THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY

A POLYNOMIAL MODELING BASED ALGORITHM IN TOP-N RECOMMENDATION

Ph.D. Thesis

ÖZGE YÜCEL KASAP

İSTANBUL, 2018

THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING

A POLYNOMIAL MODELING BASED ALGORITHM IN TOP-N RECOMMENDATION

Ph.D. Thesis

ÖZGE YÜCEL KASAP

Supervisor: ASSOC. PROF. DR. M. ALPER TUNGA

İSTANBUL, 2018

THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY
The Graduate School of Natural and Applied Sciences
Computer Engineering

Title of the Ph.D. Thesis	:	A Polynomial Modeling Based Algorithm in Top-N Recommendation
Name/Last Name of the Student	:	Özge YÜCEL KASAP
Date of Thesis Defense	:	January 25, 2018

The thesis has been approved by The Graduate School of Natural and Applied Sciences.

Prof. Dr. Nafiz Arıca
Acting Director

I certify that this thesis meets all the requirements as a thesis for the degree of Doctor of Philosophy.

Assist. Prof. Dr. Tarkan Aydın
Program Coordinator

This is to certify that we have read this thesis and that we find it fully adequate in scope, quality and content, as a thesis for the degree of Doctor of Philosophy in Computer Engineering Department.

Examining Commitee Members:	Signature
Assoc. Prof. Dr. M. Alper Tunga (Supervisor)	: .....
Prof. Dr. Adem Karahoca	: .....
Assoc. Prof. Dr. Ahmet Kırış	: .....
Assist. Prof. Dr. Tevfik Aytekin	: .....
Assoc. Prof. Dr. Devrim Ünay	: .....

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisor Assoc. Prof. Dr. M. Alper Tunga for his unprecedented academic guidance, advice and encouragement. Without his patience and guidance, this thesis would not have been possible to accomplish.

Secondly, I would like to express my immense gratitude to Taha Yasin Toraman, who provided me the necessary data for this thesis. Considering how hard it is to gather the data from industry, I am really thankful to them.

My special thanks to my friends, especially Efsun Karaca and Ertunç Erdil for their comments, suggestions and supports all through this thesis.

I am thankful to the Scientific and Technological Research Council of Turkey (TÜBİTAK) for providing me financial support throughout my PhD study with the BİDEB2211 programme.

Finally, I would also like to thank my parents who helped me a lot in finalizing this thesis within the limited time frame, especially my husband Hakan Kasap for his patience and support.

İstanbul, 2018 Özge YÜCEL KASAP

ABSTRACT

A POLYNOMIAL MODELING BASED ALGORITHM IN TOP-N RECOMMENDATION

Özge Yücel Kasap

Computer Engineering
Supervisor: Assoc. Prof. Dr. M. Alper Tunga

January 2018, 59 Pages

Recommendation is the process of identifying and recommending items that are more likely to be of interest to a user. Recommender systems have been applied in variety of fields including e-commerce web pages to increase the sales through the page by making relevant recommendations to users. In this thesis, we pose the problem of recommendation as an interpolation problem, which is not a trivial task due to the high dimensional structure of the data. Therefore, we deal with the issue of high dimension by representing the data with lower dimensions using High Dimensional Model Representation (HDMR) based algorithm. We combine this algorithm with the collaborative filtering philosophy to make recommendations using an analytical structure as the data model based on the purchase history matrix of the customers. The proposed approach is able to make a recommendation score for each item that have not been purchased by a customer which potentiates the power of the classical recommendations. Rather than using benchmark data sets for experimental assessments, we apply the proposed approach to a novel industrial data set obtained from an e-commerce web page from apparels domain to present its potential as a recommendation system. We test the accuracy of our recommender system with several pioneering methods in the literature. The experimental results demonstrate that the proposed approach makes recommendations that are of interest to users and shows better accuracy compared to state-of-the-art methods.

Keywords: Recommender Systems, Purchase History Matrix, HDMR, e-commerce

ÖZET

İlk-N Tavsiye Sisteminde Polinom Modelleme Tabanlı Algoritma

Özge Yücel Kasap

Bilgisayar Mühendisliği
Tez Danışmanı: Doç. Dr. M. Alper Tunga

Ocak 2018, 59 Sayfa

Tavsiye, bir kullanıcı için daha fazla ilgi çekici olan öğeleri tanımlama ve önerme işlemidir. Tavsiye sistemleri, e-ticaret web sayfaları da dahil olmak üzere çeşitli alanlarda, kullanıcılara ilgili öneriler yaparak satışları artırmak için kullanılmaktadır. Bu tezde, verilerin yüksek boyutlu yapısından dolayı oldukça zorlu bir işlem olan tavsiye oluşturmayı bir interpolasyon problemi olarak ortaya koymaktayız. Dolayısıyla, Yüksek Boyutlu Model Gösterilim (YBMG) tabanlı algoritma kullanılarak, verileri daha düşük boyutlarla temsil ederek yüksek boyut sorunu ile ilgileniyoruz. Müşterilerin satın alma geçmişi matrisine dayanan veri modelini baz alan analitik bir yapı kullanarak tavsiyelerde bulunmak için, bu algoritmayı işbirlikçi filtreleme felsefesiyle birleştirdik. Önerilen bu yaklaşım, bir müşteri tarafından satın alınmamış her bir öğe için bir öneri puanı verebilmekte, bu da klasik tavsiye sistemlerinin gücünü arttırmaktadır. Deneysel değerlendirmeler için kıyaslama veri kümeleri kullanmak yerine, önerilen yaklaşımın bir tavsiye sistemi olarak potansiyelini ortaya koymak için, hazır giyim alanındaki bir e-ticaret web sayfasından elde edilen ve daha önce hiç bir akademik çalışmada kullanılmamış olan özgün bir endüstriyel veri kümesi kullanılmıştır. Tavsiye sistemimizin doğruluğunu, literatürde bulunan birkaç öncü yöntemle test ettik. Deneysel sonuçlar, önerilen yaklaşımın kullanıcıların ilgi alanına giren tavsiyeler sunduğunu ve en yeni yöntemlere kıyasla doğruluk ve tahmin gücü bakımından daha iyi olduğunu göstermektedir.

Keywords: Tavsiye Sistemleri, Satın Alma Geçmişi Matrisi, YBMG, e-ticaret

CONTENTS

TABLES

FIGURES

ABBREVIATIONS

HDMR	:	High Dimensional Model Representation
IHDMR	:	Indexing High Dimensional Model Representation
IDE	:	Integrated Development Environment
IBCF	:	Item-Based Collaborative Filtering
KNN	:	K-Nearest-Neighbor
MAE	:	Mean Absolute Error
MS	:	Microsoft
R-HDMR	:	Recommender HDMR
RMSE	:	Root Mean Square Error
SQL	:	Structured Query Language
UBCF	:	Used-Based Collaborative Filtering

SYMBOLS

Cartesian Product Set	:	$\mathcal{D}$
Class Information	:	φ
Direc Delta	:	$\delta(x)$
Independent Variable	:	x
Index Node	:	ξ
Normalized Value	:	z
Number Of Attributes	:	N
Number Of Nodes	:	m
Prediction Error	:	e
Prime Factor	:	n
Product Type Weight	:	$W(x_1, \ldots, x_N)$
Purchase History Matrix	:	C
Purchase Quantities Vector	:	Q
Training Node	:	v
Vector	:	$\vec{A}$,$\vec{B}$

1. INTRODUCTION

In recent years, for their usage on e-commerce Web sites, recommendation systems have become extremely common. Among a set of existing choices, recommendation systems can help people to recognize their interests (Véras et al., 2015). They use user behaviors such as items purchased and numerical ratings given to those items. Almost every day, when we are at work, at home or on the way to somewhere, we use e-commerce sites to purchase things. These e-commerce web sites use recommendation systems for multiple reasons like to attract the users, increase the sale amount or to come in first in the market. It is really challenging to find the information people need because of the information overload on the web (Polatidis & Georgiadis, 2016). As a result of this, many people find it annoying to look online what they need. Recommendation systems have changed the way people find information, products, and even other people. They study patterns of behavior to know what someone will prefer from among a collection of things that have never experienced by this person. These systems can be considered as social navigation as well, following in the footsteps of others to find what you want. The concept of social navigation more generally gets into the idea of social information reuse which states that we can learn from each other. Most recommendation algorithms use this logic to model the data and find a set of customers whose purchased and rated items overlap the user's purchased and rated items. Nowadays, many companies use tools named "automatic recommendation systems ". These tools can be classified as decision tools. They attempt to analyze purchase history of a customer and try to identify items the customer may buy in the future. Most of researches today, attempt to recommend products using ratings data model and a limited number of researches uses binary data model. In software engineering, forming a data model by using some basic modeling techniques for an information system is known as data modeling. Because of the computational complexities, if the data has higher dimensions, it becomes much more difficult to find and represent an analytical structure with standard interpolation techniques. When finding solutions for different areas of engineering problems, there are a lot of methods that can be used. Using standard

interpolation methods is just one of them but it can cause some numerical problems when the dimensionality increases. When the issue is the curse of dimensionality, it impulses scientist to develop a divide-and-conquer methods. High Dimensional Model Representation (HDMR) is an competent technique which is constructed by decomposing a multivariate function into a constant term, N number of univariate components and N(N-1)/2 number of bivariate components and so on. There are various HDMR-based methods that other scientists developed for different research areas. According to the structure of the given problem, each HDMR method can have different technical problems. In this thesis, we propose a hybrid recommendation system approach that is based on HDMR and collaborative filtering. This new algorithm is called "Recommender HDMR (R-HDMR)". In our approach, we store items that have been purchased by each customer together with the purchase quantity of each item. Given a customer that is targeted for recommendation, the target customer, we find the similar customers using collaborative filtering philosophy. Then, we fit a model to the data that keep the purchase history of the target customer and the most similar customers using Recommender HDMR with Lagrange interpolation. Given an item that has not been purchased by the target customer, the motivation of using interpolation is to estimate the purchase quantity of this item for a particular customer. Then, we recommend top-N items that have the highest purchase quantities to the customer. However, interpolating to a high dimensional data is not a trivial task due to the computational inefficiency of the interpolation methods (Tunga & Demiralp, 2009) which we solve by using HDMR philosophy (Tunga & Demiralp, 2009; Tunga, 2011). Eventually, the main aim of this research is to use Recommender HDMR for making categorical recommendations based on a specific time period.

The main objective of this thesis is to develop a reliable recommendation system that can uncover the unseen products for the target user that the user will probably find interesting. Secondly, we are interested in measuring how the state-of-the-art recommendation algorithms perform with respect to our approach. The challenge, however, various recommendation systems have been developed nowadays for different domains. Therefore, it is necessary to build a high quality recommendation system for more accurate results.

Another important challenge is that we are trying to model human behavior which is very complex. It is particularly more complex when the issue is recommending something to a customer.

To this end, the major contributions of this thesis is three-fold. First, we propose an interpolation-based recommendation system that exploits the collaborative filtering and the Indexing HDMR method. Second, to the best of our knowledge, interpolation-based recommendation systems have not been proposed in the literature. Third, we apply the proposed method to a novel data set that have not been used prior to this work for recommendation.

2. LITERATURE REVIEW

This chapter examines the literature and refers to the mostly used and known recommendation systems in the academic domain.

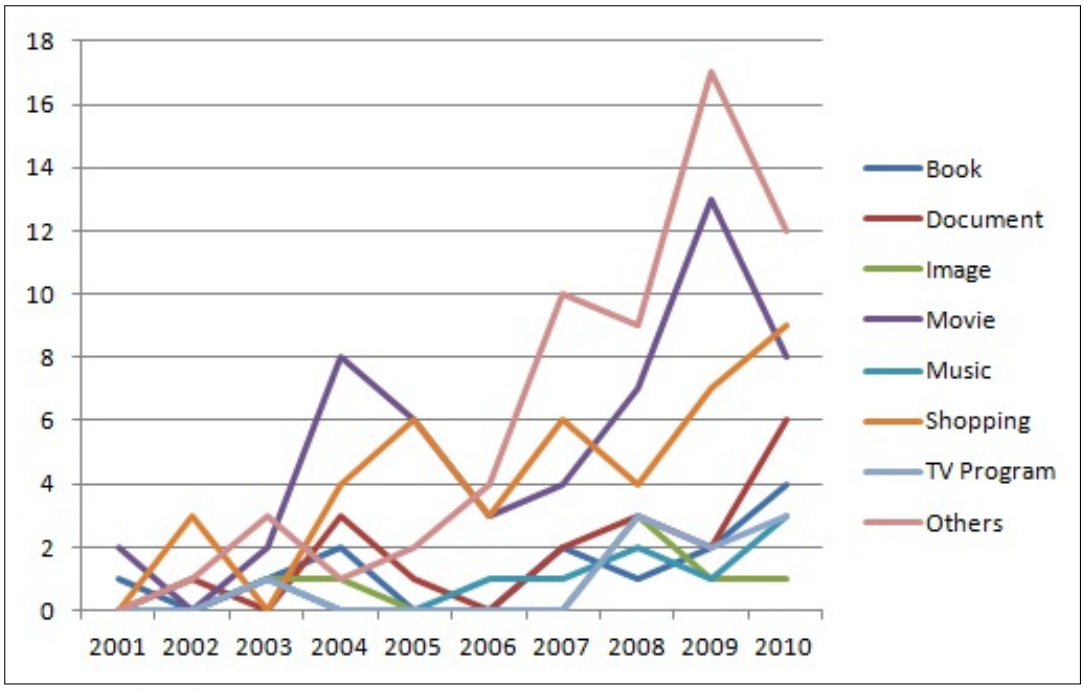
In the literature, several types of recommendation systems have been proposed that are differ from each others in terms of the data that have been applied and the underlying method used to generate recommendations (Campos et al., 2014). Tapestry (Resnick et al., 1994) is the first recommendation system designed to recommend documents from newsgroups. The thesis also introduces the term collaborative filtering as they use social collaboration to help users with large volume of documents. Konstan and Riedl state that there are a lot of different ways to get recommendations (Konstan & Riedl, 2012).

In 1990s, after the first collaborative filtering paper has been published, recommendation systems became a very significant research area (Resnick et al., 1994). As mentioned before, RSs help people to find interesting or helpful contents according to their desire basically using data mining algorithms. Since the first attempt of developing a RS, engineers try to overcome some difficulties or problems there system have.

Figure 2.1 displays the distribution of recommendation system research papers by publication year and the corresponding application fields. AS it can be seen from the figure, there is decrease in nearly all of the fields around 2006. Despite that, starting from 2007 to until today, the researches have started to be extended especially in the fields of movie and shopping.

Since collaborative filtering is generally used to generate recommendations, when we look at the literature we can see several examples of it. GroupLens uses a big database of news to recommend articles to users using collaborative methods (Park et al., 2012). Ringo which is a social information filtering for music recommendation, also uses collaborative filtering approach to build recommendations based on the user ratings for music albums (Chen et al., 2008). Amazon, which is one of the biggest e-commerce web site, developed

Figure 2.1: Distribution of recommendation system research papers by publication year and application fields



Source: (Park et al., 2012)

its own recommendation system by generating a table of similar items offline and makes an online recommendation using this table and users' purchase history matrix which is a nxm matrix where n expresses the number of customers and m represents the number of items. Values in this purchase history matrix can represent the number of purchases, ratings or it can be either 0 (unliked/not purchased) or 1(liked/purchased). The structure of this matrix can be changed according to the problem definition. The structure of the purchase history matrix used in this thesis is explained in detail in chapter 3.

When it comes to content-based recommendations, Letizia tries to predict web-pages which might be interesting for a target user by tracking the browsing pattern (Lieberman et al., 1995). Using the naive Bayesian classifier, Pazzani et al (Pazzani, 1999) developed an agent to forecast different web pages that can awaken user's interest. This agent grants the users to give a rating to several different web pages while creating the recommendations.

