating more accurate results compared to commonly used CF algorithm.

6. CONCLUSION AND FUTURE WORK

In this competitive world, it became necessary for sellers to introduce new items to their customers. However, deciding which item to introduce to their customers is not an easy task for sellers. Therefore, in this thesis to help sellers we build a new RS. The main function of the new RS is to recommend new items to sellers not available in the shop previously. To build the proposed RS first customer behavior is modeled using a probabilistic generative model and predictions are made for his/her future preferences. To model customer behavior for predicting future preferences we used 3 observable variables; customer, customer-shop and customer-item preference and an unobservable-latent variable called “need”. In consideration of building the model, an early probabilistic topic model Probabilistic Latent Semantic Analysis (pLSA) was extended, because, the original pLSA analyzes 2 observable variables with an unobservable variable. The RS which was built in this thesis is called “Customer Need Aspect Model (CNAM)”. To evaluate the CNAM real world datasets are used. In order to compare the performance of CNAM, CF algorithm was used as a baseline algorithm. Our experiments yield that CNAM outperformed CF as a new item RS.

One of the very important conclusions that can be drawn by analyzing experimental results under the Section 5.4.4 to predict item sale rate for a given shop or recommend item/s to shops considering not only similar shops but also similar customer profile has high impact on the accuracy of the task. Because, CF algorithm generated item recommendations are based on only shops similarity. However, CNAM relies on customer similarity as well as shops similarity.

As it is described in Section 4; the purpose of using latent variable is to reveal the underlying relation of observable variables. The experimental results (Section 5.4.4), proved that hidden variable successfully conducted its duty for this item recommendation task as well.

Last but not least, as described CNAM is a probabilistic generative model, it is very flexible. In other words by only applying basic Bayes rule different probabilities can be calculated very easily. For example, in Section 5.3, similar items are listed under the same latent variables. In other words those items are mostly associated with the same latent variable. This can be interpreted mathematically as probability of an item given need $\sim P(i|n)$. This value was estimated by applying Bayes rule to the joint probability of the CNAM.

In summary we proposed a topic model based algorithm for RS, an idea which is only recently being explored. While there are some LDA based topic models proposed in the literature for transactional data, our work seems to be the first method that extends pLSA for the problem. While the addition of shop observable variable to the model enables us to recommend items to sellers, we also believe that the shop also groups customers with additional features yielding better results.

As for future work, to build CNAM we only considered customer, item, and shop as data sources. For the future, CNAM can be improved by integrating more data sources such as;

- Time,
- Customer demographic information,
- Price.

Time is an important constraint for designing a RS because customer preferences or needs can change based on time. Further, while modeling customer behavior, customer demographic information can also enhance the RS efficiency. Price is also an important factor because some customers give up buying some items just because the price is slightly higher. For the future research, we expect CNAM to be improved by using more data source.

Further, for future improvements LDA can also be used for modeling customer behavior. Because, LDA provides very flexible framework especially in regards of increasing or decreasing number of variables.

7. APPENDICES

Appendix A: 10 Fold Cross Validation Experimental Results

Iteration Number	**Spearman’s correlation value for CNAM**	**Spearman’s correlation value for CF algorithm**
0	0.7560975609756097	0.4878048780487804
1	0.7818181818181819	0.6
2	0.9515151515151515	0.793939393939394
3	0.7454545454545455	0.7333333333333333
4	0.8666666666666667	0.7696969696969697
5	0.8181818181818182	0.793939393939394
6	0.9151515151515152	0.7696969696969697
7	0.7333333333333333	0.696969696969697
8	0.5757575757575758	0.5636363636363636
9	0.8292682926829267	0.7682926829268292
Average	0.7973244641537325	0.697730968218773

Table 7.1: 10 fold cross validation results for the each iteration is represented for the first test set