To avoid several limitations of these two filtering approach, a hybrid prediction technique

was developed by Ghazantar and Pragel-Benett (Ghazanfar & Prugel-Bennett, 2010). I this method, user profiles are used to find similar users to make recommendations. Another example is from an information filtering agent, which was combined with collaborative filtering to create a hybrid framework (Sarwar et al., 1998). By Cunningham et al. a sophisticated yet simple method was proposed by combining content-based and collaborative filtering approaches(Cunningham et al., 2001). Konstas et al. developed a music recommendation system by integrating number of plays, social relationships and the tagging information (Konstas et al., 2009). Lee and Brusilovsky planted social information into collaborative filtering approach to discover neighbor number that can be automatically connected on a social platform (Lee & Brusilovsky, 2010). Condiff et al. introduced a framework using Bayesian method that compounds user ratings and item features (Condli et al., 1999).

When we look at the real-world recommendation systems, which are frequently cited by the academic domain, MovieLens, LIBRA and Dooyoo are the popular three ones. Table 2.1 summarizes the popular recommendation systems, the algorithms they use, their features, advantages and disadvantages.

In order all of these systems to work effectively, the key point is data. These systems cannot function accurately unless user profiles or recommendation models are well constructed. The user profiles represent interests and preferences and they allow users to be modelled. The system needs to get as mush as from the user. For reasonable recommendations, each system can rely on different type of data. The data can be gathered explicitly, implicitly or as hybrid feedback.

In explicit data collection, the system asks the user to give ratings for items. The quality of these ratings directly affect the recommendation accuracy. Even though this type of feedback does not extract preferences from user actions, it requires great amount of effort from the users which results having more reliable data (Buder & Schwind, 2012). On the other hand, the implicit systems automatically gathers preferences from user actions like click actions, time spend on the system, browse history, purchase history and so on.

Table 2.1: Features of popular recommendation systems

System	Method	Features	Pro/Con
MovieLens	Collaborative filtering	Asks the user to rate movies, finds similar profiles, uses stochastic and heuristic methods for profile matching	Easy to explain how recommendations are populated
LIBRA	Content-based filtering and machine learning	Uses Bayesian text-categorisation machine learning techniques to build models of user preferences relative to a specific item	Easily produce explanations, inappropriate to non-textual items like images or video
Pandora	Deep item analysis	Represents user preferences as collection of items	Low cost of entry for the user
Amazon	Personalised, social and item based approach	Recommendations are based on items other users purchased	Aims to add more items to users shopping cart, overcomes the cold-start problem
Dooyoo	Hybrid system	Qualitative opinions are taken from users, displays the result like search engines, creates similar user groups	Easy to understand, requires each item to be reviewed and rated

Source: (Fournier, 2011)

Despite the fact that implicitly collected data decreases the user effort, it is less reliable. To combine the advantages and minimize the weaknesses of these methods, hybrid feedback collection is always an option.

When we think about the real world data, it is very huge and sparse. Working with this

kind of data, brings up some problems like dimension reduction. Most recommendation systems represent customers and items as vectors and generates a matrix accordingly. Since in real world the number of customers and items are huge, this matrix will be huge and sparse too. To analyze and organize such data is expensive and not easy. The main aim of this thesis is to use HDMR philosophy to make personalized recommendations based on the purchase history of a customer by constructing the analytical structure of the data.

The other leg of this thesis is implementing the HDMR philosophy to make recommendations. HDMR is a mathematical based divide and conquer algorithm and its structure can be affected by the data set. HDMR is an competent technique which is constructed by decomposing a multivariate function consisting of N independent variables into less variate functions starting with a constant term and followed by $2^N - 1$ number of terms with increasing number of independent variables. In 1993, the first HDMR method was proposed by I.M. Sobol (Sobol, 1993). Following Sobol's work, many HDMR based methods were developed by Rabitz Rabitz & Aliş (1999); Alış & Rabitz (2001); Li et al. (2002b) and each of them were used for different purposes such as data modeling (Tunga & Demiralp, 2008; Tunga, 2011; Tunga & Demiralp, 2012b), weight and parameter optimization (Demiralp & Tunga, 2015; Tunga & Demiralp, 2012a), parallelization (Kanal & Demiralp, 2012), sensitivity and reliability analysis (Cooling et al., 2016; Fang et al., 2015; Balu & Rao, 2014) and approximation (Özay & Demiralp, 2014; Li et al., 2015; Li & Rabitz, 2014, 2012; Huang et al., 2015; Hu et al., 2014). Image processing is another research area which is an up-to-date topic on the implementation of HDMR based algorithms (Altin & Tunga, 2014; Tunga, 2014; Karaca & Tunga, 2016).

To name a few, there is ANOVA-HDMR (Rabitz et al., 1999; Shorter et al., 1999) which is used in the statistics, CUT-HDMR (Li et al., 2001a,b) which uses multivariate function values on lines, planes and hyperplanes passing through a cut center and RS-HDMR (Li et al., 2002a, 2003b,a; Wang et al., 2003) where rs stands for random sampling. These techniques were used in financial applications, risk analysis researches, econometrics ap-

plications and some chemical applications.

M. Demiralp gathered a group of students and lecturers to develop different HDMR based methods since 2000. There are many HDMR types developed by this group for different kinds of scientific problems (Demiralp, 2003; Demiralp & Tunga, 2001). Factorized HDMR (Kurşunlu & Demiralp, 2003; Alper Tunga & Demiralp, 2004), Logarithmic HDMR [24, 25] and Hybrid HDMR (Tunga & Demiralp, 2003a, 2006; Demiralp & Tunga, 2003), which is the combination of Factorized and Logarithmic HDMR, are to name a few. These methods were applied to many different areas of engineering analysis. In these developed methods, weight function is multiplicative, which is not realistic. For this situation, Generalized HDMR (Tunga & Demiralp, 2003b; Kanmaz & Demiralp, 2003) was developed. Cut HDMR (Li et al., 2001b) has been developed in response to the fact that a number of very high input-output sequences are available. Multicut-HDMR (Li et al., 2004) which is the general state of this method was also developed. In addition to these mentioned methods, there are also RS-HDMR (Li et al., 2003a, 2002a), Transformational HDMR (Demiralp, 2006) were developed.

These HDMR methods are used in algebraic eigenvalue problems, modelling, Schrödinger's equation, hyperrotation based applications, optimal control of harmonic oscillator, multivariate diffusion equation, Laplace transform applications, exponential matrix evaluation, evolution operators, parametric sensitivity analysis and so on Tunga & Demıralp (2008); Baykara & Demiralp (2003); Akkemik & Demiralp (2003); Civlekoğlu & Demiralp (2003); Fırat et al. (2003); Kaman & Demiralp (2003); Şenol et al. (2003); Yaman & Demiralp (2003, 2004); Kaman & Demiralp (2004). With HDMR, rather that specifying the analytic structure of the function, the values of the multivariate function can be given as a finite number of points. These nodes can be shown by the cartesian product of the given values, which has N-tuples, for each independent variable. If the interpolation contains the values of the function $f(x_1, \ldots, x_N)$ on this cartesian product's elements, then HDMR can be used. The HDMR philosophy is explained in more detail in Section 4.1.8.

3. DATA PREPARATION

Constructing a relationship between the products and users, and making decisions to find the most appropriate product for them is the main idea behind the recommendation systems for e-commerce. These kinds of recommendation systems includes usually three steps. First step is to acquire preference from customers' purchase data, second step is to compute the recommendation using proper techniques or algorithms and finally the last step is to present the recommendation results to the customer. In order to make qualified recommendations, the data should be modeled well.

In this thesis, the data set is obtained from an e-commerce web page which deals with apparels domain for woman. The item catalog of this company contains evening dresses, sportswear, swimwear, accessories, bags, outerwear and regular clothing like dresses, tops like vest and jacket, bottoms like skirts and pants and knitwear. This company has been actively involved in the market since 2012. Lately, it also offers shopping opportunities for overseas customers. This web page currently is not using any recommendation system and the data they shared with us, has never been used in any research before. There were some handicaps of this data. It was really unorganized, difficult to understand and is not suitable for most of the data mining techniques and plain HDMR method. The owner of this e-commerce page sent the data separately for each month of a year. Therefore, multiple data sources are combined to obtain one big data set. The data had too many attributes, some of them were unnecessary. Thus, only the relevant ones to the analysis were retrieved.

At the end of these data selection and integration steps, the data set consists of purchasing data, purchase quantity, price and user names. The data set is imported to MS SQL Server Management Studio to manipulate. There are 1123 unique customers, 1600 unique products and a total of 183514 purchases in the data set. Originally, the data set has more products and more customers. However, there were noise and some outliers, we performed data cleaning to remove these inconsistent data. Table 3.1 shows the summary

Table 3.1: Summary of the data set

Number of customers	1123
Number of products	1600
Number of purchases	183514
Number of models	65
Number of categories	14
Number of color values	17
Number of price ranges	14

of the data.

Products were represented by a very long textual product name. In order to create attributes for HDMR method, this product names were divided into models, types and colors which were available in the product name. All of the product names start with the brand name (which was not used in the modelling process), then the type of the product (e.g. if it is a pant or a skirt), then the model of the product (e.g if the product is a pant, is it skinny leg or straight cut), then the color of the product (e.g a black or a blue skirt) and finally the size of the product (e.g large or medium) which is not used in the modelling phase.

Each row of the data set consists of customer name, product name, purchase quantity and price columns. The product name column includes information about brand, type, model, size and color of the product. We have re-structured the data set to be able to process for generating top-N recommendations. Table 3.2 shows a simple database with records (i.e., ”rows”) that describe an order before restructured.

To be able to work on this data, as a first step, it was restructured. To restructure the data, the product name values are divided into words by a java program. Then the brand and size values are ignored since they will not be used while modeling the data. Rests of the words are used to determine the product type and model values. Each category item has a unique id number from 1 to number of items in that category as shown Table 3.3. Only the color value are sorted according to the color scale, rest of them are created randomly. The

Table 3.2: Sample records of the data set from the apparels domain

CustomerName	ProductName	Price(TL)	Quantity
xxx	Boat Neck Dress- Blue EO41025-17 40	80	2
xxx	Bicycle Collar Tunic- Pink 8271-008-43 38	40	1
yyy	Crew Neck Tunic- Pink 8300-049-43 38	120	1

Table 3.3: Category details

Categories	Values
Category1 (types)	{pant=1, skirt=2, tunic=3, dress=4, topcoat=5, catsuit=6, jacket=7, blouse=8, shirt=9, vest=10,tracksuit=11, overalls=12, vest=13, ferace=14}
Category2 (models)	{classic cut=1, skinny leg=2, bell bottoms=3,straight cut=4, flared=5, round collar=6, sharp collar=7, crew neck=8, asymmetric=9, shirtwaist=10, neckband=11, hoodie=12, double-breasted=13, v neck=14, dressy=15, casual=16, . . . , pajamas=65}
Category3 (colors)	{white=1, yellow=2, powder color=3, beige=4,salmon=5, orange=6, red=7, claret red=8, pink=9, coral=10, purple=11, blue=12, green=13, mink=14, brown=15, grey=16, black=17}
Category4 (price range)	{0-19.99=1, 20.00-29.99=2, 30.00-39.99=3, 40.00-45.99=4,46.00-50.99=5, 51.00-60.99=6, 61.00-69.99=7, 70.00-79.99=8, 80.00-89.99=9, 90.00-99.99=10, 100.00-119.99=11, 120.00-159.99=12, 160.00-499.99=13, $>$500.00=14}

color values are ordered according to the Munsell colour system (Munsell et al., 1950).

Table 3.4: Restructured product database example

CustomerId	Type	Model	Color	Price Range	Quantity
1	4	21	12	9	2
1	3	18	9	4	1
2	3	8	9	12	1

Next, each customer designated with a unique id number. Then, each product categorized according to the color, model and type using a java program. In the data, we have 65 models, 14 types, 17 color values and 14 price ranges. The java code simply retrieves the product names and assigns unique values to each type, model and color value. For the price values, different price ranges was determined, having nearly same number of items in each range, and assigned a value accordingly.

The restructured data version for Table 3.2 is shown in Table 3.4. The column names (e.g., Type or Color) are properties of products. These properties can also be called "characteristics "or "variables ". Each record contains a value for each attribute.

After a few trials, we decided to normalize this data set by scaling values between 0 and 1, in order to have the same range of values for each of the variables to increase the efficiency. Each variables range is set to $[0 - 1]$ and the values are calculated using the normalization formula below, where $m_1 = 14$, $m_2 = 65$, $m_3 = 17$ and $m_4 = 14$, and x_1, x_2, x_3, x_4 represent the values in each category (type, model, color, price range). Also, $x_i^{(j)}$ and $z_i^{(j)}$ correspond to the unnormalized and the normalized values, respectively.

We perform this normalization process for every dimension in our dataset. This process helps us to obtain a unit Gaussian distribution, having a mean of 0 and a standard deviation of 1, for every dimension. Hence, regardless of the distribution of the data set, our method behaves as if it were derived from this distribution. So, we can expect to obtain similar results with other data with different distributions.

$$z_i^{(j)} = \frac{x_i^{(j)} - min(x_i)}{max(x_i) - min(x_i)}, \quad 1 \leq i \leq 4, \quad 1 \leq j \leq m_i \tag{3.1}$$

Table 3.5: Normalized category details

Categories	Values
Category1 (types)	{pant=0, skirt=0.077, tunic=0.154, dress=0.231, topcoat=0.308, catsuit=0.385, jacket=0.462, blouse=0.538, shirt=0.615, vest=0.692, tracksuit=0.769, overalls=0.846, vest=0.923, ferace=1}
Category2 (models)	{classiccut=0, skinnyleg=0.016, bellbottoms=0.031, straight cut=0.047, flared=0.063, roundcollar=0.078, sharpcollar=0.094, crewneck=0.110, asymmetric=0.125, shirtwaist=0.141, neckband=0.156, hoodie=0.172, double-breasted=0.186, vneck=0.203, dressy=0.219, casual=0.25 . . . pajamas=1}
Category3 (colors)	{white=0, yellow=0.063, powder color=0.125, beige=0.186, salmon=0.25, orange=0.313, red=0.375, claret red=0.438, pink=0.5, coral=0.563, purple=0.625, blue=0.686, green=0.75, mink=0.813, brown=0.875, grey=0.938, black=1}
Category4 (price range)	{0-19.99=0, 20.00-29.99=0.077, 30.00-39.99=0.154, 40.00-45.99=0.231, 46.00-50.99=0.308, 51.00-60.99=0.385, 61.00-69.99=0.462, 70.00-79.99=0.538, 80.00-89.99=0.615, 90.00-99.99=0.692, 100.00-119.99=0.769, 120.00-159.99=0.846, 160.00-499.99=0.923, >500.00=1}

According to the categories in Table 3.5, purchase history matrix, shown in Equation 3.2, for each customer is created.

$$PurchaseHistory = \begin{pmatrix} type_1 & model_1 & color_1 & price_1 \\ type_2 & model_2 & color_2 & price_2 \\ type_3 & model_3 & color_3 & price_3 \\ \vdots & \vdots & \vdots & \vdots \\ type_n & model_n & color_n & price_n \end{pmatrix} \tag{3.2}$$

In this matrix, each column represents a category and each row represents an item. The numbers in the matrix are the category item numbers and they will be used in R-HDMR algorithm as the training node values and the total number of different values that the original space parameters (each category) can take on are 65, 17, 14, and 14 respectively for each category. We know that Customer xxx purchased *Boat Neck Dress-Blue EO41025-17 40* and *Bicycle Collar Tunic- Pink 8271-008-43 38*. This customers' purchase matrix is shown below.

$$Customerxxx = \begin{bmatrix} 0.231 & 0.313 & 0.686 & 0.615 \\ 0.154 & 0.266 & 0.5 & 0.231 \end{bmatrix} \tag{3.3}$$

In our approach, we represent each customer with the customer's purchase history matrix and purchase quantity vector. Then, we exploit these matrices and corresponding vectors for recommending top-N items to a particular customer. In the purchase history matrix, each row contains the information about the purchased item and each row of the corresponding quantity vector stores number of times that the item purchased by the corresponding customer. We represent purchase history matrix and the purchase quantity vectors by C_k and φ_j for the k^{th} customer, respectively. For instance, Customer xxx (where k=1) has purchased the item *Boat Neck Dress-Blue EO41025-17 40* for 2 times and the item *Bicycle Collar Tunic- Pink 8271-008-43 38* for only 1 time. Then, the purchase history matrix and the purchase quantity vector of the k^{th} customer are written as follows:

$$C_k = \begin{bmatrix} 0.231 & 0.313 & 0.686 & 0.615 \\ 0.154 & 0.266 & 0.5 & 0.231 \end{bmatrix}, \quad \varphi_k = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \tag{3.4}$$

4. METHODS

In this chapter, we gave a general overview of recommendation systems and the basic approaches of recommendation generation. In addition, mathematical background of HDMR is also given.

4.1 RECOMMENDATION SYSTEMS

The issue of information search and selection has become increasingly ineligible because of the growth of online environments; users are bored by suggestions which they may not have the time or knowledge to assess (Gavalas et al., 2014). Researchers tend to develop more effective recommenders in the cause of the technology used for recommendation systems (RS) being grown over the past years into a rich collection of tools. RSs are software tools used to provide recommendations to be helpful to a specific user (Ricci et al., 2011). Tapestry was the first recommendation system designed to recommend documents from newsgroups. The authors also introduced the term collaborative filtering as they used social collaboration to help users with large volume of documents (Resnick et al., 1994). A RS must be reliable providing good recommendations and showing information about the recommendations. Another important point of RSs is the way they should display the information about the recommended items:

a) The recommended item must be easy to recognize by the user

b) The item must be easy to assess

c) The ratings must be easy to understand and meaningful

d) Explanations must provide a quick and easy way for the user to evaluate the recommendation.

The personalization of recommendations can be different for each site. Galland (Galland & Cautis, 2010) classified the recommendations into four groups; generic group where everyone gets recommendations, demographic group where everyone in the same category gets the same recommendation, contextual group where only the current activity affects the recommendation and persistent group where recommendation depends on long term interest. Konstan and Riedl (Konstan & Riedl, 2012) state that there are a lot of different ways to get recommendations. The most frequently used ways are depend on the previous knowledge of alike users or alike contents. Mostly used versions of these algorithms are called collaborative filtering and cluster models. These algorithms use items of these similar customers and when the purchased items by this user are eliminated, a recommendation is made to the user from the remaining item list. The main approach in these algorithms is ”people who agreed in the past, will agree on future too ”. The recommendation problem consists of suggesting items that should be the most appealing ones to a user according to her preferences. In the literature several types of RS have been proposed, varying, e.g. in the types of data used, and in the methods with which recommendations are generated (Campos et al., 2014). While designing a recommendation system, one approach, that has seen wide use, is collaborative filtering (Breese et al., 1998). The main idea of collaborative filtering is ”similar users share similar interests ”(Moradi & Ahmadian, 2015). If the number of distinct products is represented by N, each customer is symbolized by an N-dimensional vector of items in collaborative filtering algorithm (Linden et al., 2003). If the items are purchased or positively rated, the components of the vector are positive and if the items are rated negatively, the components of the vector are negative. The algorithm finds best similar customers to the user and generated recommendations accordingly. Many different ways can be used to find the similarity between two customers as explained in 4.1.7. Like the similarity methods, there different kinds of techniques that can be used by the algorithm to select recommendations from the similar customers' items. Most common one is to calculate how many customers purchased it and then use that value to rank each item. In cluster models, generating recommendations are treated like a classification problem. The algorithm splits customers into numerous segments. Using these segments the algorithm tries to find the most simi-

lar customers to the user. When the segments containing the most similar customers are found, the user is assigned to that segment. These segments are formed mostly using a clustering or any other unsupervised learning techniques. Once the algorithm generates the segments, vectors that summarize each segment are formed and the similarity between the user and these vectors are computed. Then the segment with the strongest similarity is chosen to classify the user. The third approach, called content-based filtering, rather than finding similar customers it focuses on finding similar items. Content-based filtering methods are basically based on two things. One of them is a description of the item and the other one is the profile of the user's preference (Brusilovski et al., 2007). For each of the user's purchased and rated items, the algorithm attempts to find similar items. In content-based filtering, the main aim is to find other common items by the same author, category and publisher or with similar keywords. For instance if a customer purchases a book, the system might recommend other books with the same category, other books with the same author, or other books published by the same publisher. This approach was first used in information recovery field by comparing text document contents and user profiles (Moreno et al., 2016). At amazon.com, to personalize the web site for each customer, they use recommendation algorithms (Linden et al., 2003). Their algorithm is called item-to-item collaborative filtering. The algorithm matches each of the user's purchased and rated items to similar items rather than matching the user to similar customers. Then a recommendation list is formed using those similar items. After finding items that the customers tend to buy, the algorithm builds a similar-item table. Most similar match for a given item is determined by the algorithm using this similar-item table. The following iterative algorithm (Linden et al., 2003), provides an approach by calculating the similarity between a single product and all related products:

For each item in product catalog, I_1

For each customer C who purchased I_2

For each item I_2 purchased by customer C

Record that a customer purchased I_1 and I_2

For each item I_2

Compute the similarity between I_1 and I_2

It is possible to compute the similarity between two items in various ways, but as mentioned earlier, a common method is to use the cosine measure. In this algorithm, each vector corresponds to an item, and the vector's M dimensions correspond to customers who have purchased that item. By developing a customized shopping experience for each customer, recommendation algorithms provide an effective form of targeted marketing. The main aim of this research is to use IHDMR philosophy to make categorical recommendations based on a specific time period.

A typical scenario for a recommendation systems is basically a Web application where the target user can interact. Generally, the Web application, the system, introduces a list of items to the user and the user chooses among these items which he or she wants to get more detail or simply wants to purchase. This Web application can simply be an e-commerce site, online news sites, movie rental sites and etc.

Another way to look at recommendation problem, is to look at it as an instant of a data mining problem, where you have the data preparation step like feature selection, dimensionality reduction or normalization, then you have a data mining step where you apply all the machine learning methods like clustering, classification and rule mining and so on. Further more, you have the postprocessing step like filtering, visualization and etc.

Most of the recommendation systems are based on the collaborative filtering and the content-based models (Breese et al., 1998) in the literature. In the following sections, we introduce the basic approaches of recommendation generation shown in Table 4.1.

4.1.1 Collaborative Filtering

Collaborative Filtering (CF) is the most common technique in literature for recommendation generation. CF is used when we recommend things based on past user behaviour.

Table 4.1: Recommender approaches

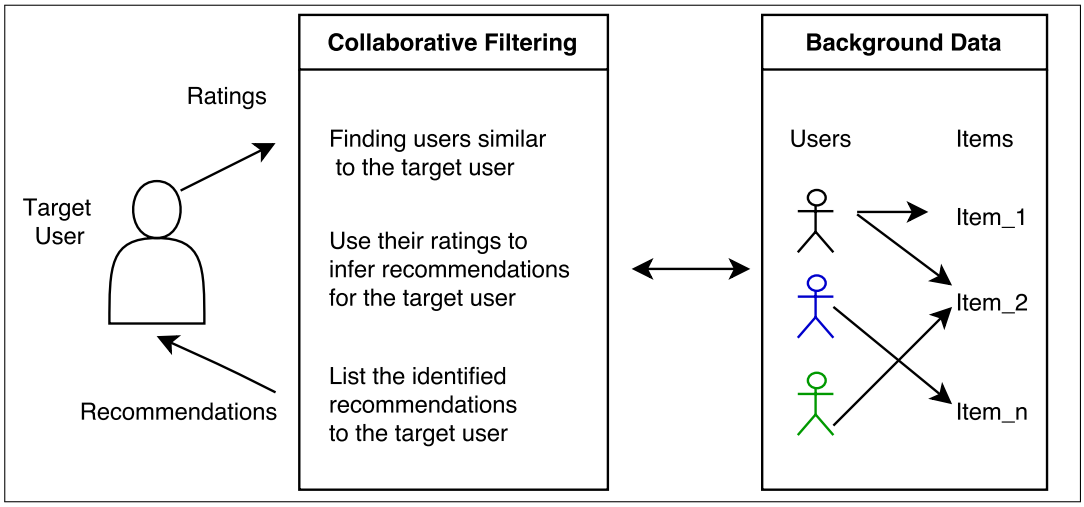
Name	Description
Collaborative Filtering	Recommend items according to users with similar tastes
Content Based	Recommend items that the user preferred in the past
Personalized Learning to Rank	Ranking problem
Social Recommendations	Trust based
Hybrid	Combination of above

This approach is you do not need any domain expertise, which means you do not need to know if you are recommending books, movies or music. The idea behind this approach is leveraging the relation between users as shown on Figure 4.1. There are two kinds of CF approaches, user-based and item-based approach. Neither of them needs to have any information about the items. The main idea of collaborative filtering is "similar users share similar interests"(Moradi & Ahmadian, 2015). The collaborative filtering-based algorithms exploit the most similar customers to a user and generate recommendations accordingly. There are many different ways of measuring the similarity between two customers where the cosine similarity is the most commonly used metric. Once the similarity information between the customers is obtained, the problem becomes recommending items purchased by the similar customers. The most intuitive approach is to recommend items that have been mostly purchased by the most similar customers.

In the item-based approach, recommendations based on the similarity between items but that similarity is based only past user behaviour. Basically, the main idea is leveraging what users did in the past to infer a similarity function between items.

In collaborative filtering, each customer is represented by an N-dimensional vector which carries the rating information for each item among N of them (Linden et al., 2003). Each user has a list of items with associated opinion, which is whether a user liked an item or not. Usually CF is applicable with explicit data, which contains the rating score for items. In addition the data, an active user whom the recommendations are generated for,

Figure 4.1: Collaborative filtering approach philosophy



Source: (Felfernig et al., 2014)

which is the target user, sa metric for measuring similarity between users and a method for selecting a subset of users are needed.

The basic steps for CF are;

a) Set of ratings for the target user

b) Set of users most similar to the target user

c) Items there similar users liked/purchased

d) Generate a rating that would be given by the target user to the items

e) Based on these predicted ratings, recommend a set of top-N items

Like every approach, CF has some advantages and disadvantages. As an advantage, CF requires minimal domain knowledge. You can apply the same method independently what you are recommending. You do not have to have ant internal or structural definition for the items and in most cases it generates good enough recommendation results. the disadvantages are, you need a large number reliable of data. In addition, you need items to be standardized in this data.

CF can be personalized or non-personalized. In personalized CF, predictions are based on the ratings expressed by similar users and similar users are different for each target user. However in non-personalized CF, recommendations are generating by averaging the recommendations of all the users, which means recommending the most popular, most selling items.

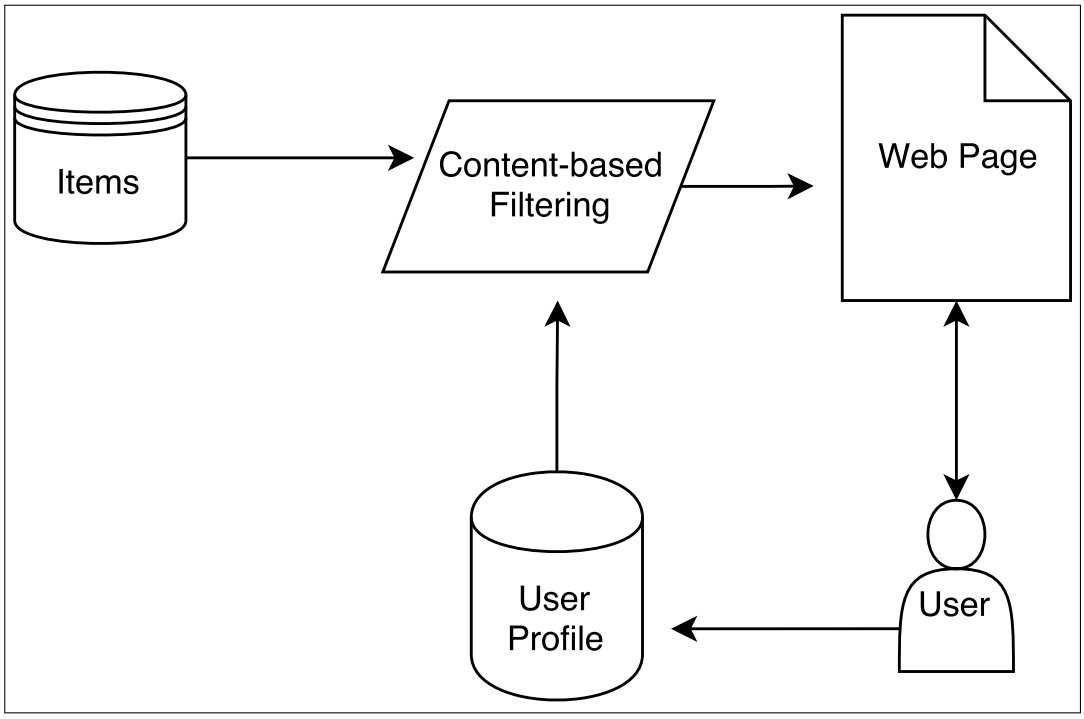
Since recommendation generation is a part of data mining, clustering, artificial neural networks and association rule mining are the most used algorithms for this domain. Some collaborative filtering-based recommendation systems can be found in (Linden et al., 2003; Sarwar et al., 2001; Resnick et al., 1994). In this thesis, Collaborative Filtering philosophy is used, which is explained in more detail in the Chapter 5.

4.1.2 Content-Based Filtering

Another type of recommendation system known as content-based filtering focuses on finding the similar items rather than finding the similar customers. In Content-Based Filtering approach, what the user did on the past is not important, it is completely based on the domain knowledge, knowing what the items are, what they mean. Algorithms basically try to find similarity functions that describe items that are similar based on the descriptions of the items. According to the item descriptions the algorithms identify products that might be interesting for a user (Pazzani & Billsus, 2007). In other words, we get recommendations based on our past purchases or browses.

This approach is first used in information recovery field by comparing text document contents and user profiles (Moreno et al., 2016). Content-based filtering methods are mostly consider two criteria: the description of the item and the profile of the user's preference (Brusilovski et al., 2007). For each of item that have been purchased or rated by the user, the algorithm attempts to find the similar items. For instance, if a customer purchases a book, the system might recommend other books in the same category, written by the same author, containing the similar keywords or published by the same publisher.

Figure 4.2: Content-based filtering approach philosophy



Content-based filtering algorithms needs data that the users provide. This data can be collected either explicitly (for instance rating data) or implicitly which means clicking a link and etc. One of the key points in this approach is the creation of user profiles, which is used to generate recommendations. As shown on Figure4.2, user profiles are created through the data that the user provides by interacting with the web page. The more data the user provides, more accurate recommendations he or she gets. Item content is also an important key for content-base algorithms. The content of an item are attributes or characteristics of it. For instance genre of a film, author of a book and so on. Based on previously purchased item content, we can get similar recommendations.

One of the advantages of this approach is that users get highly suited recommendations since content-based recommendations rely on only the content of items themselves. Unlike the black-box process of CF, users can easily understand why they are getting that recommendation. This approach also avoids the cold-start problem CF has since not much data is needed to start recommending. In addition to these advantages, new items in the

catalog can be recommended to the users immediately because in content-based filtering it is not required other users interaction with an item before it gets recommended.

On the other hand, there are several challenges. First of all, the biggest problem is diversity. It is very important for a recommendation system to produce novel results which means users what to see items that they was not expecting. As mentioned before, in content-base approaches domain knowledge is enough. However content-based recommenders are common for text based data. Therefore, the data should be well organized in order to create user profiles which raises scalability as a second challenge. Related works about content-based filtering can be found in (Van Meteren & Van Someren, 2000; Basu et al., 1998; Zeng et al., 2003).

Due to the content of our data, content-based approach was not applicable for this thesis.

4.1.3 Personalized Learning to Rank

The final goal of most recommendation systems is to produce a ranking. A set of possible items are available to present to the target user and an order list or a ranking of these items has to established. Popularity, which means recommending the most popular items is always a good idea. Users commonly pay attention to the few items at the top of the recommendation list. Hence, the challenge is to rank the most relevant items as high as possible in this list. In this approach, the main aim is not trying to produce a rating score. Instead, the order is important. Learning to rank problems can also be considered as a standard supervised classification problem by contracting a ranking model from the data. It can be said that learning to rank models are divided into two categories, point-wise and pair-wise ranking methods.