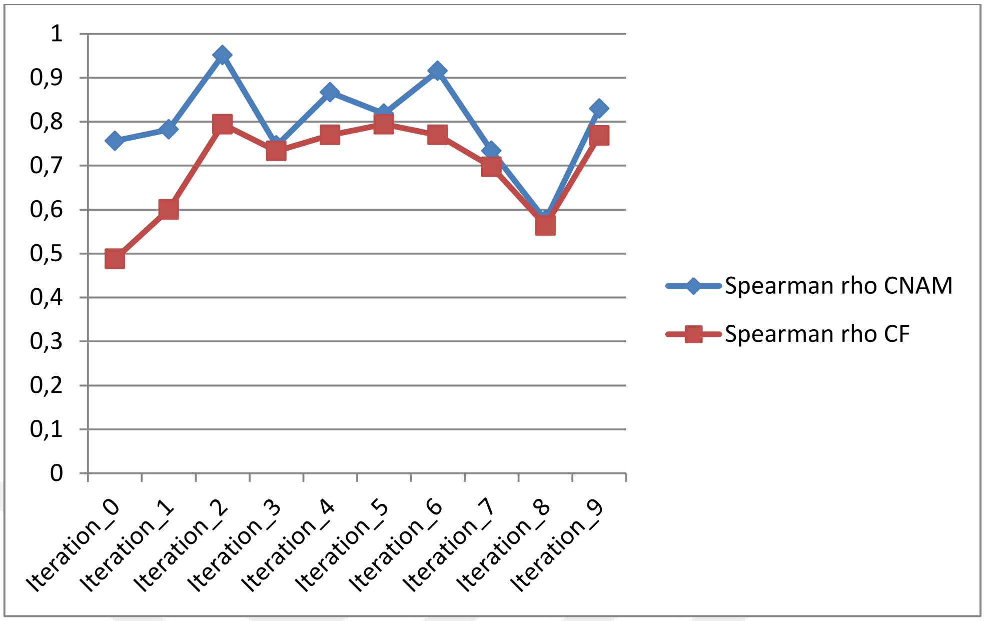

Figure 7.1 Graphical representation of 10 fold cross validation results for the first data set

Iteration Number	Spearman’s correlation value for CNAM	Spearman’s correlation value for CF algorithm
0	0.573170731707317	0.451219512195122185
1	0.7575757575757576	0.7818181818181819
2	0.9393939393939394	0.8424242424242424
3	0.7818181818181819	0.793939393939394
4	0.8666666666666667	0.7696969696969697
5	0.9515151515151515	0.8545454545454545
6	0.8787878787878788	0.7212121212121212
7	0.7333333333333333	0.696969696969697
8	0.5757575757575758	0.5636363636363636
9	0.33027677373559056	0.293579354431636
Average	0.7388295990291394	0.6769041290869182

Table 7.2: 10 fold cross validation results for the each iteration is represented for the second test data set