Point-wise ranking models are CF algorithms that uses preferences scores of each items to learn a ranking model (Koren & Sill, 2011). On the other hand, if the CF algorithms are developed considering the preferences of each user to a pair of items, they can be classified as pair-wise learning to rank methods (Karatzoglou et al., 2013).

4.1.4 Social Recommendations

Social recommendations are different from the above mentioned approaches. They are also called Trust-Based Recommendation Systems. The basic idea for these approaches is to use explicit connection between users to define a notion of trust. The relation between users are not based on the correlation of item they purchased or liked or watched. The basic concept of this approach is trust. If a user has a high level of trust in another user, whatever the second user likes, the target user will also like. The trust concept is not in the traditional sense, its trust in the sense of how much you trust the recommendation from the other users. The trust in recommendation systems is usually used to explain similarity in opinions. The trust is used as a way to give weight to a user. Social connections of users can also be used.

Social recommendation systems uses trust as a score or combines trust and similarity scores while giving recommendations (Golbeck, 2009). A very known example can be given from Epinions web site. In this web site items are recommended by trusted users (Selmi et al., 2016).

4.1.5 Cluster Models

In cluster models, generating recommendations are considered as a classification problem. First, the algorithm splits customers into numerous clusters by using a clustering or any other unsupervised learning techniques. Once the algorithm generates the clusters, representative vectors that summarize each cluster are formed. Then, the similarity between the target user and these vectors are computed. Finally, the target user is assigned into the most similar cluster. The recommendation is performed using the historical information of the customers in that cluster.

Table 4.2: Hybridization methods

Hybridization Method	Description
Weighted	Outputs of several different methods are combined. Each output has different weight of importance to affect the final result
Switching	System changes the used recommendation generation technique to another under a switching condition
Mixed	Recommendation results of more than one methods are shown to the user at the same time
Cascade	One method uses another methods output as an input
Feature Combination	Features from several recommendation sources are combined to create input for a specific method
Meta-level	The establish model from a recommendation system is used as an input for another method

4.1.6 Hybrid Approaches

Hybrid approaches usually uses the combination of content-based and collaborative filtering approaches to produce recommendations (Porcel & Herrera-Viedma, 2010). Probabilistic methods are usually used by hybrid approaches (Bobadilla et al., 2013) for instance genetic algorithms (Ho et al., 2007), neural networks (Ren et al., 2008), Bayesian networks (De Campos et al., 2010), clustering (Shinde & Kulkarni, 2012) and latent features (Maneeroj & Takasu, 2009). A hybrid system tries to use the advantages of an algorithm to fix the disadvantages of the other algorithm.

The summary of different methods for hybrid recommendations are given in Table4.2.

The proposed approach in this thesis is also a hybrid system which combines collaborative filtering with high dimensional model representation philosophy.

4.1.7 Similarity Measures in Recommendation System

In a recommendation system, similarity is about finding items or users that are similar to each other. Depends on what kind of algorithm is being used, the technique to measure

the similarity is also differs. The following similarity measures are the popular metrics used mostly for generating recommendations.

Euclidean Distance

The Euclidean distance is probably the easiest similarity measure to implement. When the problem is finding the similarity or dissimilarity Euclidean distance forms the basis. The distance between two-dimensional vectors $u = (x_1, y_1)$ and $v = (x_2, y_2)$ is given by following expression where x_i and y_i are rating scores of a specific item given by different users.

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} = \sqrt{\sum_{1=1}^{2}(x_i - y_i)^2} \tag{4.1}$$

If we put it in other words, Euclidean distance is the square root of the sum of squared differences between corresponding elements of the two vectors which is scaled from 0 to 1 (Bandyopadhyay & Saha, 2012). Even if this is one of the mostly known similarity metric, in this thesis it has not used.

Pearson Correlation Coefficient

Pearson correlation coefficient simply measures the statistical relationship between two variables showing how highly correlated they are (Ricci et al., 2015). Unlike Euclidean distance, a Pearson correlation measures from -1 to $+1$. If a Pearson correlation coefficient is 1 that means the variables are correlated, if it is -1 it means the opposite, meaning the variables are not correlated. The Pearson correlation coefficient expression is shown below.

$$PC(u, v) = \frac{\sum_n^{i=1}(x_i - \bar{x})(y_i - \bar{y}))}{\sqrt{\sum_n^{i=1}(x_i - \bar{x})^2}\sqrt{\sum_n^{i=1}(y_i - \bar{y})^2}} \tag{4.2}$$

In recommendation systems using the correlation between the target user and the other user/users can be determined with the Pearson correlation coefficient to give a weight to a user's ratings. Several collaborative filtering systems, for instance GroupLens (Resnick et al., 1994) and Ringo (Shardanand, 1994).

Cosine Similarity

In cosine similarity, different from the previously explained methods, the cosine of the angle between two vectors are calculated. Cosine similarity is frequently used in the recommendation domain because it is easy to implement, easy to understand, very efficient to evaluate (Ricci et al., 2015). It also gives the values in between 0 to 1 like the Euclidean distance.

Suppose we have a $n * m$ ratings matrix, it could be the user-item matrix, similarity between the arbitrarily items i and j is denoted with the following formula.

$$sim(i, j) = cos(i, j) = \frac{i \cdot j}{\|i\| * \|j\|} \tag{4.3}$$

In this thesis, while finding similar customers the cosine similarity metric was used.

Jaccard Coefficient

The Jaccard coefficient, which is also referred as the Tanimoto coefficient, evaluates the similarity by dividing the intersection to the union of products (Ricci et al., 2015). For instance, lets assume user A purchased items $7, 3, 2, 4, 1$ and user B purchased items $4, 1, 9, 7, 5$. The products in common (the intersection) are $1, 4, 7$. The union of products are $1, 2, 3, 4, 5, 7, 9$. According to the Jaccard coefficient formula shown below, the similarity measure is number of common items divided by the number of union of items, which is $3/7 = 0.429$. Like Euclidean and cosine measures, this ones similarity range is

also between 1 and 0.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cup B|} \tag{4.4}$$

4.1.8 High Dimensional Model Representation

HDMR is a divide and conquer algorithm. The main aim of HDMR is to partition multivariate data into a number of sets of low-variate data. Only the constant term, the univariate and bivariate terms of HDMR function will be used due to interpolate of each element of the data set by the standing methods. Due to the orthogonality condition, all of these components are forced to be orthogonal. When the constant and univariate terms are used, from N one-dimensional interpolations, one N-dimensional interpolation can be estimated. To decompose a multivariate function into a number of less-variate functions HDMR uses the following expansion.

$$f(x_1, \ldots, x_N) = f_0 + \sum_{i_1=1}^{N} f_{i_1}(x_{i_1}) + \sum_{\substack{i_1, i_2 = 1 \\ i_1 < i_2}}^{N} f_{i_1 i_2}(x_{i_1}, x_{i_2}) + \cdots$$
$$+ f_{1 \ldots N}(x_1, \ldots, x_N) \tag{4.5}$$

When the above expansion is examined, the terms on the right hand side are the constant term, univariate terms, bivariate terms and so on respectively. The following vanishing conditions are used to individually determine these terms,

$$\int_{a_1}^{b_1} dx_1 \ldots \int_{a_N}^{b_N} dx_N W(x_1, \ldots, x_N) f_i(x_i) = 0, \quad 1 \leq i \leq N \tag{4.6}$$

where $W(x_1, \ldots, x_N)$ is a product type weight having the following structure and normalization conditions.

$$W(x_1,\ldots,x_N) \equiv \prod_{j=1}^{N} W_j(x_j),$$
$$\int_{a_j}^{b_j} dx_j W_j(x_j) = 1, \quad x_j \in [a_j, b_j], \quad 1 \le i \le N \tag{4.7}$$

The following orthogonality conditions are defined using the inner product definition to extend the vanishing condition given in (4.6).

$$\left(f_{1i_1\ldots i_k}, f_{1i_2\ldots i_l}\right) = 0, \{i_1, i_2, \ldots, i_k\} \not\equiv \{i_1, i_2, \ldots, i_l\}, \quad 1 \le k, \quad l \le N \tag{4.8}$$

HDMR components must satisfy these orthogonality conditions. The general formula for an inner product of two arbitrary functions $u(x_1,\ldots,x_N)$ and $v(x_1,\ldots,x_N)$ can be written as follows.

$$(u,v) \equiv \int_{a_1}^{b_1} dx_1 \ldots \int_{a_N}^{b_N} dx_N W(x_1,\ldots,x_N) u(x_1,\ldots,x_N) v(x_1,\ldots,x_N) \tag{4.9}$$

We can obtain the constant term of HDMR expansion considering the properties of the weight function and orthogonality conditions. This operation makes the vanishing conditions applicable to find the necessary terms by multiplying both sides of the HDMR expansion with the weight function, $W_1(x_1)W_2(x_2)\ldots W_N(x_N)$, and are integrated over the whole Euclidean space defined by independent variables.

$$\mathcal{I}_0 F(x_1,\ldots,x_N) \equiv \int_{a_1}^{b_1} dx_1 \ldots \int_{a_N}^{b_N} dx_N W(x_1,\ldots,x_N) F(x_1,\ldots,x_N) \tag{4.10}$$

Using this $\mathcal{I}_0$ operator as the following, the constant term of the HDMR expansion can be obtained.

$$f_0 \equiv \mathcal{I}_0 f(x_1,\ldots,x_N) \tag{4.11}$$

In a similar manner, determination of univariate terms can be achieved. We can obtain the

univariate terms $f_i(x_i)$ with the constant term f_0 by eliminating independent variable x_i.

$$\mathcal{I}_i F(x_1,\ldots,x_N) \equiv \int_{a_1}^{b_1} dx_1 W_1(x_1)\ldots\int_{a_{i-1}}^{b_{i-1}} dx_{i-1} W_{i-1}(x_{i-1})$$
$$\int_{a_{i+1}}^{b_{i+1}} dx_{i+1} W_{i+1}(x_{i+1})\times\ldots\times\int_{a_N}^{b_N} dx_N W_N(x_N) F(x_1,\ldots,x_N),$$
$$1\le i\le N \qquad (4.12)$$

When the $\mathcal{I}_i$ operator is applied to both sides of HDMR expansion, we achieve HDMR component $f_i(x_i)$ through the following relation.

$$\mathcal{I}_i f(x_i,\ldots,x_N) = f_0 + f_i(x_i), \quad 1\le i\le N \qquad (4.13)$$

Equation (4.13) can be rewritten in the form of;

$$f_i(x_i) = \mathcal{I}_i f(x_i,\ldots,x_N) - f_0, \quad 1\le i\le N \qquad (4.14)$$

To determine bivariate terms of the HDMR expansion, two independent variables will be eliminated. Both sides of the HDMR expansion given in (4.5) are multiplied by $W_1(x_1)W_2(x_2)\ldots W_{i_1-1}(x_{i_1-1})W_{i_1+1}(x_{i_1+1})\ldots W_{i_2-1}(x_{i_2-1})\,W_{i_2+1}(x_{i_2+1})\ldots W_N(x_N)$ and are integrated over whole Euclidean space defined by independent variables except x_{i_1} and x_{i_2}.

$$\mathcal{I}_{i_1 i_2} F(x_1,\ldots,x_N) \equiv \int_{a_1}^{b_1} dx_1 W_1(x_1)\ldots\int_{a_{i_1-1}}^{b_{i_1-1}} dx_{i_1-1} W_{i_1-1}(x_{i_1-1})$$
$$\int_{a_{i_1+1}}^{b_{i_1+1}} dx_{i_1+1} W_{i_1+1}(x_{i_1+1})\times\ldots\times\int_{a_{i_2-1}}^{b_{i_2-1}} dx_{i_2-1} W_{i_2-1}(x_{i_2-1})$$
$$\int_{a_{i_2+1}}^{b_{i_2+1}} dx_{i_2+1} W_{i_2+1}(x_{i_2+1})\times\ldots\times\int_{a_N}^{b_N} dx_N W_N(x_N) F(x_1,\ldots,x_N) \qquad (4.15)$$

We can again use the orthogonality condition to obtain $f_{i_1 i_2}(x_{i_1}, x_{i_2})$.

$$\mathcal{I}_{i_1 i_2} f(x_1, \ldots, x_N) \equiv f_0 + f_{i_1}(x_{i_1}) + f_{i_2}(x_{i_2}) + f_{i_1 i_2}(x_{i_1}, x_{i_2}), \quad 1 \leq i_1 < i_2 \leq N \tag{4.16}$$

This equation can be rewritten as

$$f_{i_1 i_2}(x_{i_1}, x_{i_2}) = \mathcal{I}_{i_1 i_2} f(x_1, \ldots, x_N) - f_{i_1}(x_{i_1}) - f_{i_2}(x_{i_2}) - f_0, \quad 1 \leq i_1 < i_2 \leq N \tag{4.17}$$

4.1.9 Data Partitioning through Hdmr

The structure of the function is specified as the values on finite points of the Euclidean space defined by the independent variables $x_1, x_2, \ldots, x_N$ rather than analytically. These points are defined through a cartesian product.

$$\mathcal{D} \equiv \mathcal{D}_1 \times \mathcal{D}_2 \times \ldots \times \mathcal{D}_N \tag{4.18}$$

$\mathcal{D}$ consists of N-tuples and can be given as follows

$$\mathcal{D} \equiv \{\tau \mid \tau = (x_1, x_2, \ldots, x_N), x_j \in \mathcal{D}_j, 1 \leq j \leq N\} \tag{4.19}$$

The data of the variable x_j is defined as follows

$$\mathcal{D}_j \equiv \left\{\xi_j^{(k_j)}\right\}_{k_j=1}^{k_j=n_j} = \left\{\xi_j^{(1)}, \ldots, \xi_j^{(n_j)}\right\}, \quad 1 \leq j \leq N \tag{4.20}$$

where each ξ is a value that the variable x_j can take on. Here, n_j is the total number of different ξ values for x_j. Because the structure which needs to be created through interpolation must include the values of the function $f(x_1, \ldots, x_N)$ at the points of this cartesian product set, we represent the weight components in terms of Dirac delta functions as

follows (Tunga & Demiralp, 2008)

$$W_j(x_j) \equiv \sum_{k_j=1}^{n_j} a_{k_j}^{(j)} \delta\Big(x_j - \xi_j^{(k_j)}\Big), \quad x_i \in [a_j, b_j], \quad 1 \leq j \leq N. \tag{4.21}$$

Replacing the above weight function in relation (4.10) and using relation (4.11), the following equation for the constant component for multivariate data partitioning process through HDMR can be formed

$$f_0 = \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} \cdots \sum_{k_N=1}^{n_N} \Big(\prod_{i=1}^{N} a_{k_i}^{(i)} \Big) f(\xi_1^{(k_1)}, \ldots, \xi_N^{(k_N)}) \tag{4.22}$$

where

$$\sum_{k_j=1}^{n_j} a_{k_j}^{(j)} = 1, \quad 1 \leq j \leq N \tag{4.23}$$

which comes from the normalization conditions defined on the weight components given in relation (4.7).

Replacing the Dirac delta type weight function in relation (4.12) and rewriting relation (4.14), we obtain the following structure for the univariate terms

$$\begin{aligned} f_m(\xi_m^{(k_m)}) = & \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} \cdots \sum_{k_{m-1}=1}^{n_{m-1}} \sum_{k_{m+1}=1}^{n_{m+1}} \cdots \sum_{k_N=1}^{n_N} \Big(\prod_{i=1}^{N} a_{k_i}^{(i)} \Big) \\ & \times f\Big(\xi_1^{(k_1)}, \ldots, \xi_m^{(k_m)}, \ldots, \xi_N^{(k_N)}\Big) - f_0, \\ & \xi_m^{(k_m)} \in \mathcal{D}_m, \quad 1 \leq k_m \leq n_m, \quad 1 \leq m \leq N. \end{aligned} \tag{4.24}$$

The above relation results in N tables of ordered pairs such that the $m-th$ table contains n_m number of ordered pairs for the univariate component, $f_m(x_m)$.

A similar replacement of the weight function in relation (4.15) is processed and the fol-

lowing structure is obtained through relation (4.17)

$$
\begin{aligned}
& f_{m_1 m_2}\left(\xi_{m_1}^{(k_{m_1})}, \xi_{m_2}^{(k_{m_2})}\right) = \\
& \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} \cdots \sum_{k_{m_1-1}=1}^{n_{m_1-1}} \sum_{k_{m_1+1}=1}^{n_{m_1+1}} \cdots \sum_{k_{m_2-1}=1}^{n_{m_2-1}} \sum_{k_{m_2+1}=1}^{n_{m_2+1}} \cdots \sum_{k_N=1}^{n_N} \left(\prod_{\substack{i=1 \\ i \neq m_1 \wedge i \neq m_2}}^{N} a_{k_i}^{(i)} \right) \\
& \times f\left(\xi_1^{(k_1)}, \ldots, \xi_{m_1}^{(k_{m_1})}, \ldots, \xi_{m_2}^{(k_{m_2})}, \ldots, \xi_N^{(k_N)}\right) - f_{m_1}\left(\xi_{m_1}^{(k_{m_1})}\right) \\
& - f_{m_2}\left(\xi_{m_2}^{(k_{m_2})}\right) - f_0, \quad \xi_{m_1}^{(k_{m_1})} \in \mathcal{D}_{m_1}, \quad \xi_{m_2}^{(k_{m_2})} \in \mathcal{D}_{m_2}, \\
& 1 \leq k_{m_1} \leq n_{m_1}, \quad 1 \leq k_{m_2} \leq n_{m_2}, \quad 1 \leq m_1 < m_2 \leq N
\end{aligned}
\tag{4.25}
$$

Now, we have $N(N-1)/2$ tables of ordered pairs. Each table has $n_{m_1} n_{m_2}$ ($1 \leq m_1 < m_2 \leq N$) number of pairs of data for the corresponding bivariate component.

Using the constant term, univariate terms and bivariate terms, the approximate analytical structure of the multivariate function can be obtained. Instead of obtaining an analytical structure for the function $f_m(x_m)$, using the terms mentioned above, a table of n_m number of pairs of data can be obtained. This table helps us to determine the function $f_m(x_m)$ under an assumed structure by providing an opportunity to interpolate the corresponding data. By the help of this, a set of univariate interpolations can be approximately reduced from multivariate interpolation. An analytical structure must be defined to determine overall structure of the function. If the function to be determined by HDMR is sufficiently smooth, then the function can be represented with a multinomial of all independent variables over the continuous region produced by the Cartesian product of the related intervals. For this reason, for $f_m(x_m)$, a polynomial representation should be built firstly. Interpolation is useful tool for estimating function values when we don't have precise data. Lagrange polynomials are used for polynomial interpolation. There will be a polynomial of degree $N-1$, if there are N data values. The Lagrange interpolation formula is

$$
P_m(x_m) = \sum_{k_m=1}^{n_m} L_{k_m}(x_m) f_m\left(\xi_m^{(k_m)}\right), \quad \xi_m^{(k_m)} \in \mathcal{D}_m, \quad 1 \leq m \leq N \tag{4.26}
$$

Here $L_{k_m}(x_m)$, $f_m\left(\xi_m^{(k_m)}\right)$ and $P_m(x_m)$ are Lagrange coefficient polynomials which are independent of the structure of the function, the known values of the function and the desired value of the function respectively. The structures of these polynomials are given below

$$L_{k_m}(x_m) \equiv \prod_{\substack{i=1 \\ i \neq k_m}}^{n_m} \frac{\left(x_m - \xi_m^{(i)}\right)}{\left(\xi_m^{(k_m)} - \xi_m^{(i)}\right)},$$
$$\xi_m^{(k_m)} \in \mathcal{D}_m, \quad 1 \leq k_m \leq n_m, \quad 1 \leq m \leq N \tag{4.27}$$

Univariate functions given in relation (4.28) are obtained as the Lagrange polynomials are constructed. These functions can be considered as univariate components of HDMR for the multivariate function, $f(x_1, \ldots, x_N)$. The following multinomial approximation is provided by the expansion formed by the summation of these functions and the constant term.

$$f(x_1, \ldots, x_N) \approx f_0 + \sum_{m=1}^{N} P_m(x_m) \tag{4.28}$$

This should be considered as a univariate additive decomposition approximation. When a table of data for the bivariate functions $f_{m_1 m_2}(x_{m_1}, x_{m_2})$ is constructed to determine the overall structure of the function, the following interpolative multinomials should be built.

$$P_{m_1 m_2}(x_{m_1}, x_{m_2}) = \sum_{k_{m_1}=1}^{n_{m_1}} \sum_{k_{m_2}=1}^{n_{m_2}} L_{k m_1}(x_{m_1}) L_{k m_2}(x_{m_2}) f_{m_1 m_2}\left(\xi_{m_1}^{(k_{m_1})}, \xi_{m_2}^{(k_{m_2})}\right),$$
$$\xi_{m_1}^{(k_{m_1})} \in \mathcal{D}_{m_1}, \xi_{m_2}^{(k_{m_2})} \in \mathcal{D}_{m_2}, \quad 1 \leq m_1, \quad m_2 \leq N \tag{4.29}$$

In terms of these multinomials and the polynomials in Equation (4.1.9) the overall approximation to $f(x_1, \ldots, x_N)$ can be written as follows.

$$f(x_1, \ldots, x_N) \approx f_0 + \sum_{m=1}^{N} P_m(x_m) + \sum_{\substack{m_1, m_2 = 1 \\ m_1 < m_2}}^{N} P_{m_1\, m_2}(x_{m_1}, x_{m_2}) \tag{4.30}$$

4.1.10 Indexing Hdmr

The HDMR method can only partition multivariate data having an orthogonal geometry. However, the data sets from real cases mostly have a non-orthogonal structure which results in implementing a different HDMR based methodology to construct an analytical model for the given multivariate data set. This thesis aims to use Indexing HDMR (Tunga, 2011) for the analytical model construction process and build the HDMR philosophy as a recommendation system. The Indexing HDMR algorithm assembles an orthogonal geometry by forcing an indexing scheme so that the orthogonal geometry will be obtained from the given multivariate data. Consequently, the HDMR method can be used to partition that new multivariate data set.

There are four main steps in Indexing HDMR algorithm to make HDMR method applicable for real cases having non-orthogonal geometry. The first step is generating an index space with orthogonal geometry (cartesian product set). To create this index space, prime factors of the number of nodes of the considered data set are calculated. The prime factors then must provide the following relation

$$m = n_1 \times n_2 \times \cdots \times n_N \tag{4.31}$$

while the number of these prime factors should best fit the number of parameters of the given problem. Each prime factor corresponds to the number of elements of each index set defined for each independent variable. The definition of these index sets are given as follows

$$\xi_1 \in \{1, 2, \ldots, n_1\}, \quad \xi_2 \in \{1, 2, \ldots, n_2\}, \quad \ldots, \quad \xi_N \in \{1, 2, \ldots, n_N\} \tag{4.32}$$

A cartesian product set is constructed using these index sets to set the orthogonal geometry that HDMR needs.

The second step is constructing a one-to-one mapping between the original space and

index space. To process this mapping the given training set is sorted by the function values in either ascending or descending order and then the following mapping schema is implemented

$$\begin{aligned}
\left(\upsilon_1^{(1)},\ldots,\upsilon_{N-1}^{(1)},\upsilon_N^{(1)},\varphi_1\right) &\Rightarrow \left(\xi_1^{(1)},\ldots,\xi_{N-1}^{(1)},\xi_N^{(1)},\varphi_1\right)\\
\left(\upsilon_1^{(2)},\ldots,\upsilon_{N-1}^{(2)},\upsilon_N^{(2)},\varphi_2\right) &\Rightarrow \left(\xi_1^{(1)},\ldots,\xi_{N-1}^{(1)},\xi_N^{(2)},\varphi_2\right)\\
&\vdots\\
\left(\upsilon_1^{(m)},\ldots,\upsilon_{N-1}^{(m)},\upsilon_N^{(m)},\varphi_m\right) &\Rightarrow \left(\xi_1^{(n_1)},\ldots,\xi_{N-1}^{(n_{N-1})},\xi_N^{(n_N)},\varphi_m\right)
\end{aligned} \tag{4.33}$$

where υ, φ and ξ stand for the training node, the class information of each training node and the index node respectively.

The third step is determining the analytical model for the index space using the HDMR algorithm. The bivariate HDMR approximant given in (4.30) is obtained at the end of this step. The achieved model is true for the index space. To predict the class (function) value of a testing node coming from the original space, we need to set an appropriate index node and insert the parameter values of that index node into the HDMR approximant. To this end, the last step is to define a rule to evaluate the parameters of that index node using the index information of the most similar training node to that testing node. To find the similarity between nodes, various similarity metrics can be used. In this study, the Euclidean distance metric given below is used to specify the most similar training node to the considered testing node

$$d_\ell^{(j)} = \sqrt{\sum_{i_1=1}^{N}(\upsilon_{i_1}^{(j)} - \mu_{i_1}^{(\ell)})^2}, \quad 1 \le \ell \le q, \quad 1 \le j \le m \tag{4.34}$$

where q stands for the total number of testing nodes while υ and μ present training and testing nodes in the original space. Finally, the index node of the considered testing node is calculated using the index node of the most similar training node through the following

relation

$$\vartheta_{i_1}^{(\ell)} = \xi_{i_1}^{(app)} + \left(\mu_{i_1}^{(\ell)} - \upsilon_{i_1}^{(\ell)}\right)\left(\frac{n_{i_1-1}}{\eta_{i_1-1}}\right), \quad 1 \leq i_1 \leq N, \quad 1 \leq \ell \leq q \tag{4.35}$$

where $\xi_{i_1}^{(app)}$, $\mu_{i_1}^{(\ell)}$, $\upsilon_{i_1}^{(\ell)}$, n_{i_1} and η_{i_1} stand for the corresponding index space node of the appropriate training node, the testing node under consideration, the appropriate training node, the total number of different values that index space parameters can take and the total number of different values that the original space parameters can take in the problem respectively. Replacing the parameter values of this index node of the testing node in the model obtained through HDMR results in finding the class information of the considered testing node.

4.1.11 Evaluation

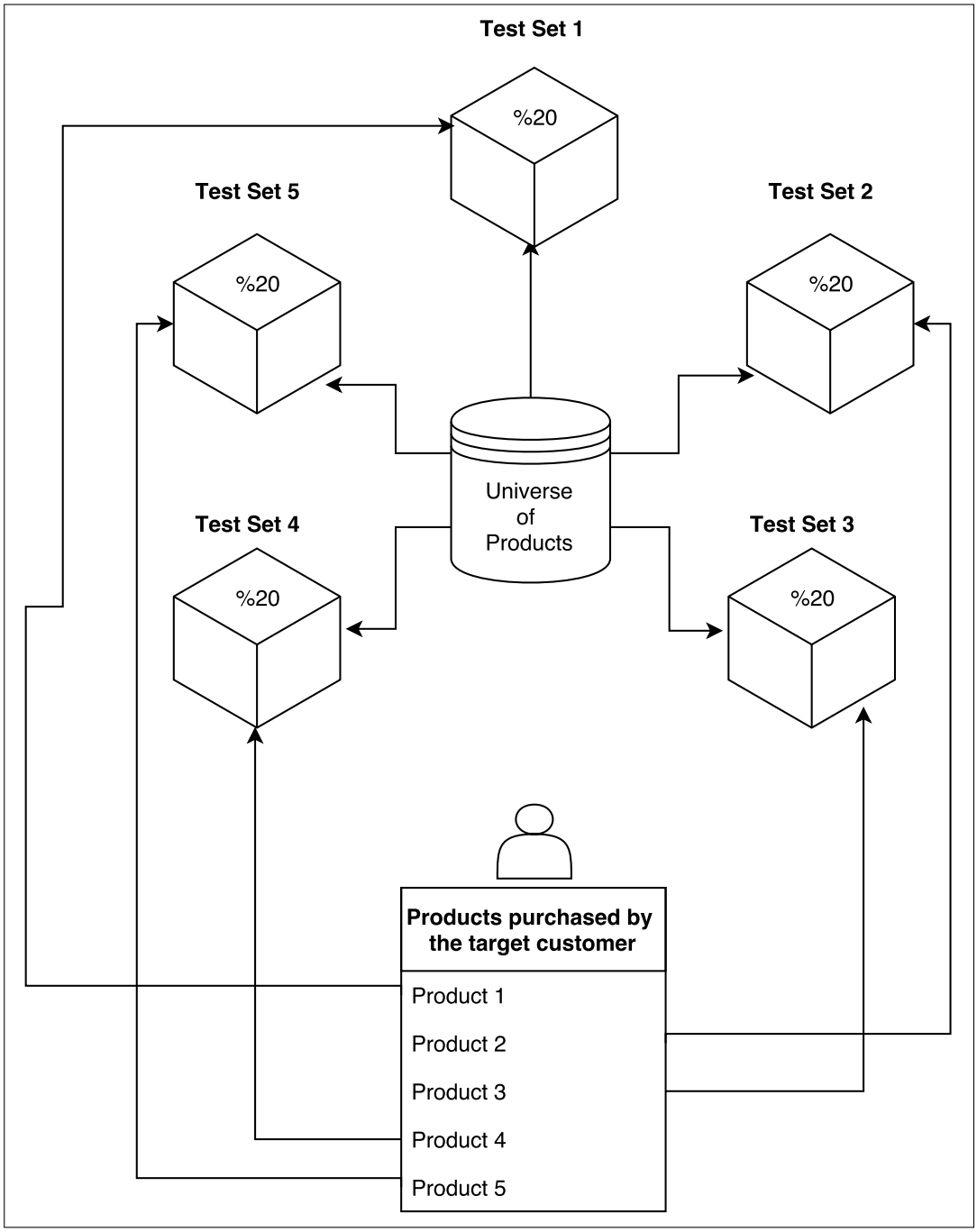
To evaluate the proposed model, similar strategy with (Aytekin & Karakaya, 2014) and (Cremonesi et al., 2010), which is to limit the domain of recommendations, was followed. For each customer, we held out five products that the customer has purchased before and we tried to recommend those products. Each customer has five test sets in addition to the training set explained in Section 5. Rather than letting the recommendation system to recommend from the entire universe of products, we have the recommendation system recommend from a subset of these held out five products, plus randomly selected $\%20$ of the products that the customer had not purchased before as shown on Figure 4.3. For instance, if there are 1000 products that the customer had not purchased before, each test set contains 200 randomly selected products plus 1 held out product.

We generated different top-N recommendation lists with the mentioned technique. If the target product is in this top-N list, it will be considered as a hit, otherwise not hit. Changes of getting a hit, increases with N. At the end, number of hits divided by five will give us the recall and diving the it by five times N will give us the precision measure as shown in Equations (4.36) and (4.37) respectively.

$$Recall(top-N) = \frac{Number\ of\ Hits}{5} \tag{4.36}$$

$$Precision(top-N) = \frac{Number\ of\ Hits}{5 \cdot N} \tag{4.37}$$

Figure 4.3: Evaluation process



F1 scores are also calculated to measure the accuracy of tests from another angle in terms of precision and recall as follows

$$F_1 = 2 \cdot \frac{1}{\frac{1}{recall} + \frac{1}{precision}} = \frac{precision \cdot recall}{precision + recall}. \tag{4.38}$$

For a single test case, recall can take either 0 or 1. Similarly, precision can assume the value 0 or $1/N$. To be more specific, recall is used to calculate the ratio of number of purchase hits to the number of all purchases for different top-N lists. However, as the recall and precision cannot be a standalone metric to be used to compare algorithms, Mean absolute error (MAE) and Root Mean Square Error (RMSE) values were also calculated as accuracy metrics as mentioned in (Beel et al., 2016). MAE is used to measure how close predictions are to the observations. In the test sets for each customer, there are 5 products that we know the real purchase quantity. Besides, we also know the predicted purchase quantity for these product by R-HDMR. Using these values, prediction errors (the differences in absolute value) for these 5 products are calculated for each customer and the sum of these prediction errors are averaged by the number of rows in the purchase history matrix to find the MAE. MAE is given by

$$MAE = \frac{1}{n} \sum_{i_1=1}^{n} \sum_{i_2=1}^{5} |e_i| \tag{4.39}$$

where n is the number of customers and e_i is the prediction error. Similarly, RMSE values are calculated according to relation given in Equation (4.40). These accuracy and evaluation metrics are calculated for R-HDMR and each of the compared algorithms separately.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i_1=1}^{n} \sum_{i_2=1}^{5} (e_i)^2} \tag{4.40}$$

5. RECOMMENDATION FRAMEWORK

The proposed method of this study combines the collaborative filtering approach and Indexing HDMR philosophy to make recommendations using an analytical structure as the data model based on the purchase history matrix of the customers. This new methodology is called "Recommender HDMR".