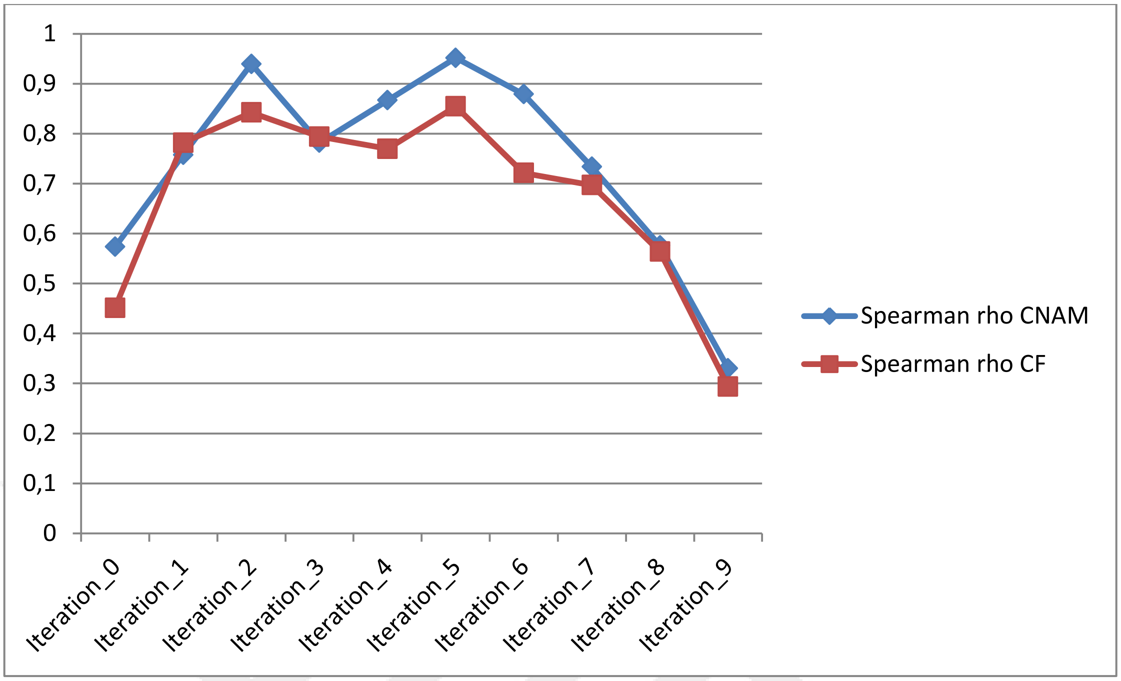

Figure 7.2: Graphical representations of 10 fold cross validation results for the second data set

Iteration Number	**Spearman's correlation value for CNAM**	**Spearman's correlation value for CF algorithm**
0	0.6341463414634145	0.41463414634146334
1	0.6121212121212121	0.6121212121212121
2	0.8666666666666667	0.9030303030303031
3	0.6727272727272727	0.6484848484848484
4	0.8181818181818182	0.6727272727272727
5	0.9030303030303031	0.6848484848484848
6	0.8303030303030303	0.7090909090909091
7	0.8909090909090909	0.7818181818181819
8	0.6484848484848484	0.6242424242424243
9	0.6341463414634145	0.5121951219512194
Average	0.7510716925351071	0.6563192904656319

Table 7.3: 10 fold cross validation results for the each iteration is represented for the third test data set

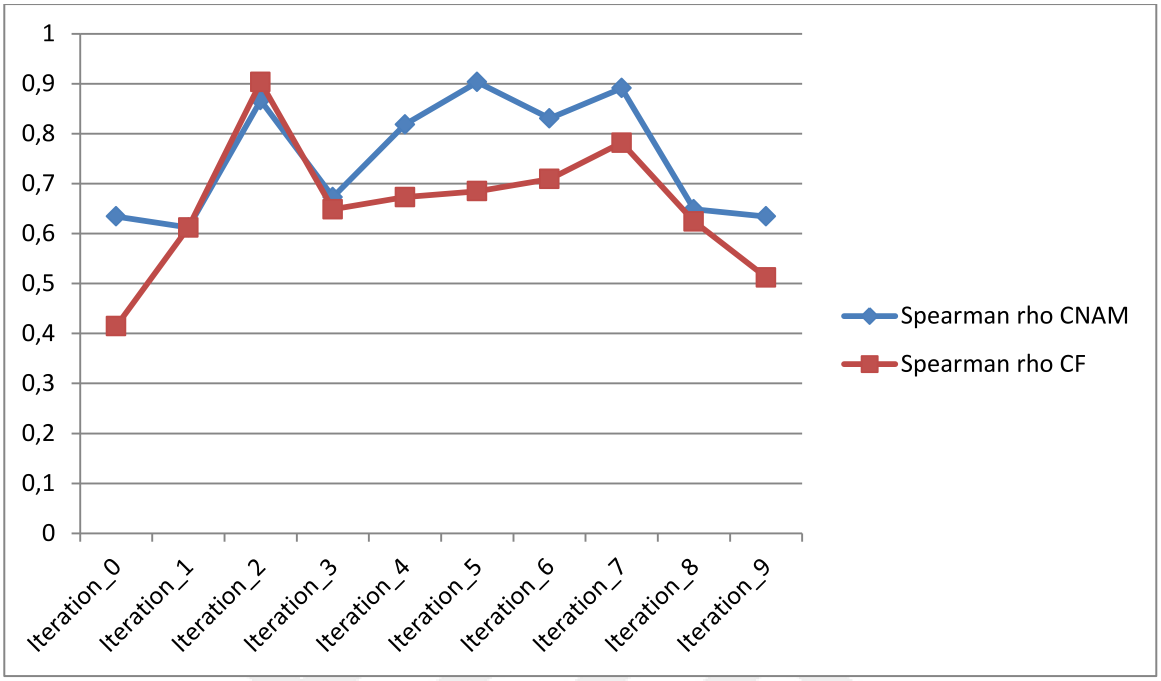

Figure 7.3: Graphical representation of 10 fold cross validation results for the third data set