The preparation and organization of data has the highest priority in Recommender HDMR (R-HDMR). HDMR based methods can produce only one output for a single set of inputs at a time during the modeling process. Therefore, a purchase history matrix and accompanying purchase quantity vector are generated to organize the data of each customer.

The data organization is based on collaborative filtering approach. Finding similar customers to the target customer while making personalized recommendations is the basic issue. For this purpose, each customer is represented with an M-dimensional purchase quantities history vector (Q vector) where M is the number of unique products. If the customer has purchased a product, the purchase quantity is assigned to the corresponding element of this vector. If the customer did not purchase the product, then the related component becomes zero. The two most similar customers to the target customer are determined by applying the cosine metric given in Relation (4.3) to the Q vectors. The purchase history matrices (C) and accompanying purchase quantity vectors (φ) of these two similar customers are added to the corresponding matrix and vector of the target customer to be used in the modeling process through R-HDMR.

The modeling part includes the implementation of the Indexing HDMR method to construct an analytical structure from this extended purchase history matrix. To generate a recommendation list for the target customer, the parameter values of the products not included in that matrix are inserted into the obtained analytical structure. The output of this step is a value for each non-purchased product. The product with the highest value is the top product in the list. To this end, a top-N list can be created using this list.

Several numerical illustrations and the algorithm of the R-HDMR method as a recommendation system is given step by step as follows:

a) Specify the number of customers which is n.

b) Build Q (purchase quantities history) vector of each customer. Let's assume that in the data we have 23 unique products. The target customers' Q vector is shown below where elements of this vector represent the purchase quantity of the corresponding item by the target customer.

$$Q_{target} = [0, 1, 0, 0, 2, 0, 0, 4, 0, 3, 2, 0, 0, 0, 0, 3, 0, 1, 3, 0, 0, 0, 2] \tag{5.1}$$

c) Build C (purchase history) matrix and φ (purchase quantity) vector of each customer.

Consider a customer's purchase history matrix shown below. Each row of C_{target} represents the items purchased by the customer and φ represents the number of purchases of each item by that customer.

$$C_{target} = \begin{bmatrix} 3 & 8 & 1 & 1 \\ 1 & 1 & 5 & 1 \\ 3 & 6 & 6 & 1 \\ 3 & 8 & 7 & 1 \\ 3 & 8 & 11 & 1 \\ 3 & 8 & 14 & 1 \\ 1 & 6 & 2 & 2 \\ 8 & 8 & 2 & 2 \\ 13 & 8 & 4 & 2 \end{bmatrix}, \quad \varphi = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 3 \\ 2 \\ 3 \\ 1 \\ 3 \\ 2 \end{bmatrix} \tag{5.2}$$

d) Find the similar customers to the target customer using the cosine similarity given in Equation (4.3) through these Q vectors.

The most two similar customers' purchase history matrices are shown below.

$$C_{similar1} = \begin{bmatrix} 4 & 8 & 5 & 2 \\ 11 & 8 & 5 & 2 \\ 3 & 12 & 10 & 2 \\ 3 & 12 & 11 & 2 \end{bmatrix}, \quad \varphi = \begin{bmatrix} 2 \\ 2 \\ 4 \\ 3 \end{bmatrix} \tag{5.3}$$

$$C_{similar2} = \begin{bmatrix} 3 & 12 & 2 & 3 \\ 4 & 17 & 2 & 3 \\ 4 & 17 & 2 & 4 \end{bmatrix}, \quad \varphi = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} \tag{5.4}$$

e) Construct the extended purchase history matrix by combining the target customer's purchase history with the two most similar customers' history to create the training set.

C_{target}, $C_{similar1}$ and $C_{similar2}$ are combined in order to obtain target customers' extended purchase history matrix. The resulted matrix is shown below. This matrix will be used as the training set in the modelling process.

$$C_{extended} = \begin{bmatrix} 3 & 8 & 1 & 1 \\ 1 & 1 & 5 & 1 \\ 3 & 6 & 6 & 1 \\ 3 & 8 & 7 & 1 \\ 3 & 8 & 11 & 1 \\ 3 & 8 & 14 & 1 \\ 1 & 6 & 2 & 2 \\ 8 & 8 & 2 & 2 \\ 13 & 8 & 4 & 2 \\ 4 & 8 & 5 & 2 \\ 11 & 8 & 5 & 2 \\ 3 & 12 & 10 & 2 \\ 3 & 12 & 11 & 2 \\ 3 & 12 & 2 & 3 \\ 4 & 17 & 2 & 3 \\ 4 & 17 & 2 & 4 \end{bmatrix}, \quad \varphi = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 3 \\ 2 \\ 3 \\ 1 \\ 3 \\ 2 \\ 2 \\ 2 \\ 4 \\ 3 \\ 2 \\ 3 \\ 1 \end{bmatrix} \tag{5.5}$$

f) Specify the number of attributes for the given problem which is N. This data set has 4 independent variables ($N = 4$) and the domains of these variables are given as follows.

$$x_1 \in \{1, 3, 4, 8, 11, 13\}, \quad x_2 \in \{1, 6, 7, 8, 9, 12, 17\},$$
$$x_3 \in \{1, 2, 4, 5, 6, 710, 11, 14\}, \quad x_4 \in \{1, 2, 3, 4\} \tag{5.6}$$

g) Specify the number of nodes for the training data set which is m.

For this example, the number of nodes for the training data set is 16 ($m = 16$).

h) Evaluate the prime factors of m as shown in Relation (4.31). The number of prime factors of m should best fit N (Tunga, 2011).

$$m = 16 = 2 \times 2 \times 2 \times 2 \tag{5.7}$$

i) Specify the index set for each independent variable (attribute) as it is given in Relation (4.32).

Each prime factor corresponds to the number of elements of each index set and the index set for each independent variable is as follows.

$$\xi_1 \in \{1,2\}, \quad \xi_2 \in \{1,2\}, \quad \xi_3 \in \{1,2\}, \quad \xi_4 \in \{1,2\} \tag{5.8}$$

j) Construct a cartesian product set using the index sets considering the Equation (4.18).

$$CartesianSet_index = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 2 & 2 \\ 1 & 2 & 1 & 1 \\ 1 & 2 & 1 & 2 \\ 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 \\ 2 & 1 & 1 & 2 \\ 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 2 \\ 2 & 2 & 2 & 1 \\ 2 & 2 & 2 & 2 \end{bmatrix} \tag{5.9}$$

k) Sort the training set (extended purchase history matrix) by the function values in ascending order.

$$C_{sorted} = \begin{bmatrix} 3 & 8 & 1 & 1 \\ 1 & 6 & 2 & 2 \\ 4 & 17 & 2 & 4 \\ 1 & 1 & 5 & 1 \\ 3 & 8 & 11 & 1 \\ 13 & 8 & 4 & 2 \\ 4 & 8 & 5 & 2 \\ 11 & 8 & 5 & 2 \\ 3 & 12 & 2 & 3 \\ 3 & 8 & 7 & 1 \\ 3 & 8 & 14 & 1 \\ 8 & 8 & 2 & 2 \\ 3 & 12 & 11 & 2 \\ 4 & 17 & 2 & 3 \\ 3 & 6 & 6 & 1 \\ 3 & 12 & 10 & 2 \end{bmatrix}, \quad \varphi = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \\ 3 \\ 3 \\ 4 \\ 4 \end{bmatrix} \tag{5.10}$$

l) Create a one-to-one mapping between the sorted training data set and the cartesian product set created in Step j as shown in Relation (4.33).

$$\begin{aligned} (3,8,1,1,1) &\Rightarrow (1,1,1,1,1) \\ (1,1,5,1,2) &\Rightarrow (1,1,1,2,2) \\ &\vdots \\ (4,17,2,3,3) &\Rightarrow (2,2,2,1,3) \\ (4,17,2,4,1) &\Rightarrow (2,2,2,2,1) \end{aligned} \tag{5.11}$$

m) Calculate α values. Equation (4.23) indicates that the sum of these α values for each independent variable (attribute) is 1. If the contribution of each attribute in each node is assumed to be equal to each other, the first attribute has the α value as $1/n_1$ while it is $1/n_2$ for the second. It is also similar for the other attributes.

n) Evaluate the constant component using the Relation given in (4.22).

The constant term can be obtained by using this data and it is as follows.

$$f_0 = 2.4375 \tag{5.12}$$

o) Obtain the partitioned data sets through the Relations (4.24) and (4.25).

p) Construct an approximate analytical structure for the problem as it is given in relation (4.30).

If the Lagrange interpolation formula is used to determine the analytical structure of this univariate data, then the obtained structure can be added to the constant term and the HDMR expansion can be constructed approximately. This expansion is a representation for the unknown function by using constant and univariate data instead of whole multivariate data. The approximate analytical structure of the multivariate function obtained via HDMR is as follows.

$$\begin{aligned} f(x_1, x_2, x_3, x_4) = 1.75 \times x_1 + 6.5 \times x_2 + 2.0 \times x_3 + 2.0 \times x_4 - 0.75 \times x_1 \times \\ x_2 + 0.25 \times x_1 \times x_3 - 0.75 \times x_1 \times x_4 - 2.25 \times x_2 \times x_3 - 1.25 \times x_2 \times \\ x_4 + 0.75 \times x_3 \times x_4 - 6.9375 \end{aligned} \tag{5.13}$$

q) Apply the Euclidean distance metric given in Relation (4.34) between the given testing node (the product which the customer has not bought before) and the training nodes (the products which the customer has bought before) to specify the most similar training node to the considered testing node.

Lets assume that one of the testing node is $\{3, 8, 10, 1\}$. The distance between the testing node and the most similar training node and the index of the most similar training node is shown below.

$$\mu_1 = \{3, 8, 10, 1\}, \quad d = 3.3166, \quad index = 3 \tag{5.14}$$

r) Use the index node of the resulting training node and the relation given in Relation (4.35) to determine the index node of the testing node under consideration.

$$\begin{gathered}\gamma_1 = 0.2000, \quad \gamma_2 = 0.1667, \quad \gamma_3 = 0.1250, \quad \gamma_4 = 0.3333 \\ \vartheta_1^{(1)} = \xi_1^{(app)} + \gamma_1\left(\mu_1^{(1)} - \upsilon_1^{(app)}\right) = 1 \\ \vartheta_1^{(2)} = \xi_1^{(app)} + \gamma_2\left(\mu_1^{(2)} - \upsilon_1^{(app)}\right) = 1 \\ \vartheta_1^{(3)} = \xi_1^{(app)} + \gamma_3\left(\mu_1^{(3)} - \upsilon_1^{(app)}\right) = 2.3 \\ \vartheta_1^{(4)} = \xi_1^{(app)} + \gamma_4\left(\mu_1^{(4)} - \upsilon_1^{(app)}\right) = 2\end{gathered} \tag{5.15}$$

s) Replace the parameter values of the index node of the testing node in the structure obtained in Step q. This results in finding the function value of the given testing node. This function value is standing for the purchase quantity prediction of the product under test.

When we replace the parameter values of the index node of the testing node in the HDMR function found before, this results in finding the class information of the given testing node.

$$f\left(\vartheta_1^{(1)}, \vartheta_1^{(2)}, \vartheta_1^{(3)}, \vartheta_1^{(4)}\right) \approx 2.72 \tag{5.16}$$

Considering the test set shown below, we can found all of the class information of these given nodes. This test set contains seven products that the customer hadn't purchase yet and one product (the second node in the set) that he or she purchased

already.

$$TestSet = \begin{bmatrix} 3 & 8 & 10 & 1 \\ 3 & 6 & 6 & 1 \\ 13 & 7 & 7 & 1 \\ 3 & 12 & 1 & 3 \\ 1 & 8 & 7 & 2 \\ 4 & 9 & 10 & 2 \\ 4 & 12 & 1 & 4 \\ 13 & 8 & 2 & 2 \end{bmatrix}, \quad \varphi = \begin{bmatrix} 3 \\ 4 \\ 3 \\ 2 \\ 1 \\ 3 \\ 2 \\ 1 \end{bmatrix} \tag{5.17}$$

Similarly, we can found all of the class information of the given testing nodes.

$$\begin{gathered} f(3,6,6,1) \approx 3.75, \quad f(13,7,7,1) \approx 3.42, \quad f(3,12,1,3) \approx 2.40, \\ f(1,8,7,2) \approx 3.64, \quad f(4,9,10,2) \approx 3.12, \quad f(4,12,1,4) \approx 3.16, \\ f(13,8,2,2) \approx 1.30 \end{gathered} \tag{5.18}$$

Out of eight testing nodes, the test node with the highest class information is $\{3,6,6,1\}$. we know that this is the product that the customer purchased before. So, it can be said that, the algorithm succeeded to find the target product correctly for this customer using this test set. Once the approximate analytical structure for this customer is constructed, in order to evaluate the proposed model, five test sets have created. In each test set, there is one product that this target customer has purchased before and the rest of these sets are consist of the $\%20$ of the universe of products (the target customer had not purchased before). The aim is to be able to recommend these (already purchased) products to this customer.The evaluation process is explained in more detail in the Section 4.1.11.

Running these steps for all non-purchased products for the target customer helps us to generate a recommendation list.

After performing some tests, we decided to use normalized values for purchase history matrices. Using the same customer's purchase history, but with normalized values this time, R-HDMR method is used again. Normalized test and train sets are shown below.

$$TrainingSet = \begin{bmatrix} 0.154 & 0.110 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.25 & 0.0 \\ 0.154 & 0.078 & 0.313 & 0.0 \\ 0.154 & 0.110 & 0.375 & 0.0 \\ 0.154 & 0.110 & 0.625 & 0.0 \\ 0.154 & 0.110 & 0.813 & 0.0 \\ 0.0 & 0.078 & 0.063 & 0.077 \\ 0.538 & 0.110 & 0.063 & 0.077 \\ 0.923 & 0.110 & 0.186 & 0.077 \\ 0.231 & 0.110 & 0.25 & 0.077 \\ 0.769 & 0.110 & 0.25 & 0.077 \\ 0.154 & 0.172 & 0.563 & 0.077 \\ 0.154 & 0.172 & 0.625 & 0.077 \\ 0.154 & 0.172 & 0.063 & 0.154 \\ 0.231 & 0.25 & 0.063 & 0.154 \\ 0.231 & 0.25 & 0.063 & 0.231 \end{bmatrix}, \quad \varphi = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 3 \\ 2 \\ 3 \\ 1 \\ 3 \\ 2 \\ 2 \\ 2 \\ 4 \\ 3 \\ 2 \\ 3 \\ 1 \end{bmatrix} \tag{5.19}$$

$$TestSet = \begin{bmatrix} 0.154 & 0.110 & 0.563 & 0.0 \\ 0.154 & 0.078 & 0.313 & 0.0 \\ 0.923 & 0.094 & 0.375 & 0.0 \\ 0.154 & 0.172 & 0.0 & 0.154 \\ 0.0 & 0.110 & 0.375 & 0.077 \\ 0.231 & 0.125 & 0.563 & 0.077 \\ 0.231 & 0.172 & 0.0 & 0.231 \\ 0.923 & 0.110 & 0.063 & 0.077 \end{bmatrix}, \quad \varphi = \begin{bmatrix} 3 \\ 4 \\ 3 \\ 2 \\ 1 \\ 3 \\ 2 \\ 1 \end{bmatrix} \tag{5.20}$$

The same steps are taken into consideration as the previous example. Once the approximate analytical structure of the multivariate function obtained via HDMR, the Euclidean distance metric between the given testing node and the training nodes is applied. Then we replace the parameter values of the index nodes of the testing nodes in the HDMR function and find the class information of the given testing nodes.

6. FINDINGS

R-HDMR and several data mining techniques were tested on the data set for all customers for top-5, top-10, top-15, top-20 and top-25 recommendation lists. Recall and precision were used as accuracy metrics, RMSE and MAE were used as error metrics to measure the performance of these methods.