Iteration Number	RMSE for CNAM TestSet1	RMSE for CNAM TestSet2	RMSE for CNAM TestSet3
0	0.013435137332314412	0.012725450974184947	0.010244051016899637
1	9.348327025131913E-4	7.858501124807442E-4	8.381659695009728E-4
2	0.0019554066384963515	0.0016863874044996036	0.0017535623383174478
3	6.412937862610495E-4	7.399098091495341E-4	7.149651830902882E-4
4	0.0010830427099941479	0.0016915759237591129	0.0013899011735341057
5	0.003136316034502775	9.206515949567133E-4	0.0011317648196196128
6	6.358816996243802E-4	0.001723881190110082	0.0011385601957332235
7	0.006491252309824907	0.005697649338463903	0.005891061246493221
8	0.00943663381898981	0.007659749279777768	0.007695578417162402
9	1.568674905238461E-4	1.8233936615563452E-4	1.6912778850851966E-4
Average	0.003790666452304488	0.003381344499353804	0.0030966738148859433

Table 7.4: RMSE values obtained from each test set

REFERENCES

[1] T. Belluf, L. Xavier, and R. Giglio, “Case study on the business value impact of personalized recommendations on a large online retailer,” *RecSys 2012 Proc. 6th ACM Conf. Recomm. Syst.*, pp. 277–280, 2012.

[2] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender Systems Handbook*, vol. 53, no. 9. 2011.

[3] J. Leskovec, R. Anand, and J. Ullman, “Recommendation Systems,” *Min. Massive Datasets*, pp. 305–339, 2011.

[4] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, “Collaborative Filtering Recommender Systems,” *Found. Trends Human-Computer Interact.*, vol. 4321, no. 1, pp. 291–324, 2007.

[5] G. D. Clifford, “Singular Value Decomposition & Independent Component Analysis for Blind Source Separation,” *Biomed. Signal Image Process.*, p. 49, 2005.

[6] J. Yang, M. Luo, and Y. Jiao, “Face Recognition Based on Image Latent Semantic Analysis Model and SVM,” vol. 6, no. 3, pp. 101–110, 2013.

[7] C. D. Manning, P. Raghavan, and H. Schütze, “Chapter 18: Matrix decompositions and latent semantic indexing,” *Introd. to Inf. Retr.*, no. c, pp. 403–419, 2008.

[8] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Incremental Singular Value Decomposition Algorithms for Highly Scalable Recommender Systems,” *Fifth Int. Conf. Comput. Inf. Sci.*, pp. 27–28, 2002.

[9] T. Bogers and A. Van Den Bosch, *Collaborative and content-based filtering for item recommendation on social bookmarking websites*, vol. 532. 2009.

[10] L. Lü, M. Medo, C. Ho, Y. Zhang, and Z. Zhang, “Recommender systems,” vol. 519, pp. 1–49, 2012.

[11] A. Alluhaidan, “Recommender System Using Collaborative Filtering Algorithm,” 2013.

[12] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang, and R. Zadeh, “WTF: The Who to Follow Service at Twitter,” *WWW 2013 Proc. 22nd Int. Conf. World wide web*, pp. 505–514, 2013.

[13] S. Sivapalan, A. Sadeghian, H. Rahnama, and A. M. Madni, “Recommender systems in e-commerce,” *World Autom. Congr. Proc.*, pp. 179–184, 2014.

[14] T. Bogers and A. Van Den Bosch, "Collaborative and content-based filtering for item recommendation on social bookmarking websites," in *CEUR Workshop Proceedings*, 2009, vol. 532, pp. 9–16.

[15] Y. Shoham, "mmende tems," *Commun. ACM*, vol. 40, no. 3, 1997.

[16] D. M. Blei and J. D. Lafferty, "Topic Models," *Text Min. Classif. Clust. Appl.*, pp. 71–89, 2009.

[17] S.-Y. Kong and L. Lee, "Improved spoken document summarization using probabilistic latent semantic analysis (plsa)," *Acoust. Speech Signal Process. 2006. ICASSP 2006 Proceedings. 2006 IEEE Int. Conf.*, vol. 1, pp. 941–944, 2006.

[18] Y. Akita, Y. Nemoto, and T. Kawahara, "PLSA-based topic detection in meetings for adaptation of Lexicon and language model," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2, no. 1, pp. 1321–1324, 2007.

[19] T. Iwata and H. Sawada, "Topic model for analyzing purchase data with price information," *Data Min. Knowl. Discov.*, vol. 26, no. 3, pp. 559–573, 2013.

[20] T. Iwata, S. Watanabe, T. Yamada, and N. Ueda, "Topic Tracking Model for Purchase Behavior Analysis," no. 6, pp. 978–987.

[21] F. Sun, M. Griss, and O. Mengshoel, "Latent Topic Analysis for Predicting Group Purchasing Behavior on the Social Web."

[22] L. T. Data, "Topic Modeling of Market Responses for Large-Scale Transaction Data," no. 35, 2015.

[23] K. Christidis, D. Apostolou, and G. Mentzas, "Exploring Customer Preferences with Probabilistic Topics Models," pp. 1–13.

[24] M. Giering, "Retail Sales Prediction and Item Recommendations Using Customer Demographics at Store Level," *SIGKDD Explor.*, vol. 10, no. 2, p. 6, 2008.

[25] D. Blei, L. Carin, and D. Dunson, "Probabilistic topic models," *IEEE Signal Process. Mag.*, vol. 27, no. 6, pp. 55–65, 2010.

[26] M. Steyvers, P. Smyth, and C. Chemuduganta, "Combining background knowledge and learned topics," *Top. Cogn. Sci.*, vol. 3, no. 1, pp. 18–47, 2011.

[27] T. Hofmann, "Probabilistic Latent Semantic Analysis," *Uncertainity Artifitial Intell. - UAI'99*, p. 8, 1999.

[28] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet Allocation,” *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 993–1022, 2003.

[29] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” *Proc. 10th …*, vol. 1, pp. 285–295, 2001.

[30] T. Hofmann, “Collaborative filtering via gaussian probabilistic latent semantic analysis,” *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Dev. informaion Retr. - SIGIR ’03*, no. v, p. 259, 2003.

[31] M. Steyvers and T. Griffiths, “Probabilistic Topic Models.”

[32] B. Jin, W. Hu, and H. Wang, “Image Classification Based on pLSA Fusing Spatial Relationships Between Topics,” vol. 19, no. 3, pp. 151–154, 2012.

[33] X. Jin, Y. Zhou, and B. Mobasher, “Web Usage Mining Based on Probabilistic Latent Semantic Analysis,” 2004.

[34] C. B. Do and S. Batzoglou, “What is the expectation maximization algorithm ?,” vol. 26, no. 8, pp. 897–899, 2008.

[35] C. Science, “Probabilistic Latent Semantic Analysis,” no. 2, pp. 1–13, 2012.

[36] P. Nagarnaik and a. Thomas, “Survey on recommendation system methods,” *2nd Int. Conf. Electron. Commun. Syst. ICECS 2015*, vol. 137, no. 7, pp. 1603–1608, 2015.

[37] FoodMart2000, Microsoft Developer Network (MSDN), http://msdn.microsoft.com/en-us/library/aa217032(v=sql.80).asp

CURRICULUM VITAE

Credentials

Name, Surname:	Rima Al Washahi
Place of Birth:	Ankara
Marital Status :	Single
E-mail :	rimaturker@gmail.com
Address :	ETA Elektronik, ODTU Ikizleri, B Blok Z kat ODTU-Teknokent 06800,Ankara

Education

High School :	Kaya Bayazıtoğlu High School
BSc. :	Çankaya University, Computer Engineering
MSc. :	Hacettepe University, Computer Engineering

Foreign Languages Turkish, English, Arabic

Work Experience ETA, more than 3 years as a Software Engineer

Areas of Experiences Computer Science and Software Engineering

Projects and Budgets

Publications

Türker R, Ercan G. “Topic Model Based Recommendation Systems For Retailers” *UBMK 2016*

Oral and Poster Presentations