The best recall value was 100% which means that all the target products for the considered customer were successfully recommended by R-HDMR. The worst recall value was 0% for the considered recommendation lists. That is, none of the target products could be recommended to that customer by R-HDMR. On the other hand, the average recall and MAE values are given in Table 6.1. The best value obtained through the execution of the methods on the same data set is given in boldface fonts in Table 6.1.

Table 6.1: Average performance values for the recommendation lists prepared for all customers.

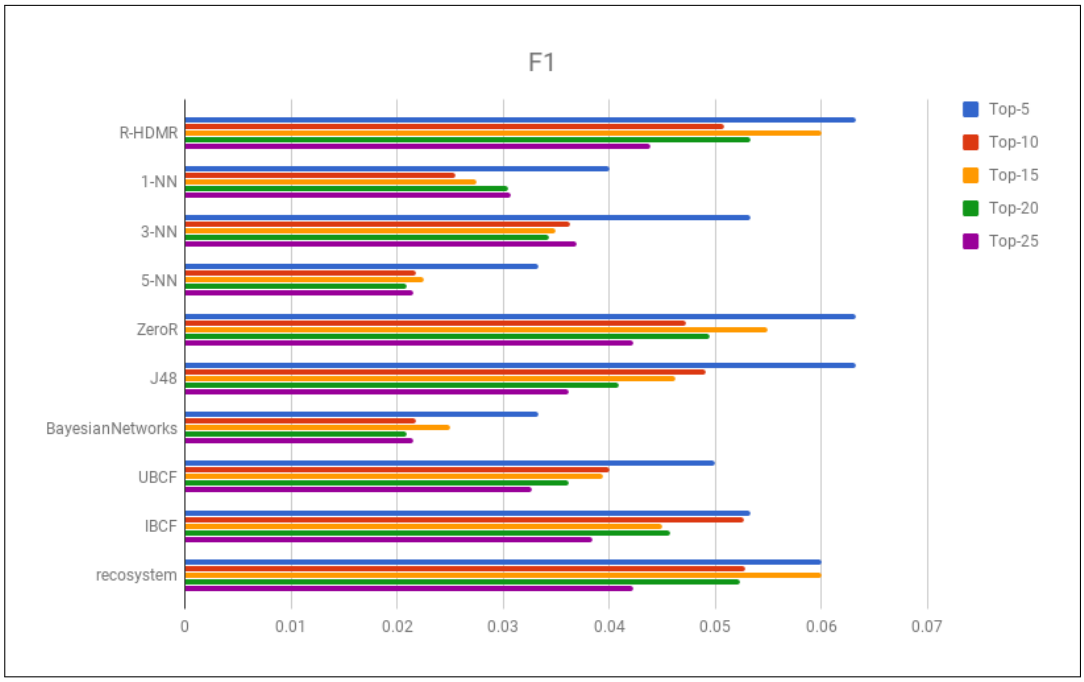
Methods	MAE	Top-5	Top-10	Top-15	Top-20	Top-25
R-HDMR	**0.10**	**19**%	**28**%	**48**%	**56**%	**57**%
1NN	0.21	12%	14%	22%	32%	40%
3NN	0.17	16%	20%	28%	36%	48%
5NN	0.37	10%	12%	18%	22%	28%
ZeroR	0.15	**19**%	26%	44%	52%	55%
J48	0.18	**19**%	27%	37%	43%	47%
Bayesian	0.29	10%	12%	20%	22%	28%
UBCF	0.22	15%	22%	32%	38%	44%
IBCF	0.19	16%	29%	36%	48%	50%
LIBMF-BMF	0.24	18%	**30**%	**48**%	55%	55%

The ideal MAE is zero. MAE shows how big of an error can be expected from the prediction on average. MAE value of R-HDMR was obtained as 0.10. Since it is very close to zero, it can be said that the system makes considerably small errors in the recommendation process. Table 6.1 also includes the performance results of some other well known data mining techniques used as a recommendation system. These methods are K-nearest-neighbor (1NN, 3NN, 5NN), ZeroR, decision tree (J48 was used) and Bayesian Network

based algorithms of the tool called WEKA version 3.6.13 (Frank & Witten, 2005; Hall et al., 2009).

All methods were implemented using the same customers with same purchase history matrices. In addition to these basic algorithms, User-Based Collaborative Filtering (UBCF) and Item-Based Collaborative Filtering (IBCF) algorithms of Apache Mahout, which is a open source machine learning library that provides tools for making recommendations (Ghodake et al., 2016), and recosystem, a R wrapper of an open source library for recommendation system using parallel marix factorization called LIBMF was also tested. LIBMF is a C++ library for large scale matrix factorization and since recosystem is a wrapper of LIBMF, it inherits the features of this library. UBCF and IBCF algorithms are implemented in Eclipse Mars IDE (Integrated Development Environment) using Java as the programming language whereas recosystem is implemented in RStudio version 1.0.136.

Figure 6.1: F1 scores



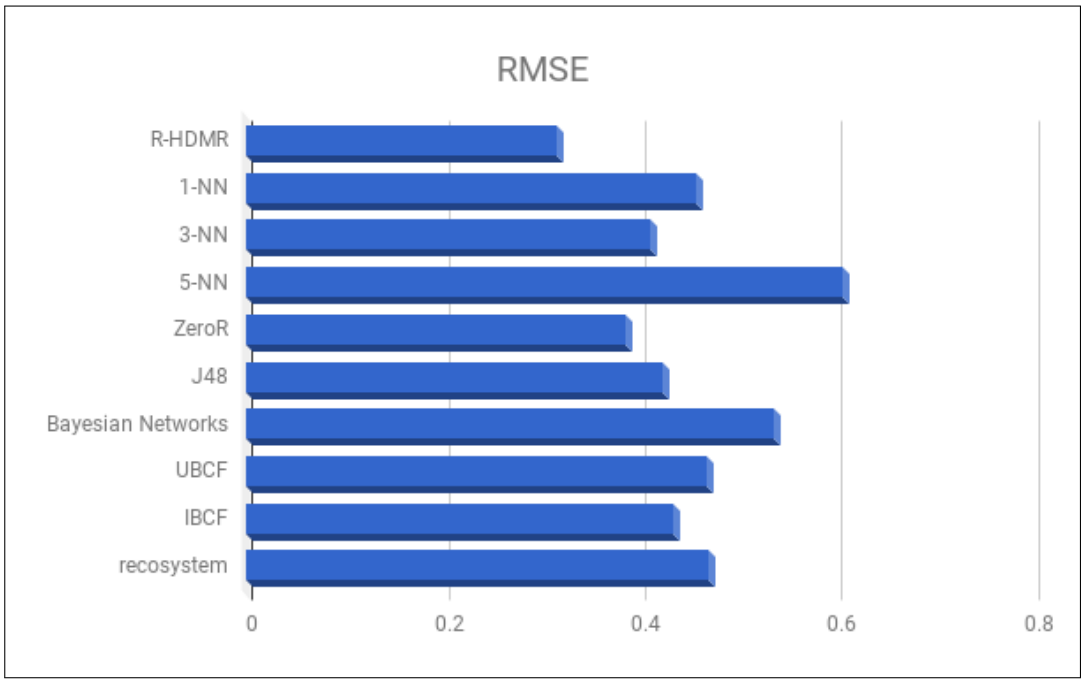
The reported results show that the R-HDMR method obtained the lowest MAE result while 1NN, 3NN, 5NN, ZeroR, J48 and Bayesian Network obtained 0.21, 0.17, 0.37,

0.15, 0.18 and 0.29 and more advanced UBCF, IBCF and recosystem methods obtained 0.22, 0.19 and 0.24 MAE values respectively.

F1 scores are shown in Figure 6.1. The larger the value of F1 score gets, the better the corresponding result. Figure 6.1 presents the power of R-HDMR in constructing top-N lists. As seen in the figure, R-HDMR method has the biggest F1 scores in most cases which can simply be interpreted as our method works better than the advanced models of the literature.

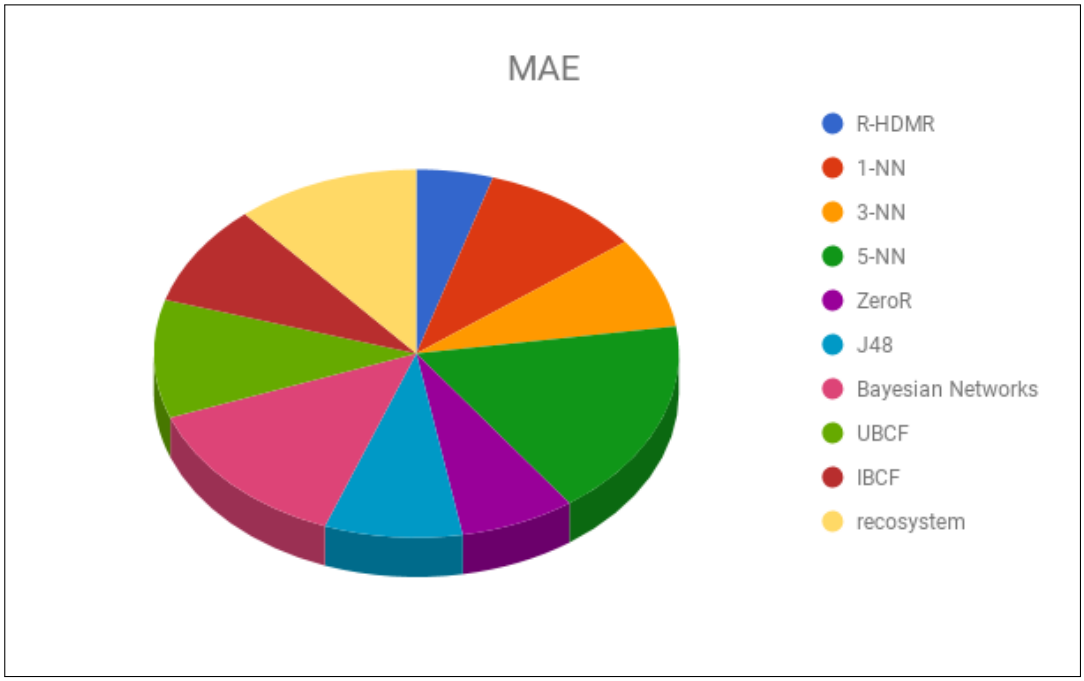
Figure 6.2 demonstrates the RMSE comparison of algorithms which further strengthens our findings. RMSE (also called the root mean square deviation) is a frequently used measure for Top-N recommendations which the difference between values predicted by the model under consideration and the values actually observed (Cremonesi et al., 2010) and evidently, from the figure it can be seen that R-HDMR has the lowest RMSE whereas 5-NN has the highest followed by the Bayesian networks. The other algorithms have RMSE values close to each other.

Figure 6.2: RMSE comparison



Moreover, Figure 6.3 graphically compares the R-HDMR method with above mentioned algorithms in terms of MAE while Figure 6.4 reports the performance of the algorithms on the data set over the full test sets. While 5-NN has the biggest MAE value, R-HDMR and Zero-R has the lowest MAE respectively. Figure 6.4 shows that, R-HDMR has a lower recall value for top-5 recommendation list and this value increases nearly up to four times for top-25 recommendation list. From both of the figures, it is apparent that R-HDMR has a significant performance disparity in terms of top-N accuracy comparing to most of the algorithms.

Figure 6.3: MAE comparison



Additionally, Figure 6.5 confirms that R-HDMR outperforms the other algorithms in terms of precision metric. Since the proposed approach is a recommendation system, we are looking to find the most relevant products for the customers, while trying to minimize the worthless products that are retrieved by the system. In this figure, each line represents the precision of the related algorithm at a given recall value. Typically, Precision and Recall are inversely related metrics and Figure 6.5 clearly shows that R-HDMR is the most appropriate algorithm can be chosen for our case and again 5-NN has the poorest results for this data set.

Figure 6.4: Recall comparison

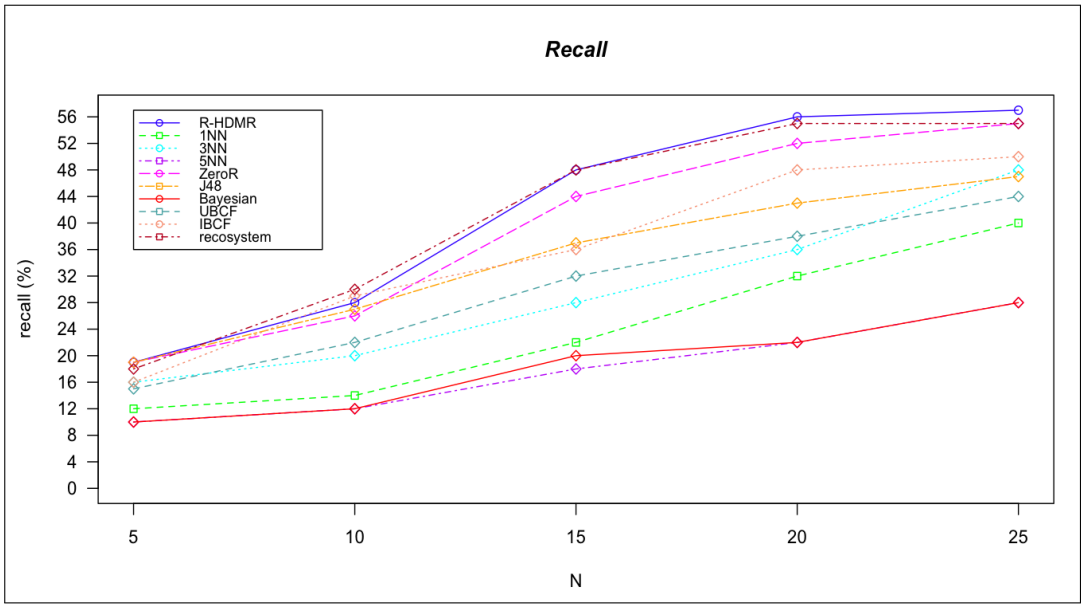
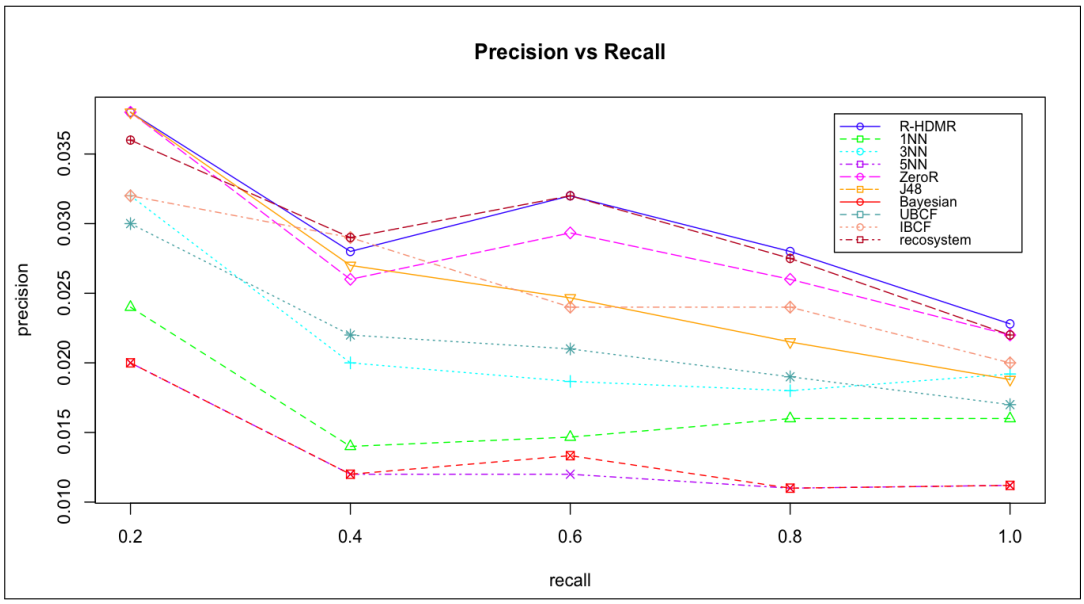


Figure 6.5: Precision vs Recall



In brief, considering these experimental outcomes, it can be said that R-HDMR is able to make better and more reliable recommendations rather than aforementioned algorithms. R-HDMR has the highest recall and lowest MAE and RMSE values among the other methods while Bayesian Network and 5NN have the lowest. This can be explained as

some of these algorithms use only neighborhood information, some of them are not interested in features of the products and some of them are only based on similarity metrics whereas R-HDMR uses the combination of all of these aspects.

7. CONCLUSION

We propose an interpolation-based recommendation system approach that exploits collaborative filtering and high dimensional model representation. The proposed method represents the high dimensional data with lower dimensions using HDMR and generate recommendations using the collaborative filtering philosophy and the Lagrange interpolation. We evaluate the performance of the proposed method on a real data set from apparels domain which have not been used for recommendation. We also perform experiments with pioneering recommendation approaches in the literature. The quantitative comparisons of the proposed approach with the state-of-the-art algorithms demonstrate that our approach generates the most accurate recommendation lists (top-5, top-10, top-15, top-20, top-25) for a target customer.

It is well known that KNN only searches for the closest k transactions, ZeroR uses only the class values and Bayesian Network uses independence assumptions between predictors to make predictions. Apache Mahouts' UBCF and IBCF algorithms are mostly based on similarity metrics and neighborhood development and recosystem uses matrix factorization while making recommendations. On the other hand, R-HDMR takes all related data into consideration to construct an analytical model to make personalized recommendations. This results in better learning the target customer's behaviors which means recommending more accurate lists. The numerical results obtained by executing R-HDMR and these techniques also show that R-HDMR works better than these well known methods.

Furthermore, there are some limitations of R-HDMR as well as good performance. At first, the data to be used must be in a specific format. Each item should have equal number of features available. Second, since R-HDMR is an interpolation-based algorithm, it can produce only one output. If multiple outputs are needed, the algorithm should be executed iteratively.

What we are trying to do is ultimately a recommendation system and since we are trying

to model human behaviour, this makes it more complex. In the evaluation process, not recommending the target product does not necessarily mean that we are making a mistake. Instead, it could mean that we found a product that the customer is not aware of and he might actually purchase it. In the data preparation step, we normalize the given data set to ensure that it has the characteristics of a normal distribution to minimize mistakes based on the distribution of the given dataset.

In particular, this thesis makes four main contributions to the area of recommendation systems. These contributions are as follows:

a) We introduce a re-contextualization of the HDMR method by applying this technique in recommendation generation context.

b) We show that HDMR method is applicable for personalized recommendations

c) We combined HDMR and collaborative filtering philosophy, that had never been combined before, to reveal a hybrid, new and useful recommendation system.

d) We apply the proposed framework to a novel data set that has never been used before

We believe that R-HDMR, a new top-N recommender algorithm, can be improved by combining aspects from both the collaborative and content-based filtering approaches. For truly personalized recommendations, only collaborative-based (computing neighborhood of similar users) approaches were used in this thesis. Inserting content-based (item-based) approaches to the current algorithm might strengthen the accuracy of recommendations by computing similarity between products. However, the industrial data used in this work does not allow us to design such algorithm. We may construct another recommendation algorithm with a different data set in the future. As another future work, since HDMR philosophy has the ability to discretize multivariate data and constructs independent data sets with different levels of multivariance, R-HDMR can also be redesigned to develop data models for big data.

REFERENCES

Books

Bandyopadhyay, S. & Saha, S., 2012. *Unsupervised classification: similarity measures, classical and metaheuristic approaches, and applications*. Springer Science & Business Media.

Brusilovski, P., Kobsa, A., & Nejdl, W., 2007. *The adaptive web: methods and strategies of web personalization*, vol. 4321. Springer Science & Business Media.

Frank, E. & Witten, I. H., 2005. *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann.

Ricci, F., Rokach, L., & Shapira, B., 2011. *Introduction to recommender systems handbook*. Springer.

Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B., 2015. *Recommender systems handbook*. Springer.

Periodicals

Akkemik, E. & Demiralp, M., 2003. Algebraic eigenvalue problem modelling via high dimensional model representation. *Math. Res* **9**, pp. 5–18.

Alış, Ö. F. & Rabitz, H., 2001. Efficient implementation of high dimensional model representations. *Journal of Mathematical Chemistry* **29(2)**, pp. 127–142.

Alper Tunga, M. & Demiralp, M., 2004. A factorized high dimensional model representation on the partitioned random discrete data. *Applied Numerical Analysis & Computational Mathematics* **1(1)**, pp. 231–241.

Aytekin, T. & Karakaya, M. Ö., 2014. Clustering-based diversity improvement in top-n recommendation. *Journal of Intelligent Information Systems* **42(1)**, pp. 1–18.

Balu, A. S. & Rao, B. N., 2014. Confidence bounds on design variables using high-dimensional model representation-based inverse reliability analysis. *Journal of Structural Engineering-ASCE* **139**, pp. 985–996.

Baykara, N. & Demiralp, M., 2003. Hyperspherical or hyperellipsoidal coordinates in the evaluation of high dimensional model representation approximants. *Mathematical Research* **9**, pp. 48–62.

Beel, J., Gipp, B., Langer, S., & Breitinger, C., 2016. paper recommender systems: a literature survey. *International Journal on Digital Libraries* **17(4)**, pp. 305–338.

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A., 2013. Recommender systems survey. *Knowledge-based systems* **46**, pp. 109–132.

Buder, J. & Schwind, C., 2012. Learning with personalized recommender systems: A psychological view. *Computers in Human Behavior* **28(1)**, pp. 207–216.

Campos, P. G., Díez, F., & Cantador, I., 2014. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction* **24(1-2)**, pp. 67–119.

Chen, L.-S., Hsu, F.-H., Chen, M.-C., & Hsu, Y.-C., 2008. Developing recommender systems with the consideration of product profitability for sellers. *Information Sciences* **178(4)**, pp. 1032–1048.

Civlekoğlu, T. & Demiralp, M., 2003. An hdmr application to the schrödinger's equation for free particles under an external field with dipole polarization and vanishing flux boundary conditions. *Math. Res* **9**, pp. 110–121.

Cooling, C. M., Ayres, D. A. F., Prinja, A. K., & Eaton, M. D., 2016. Uncertainty and global sensitivity analysis of neutron survival and extinction probabilities using polynomial chaos. *Annals of Nuclear Energy* **88**, pp. 158–173.

Cunningham, P., Bergmann, R., Schmitt, S., Traphöner, R., Breen, S., & Smyth, B., 2001. Websell: Intelligent sales assistants for the world wide web. *KI* **15(1)**, pp. 28–32.

De Campos, L. M., Fernández-Luna, J. M., Huete, J. F., & Rueda-Morales, M. A., 2010. Combining content-based and collaborative recommendations: A hybrid approach based on bayesian networks. *International Journal of Approximate Reasoning* **51(7)**, pp. 785–799.

Demiralp, E. & Tunga, M., 2003. A hybrid programming for projective displaying of high dimensional model representation approximants. *Math. Res* **9**, pp. 132–145.

Demiralp, M., 2003. High dimensional model representation and its application varieties. *Math. Res* **9**, pp. 146–159.

Demiralp, M. & Tunga, B., 2015. Weight optimization in hdmr with perturbation expansion method. *Journal of Mathematical Chemistry* **53(10)**, pp. 2155–2171.

Demiralp, M. & Tunga, M., 2001. High dimensional model representation of multivariate interpolation via hypergrids. In *The Sixteenth international symposium on computer and information sciences (ISCIS XVI)*. pp. 416–423.

Fang, G. C., Lu, Z. Z., & Cheng, L., 2015. A new methodology based on covariance and hdmr for global sensitivity analysis. *Applied Mathematical Modelling* **39**, pp. 5399–5414.

Fırat, B., Şenol, N., & Demiralp, M., 2003. Hyperpolyhedral weight construction through high dimensional model representation (hdmr): a hyperrotation based application. *Math. Res* **9**, pp. 183–192.

Fournier, F., 2011. Recommender systems technical report and literature review. *On http://knol. google. com/k/recommender-systems#, Last accessed on 8th September* .

Galland, A. & Cautis, B., 2010. Recommender systems. *at INRIA-Saclay, http://alban. galland. free. fr/Documents/Enseignements/INF396/recom mendersystems-slides. pdf, INRIA-Saclay* **18**.

Gavalas, D., Konstantopoulos, C., Mastakas, K., & Pantziou, G., 2014. Mobile recommender systems in tourism. *Journal of Network and Computer Applications* **39**, pp. 319–333.

Ghodake, S., Koren, B., & Paswan, R., 2016. Survey on recommender system using distributed framework. *International Journal of Science and Research* **5(1)**.

Golbeck, J., 2009. Trust and nuanced profile similarity in online social networks. *ACM Transactions on the Web (TWEB)* **3(4)**, p. 12.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H., 2009. The weka data mining software: An update. *SIGKDD Explorations* **11(1)**, pp. 10–18.

Hu, Z. Z., Smith, R. C., Willert, J., & Kelley, C. T., 2014. High-dimensional model representations for the neutron transport equation. *Nuclear Science and Engineering* **177**, pp. 350–360.

Huang, Z. Y., Qiu, H. B., Zhao, M., Cai, X. W., & Gao, L., 2015. An adaptive svr-hdmr model for approximating high dimensional problems. *Engineering Computations* **32**, pp. 643–667.

Kaman, T. & Demiralp, M., 2003. A high dimensional model representation application to the optimal control of harmonic oscillator. *Math. Res* **9**, pp. 255–269.

Kaman, T. & Demiralp, M., 2004. A parametric sensitivity analysis for the solution of extrema evaluation problems via a dimensionality reducing approximation method. *Applied Numerical Analysis & Computational Mathematics* **1(1)**, pp. 260–269.

Kanal, M. E. & Demiralp, M., 2012. A modified method of calculating high dimensional model representation (hdmr) terms for parallelization with mpi and cuda. *Journal of Supercomputing* **62(1)**, pp. 199–213.

Kanmaz, A. A. & Demiralp, M., 2003. Symbolic computer programming for generalized high dimensional model representation. *Math. Res* **9**, pp. 281–289.

Konstan, J. A. & Riedl, J., 2012. Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction* **22(1-2)**, pp. 101–123.

Kurşunlu, A. & Demiralp, M., 2003. Additive and factorized high dimensional model representation applications to the multivariate diffusion equation under vanishing derivative boundary conditions. *Math. Res* **9**, pp. 315–327.

Li, G., Artamonov, M., Rabitz, H., Wang, S.-w., Georgopoulos, P. G., & Demiralp, M., 2003a. High-dimensional model representations generated from low order terms—lp-rs-hdmr. *Journal of computational chemistry* **24(5)**, pp. 647–656.

Li, G., Rabitz, H., Wang, S.-W., & Georgopoulos, P. G., 2003b. Correlation method for variance reduction of monte carlo integration in rs-hdmr. *Journal of computational chemistry* **24(3)**, pp. 277–283.

Li, G., Rosenthal, C., & Rabitz, H., 2001a. High dimensional model representations. *The Journal of Physical Chemistry A* **105(33)**, pp. 7765–7777.

Li, G., Schoendorf, J., Ho, T.-S., & Rabitz, H., 2004. Multicut-hdmr with an application to an ionospheric model. *Journal of computational chemistry* **25(9)**, pp. 1149–1156.

Li, G., Wang, S.-W., & Rabitz, H., 2002a. Practical approaches to construct rs-hdmr component functions. *The Journal of Physical Chemistry A* **106(37)**, pp. 8721–8733.

Li, G., Wang, S.-W., Rabitz, H., Wang, S., & Jaffé, P., 2002b. Global uncertainty assessments by high dimensional model representations (hdmr). *Chemical Engineering Science* **57(21)**, pp. 4445–4460.

Li, G., Wang, S.-W., Rosenthal, C., & Rabitz, H., 2001b. High dimensional model representations generated from low dimensional data samples. i. mp-cut-hdmr. *Journal of Mathematical Chemistry* **30(1)**, pp. 1–30.

Li, G. Y., Bastian, C., Welsh, W., & Rabitz, H., 2015. Experimental design of formulations utilizing high dimensional model representation. *Journal of Physical Chemistry A* **119(29)**, pp. 8237–8249.

Li, G. Y. & Rabitz, H., 2012. General formulation of hdmr component functions with independent and correlated variables. *Journal of Mathematical Chemistry* **50(1)**, pp. 99–130.

Li, G. Y. & Rabitz, H., 2014. Analytical hdmr formulas for functions expressed as quadratic polynomials with a multivariate normal distribution. *Journal of Mathematical Chemistry* **52(8)**, pp. 2052–2073.

Lieberman, H. et al., 1995. Letizia: An agent that assists web browsing. *IJCAI (1)* **1995**, pp. 924–929.

Linden, G., Smith, B., & York, J., 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* **7(1)**, pp. 76–80.

Moradi, P. & Ahmadian, S., 2015. A reliability-based recommendation method to improve trust-aware recommender systems. *Expert Systems with Applications* **42(21)**, pp. 7386–7398.

Moreno, M. N., Segrera, S., López, V. F., Muñoz, M. D., & Sánchez, Á. L., 2016. Web mining based framework for solving usual problems in recommender systems. a case study for movies' recommendation. *Neurocomputing* **176**, pp. 72–80.

Munsell, A. H. et al., 1950. Munsell book of color .

Özay, E. K. & Demiralp, M., 2014. Reductive enhanced multivariance product representation for multi-way arrays. *Journal of Mathematical Chemistry* **52(10)**, pp. 2546–2558.

Park, D. H., Kim, H. K., Choi, I. Y., & Kim, J. K., 2012. A literature review and classification of recommender systems research. *Expert Systems with Applications* **39(11)**, pp. 10059–10072.

Pazzani, M. J., 1999. A framework for collaborative, content-based and demographic filtering. *Artificial intelligence review* **13(5-6)**, pp. 393–408.

Polatidis, N. & Georgiadis, C. K., 2016. A multi-level collaborative filtering method that improves recommendations. *Expert Systems with Applications* **48**, pp. 100–110.

Porcel, C. & Herrera-Viedma, E., 2010. Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries. *Knowledge-Based Systems* **23(1)**, pp. 32–39.

Rabitz, H. & Aliş, Ö. F., 1999. General foundations of high-dimensional model representations. *Journal of Mathematical Chemistry* **25(2-3)**, pp. 197–233.

Rabitz, H., Aliş, Ö. F., Shorter, J., & Shim, K., 1999. Efficient input—output model representations. *Computer Physics Communications* **117(1)**, pp. 11–20.

Selmi, A., Brahmi, Z., & Gammoudi, M. M., 2016. Trust-based recommender systems: An overview .

Şenol, N., Fırat, B., & Demiralp, M., 2003. Hyperpolyhedral weight construction through high dimensional model representation: a laplace transform application. *Math. Res* **9**, pp. 366–375.

Shinde, S. K. & Kulkarni, U., 2012. Hybrid personalized recommender system using centering-bunching based clustering algorithm. *Expert Systems with Applications* **39(1)**, pp. 1381–1387.

Shorter, J. A., Ip, P. C., & Rabitz, H. A., 1999. An efficient chemical kinetics solver using high dimensional model representation. *The Journal of Physical Chemistry A* **103(36)**, pp. 7192–7198.

Sobol, I. M., 1993. Sensitivity estimates for nonlinear mathematical models. *Mathematical Modeling and Computational Experiments* **1**, pp. 407–414.

Tunga, B. & Demiralp, M., 2003a. Hybrid high dimensional model representation approximants and their utilization in applications. *Math. Res* **9**, pp. 438–446.

Tunga, B. & Demiralp, M., 2012a. Hybrid hdmr method with an optimized hybridity parameter in multivariate function representation. *Journal of Mathematical Chemistry* **50(8)**, pp. 2223–2238.

Tunga, B. & Demiralp, M., 2012b. A novel approximation method for multivariate data partitioning fluctuation free integration based hdmr. *Engineering Computations* **29(7-8)**, pp. 743–765.

Tunga, M. A., 2011. An approximation method to model multivariate interpolation problems: Indexing hdmr. *Mathematical and Computer Modelling* **53(9)**, pp. 1970–1982.

Tunga, M. A. & Demiralp, M., 2003b. Data partitioning via generalized high dimensional model representation (ghdmr) and multivariate interpolative applications. *Math. Res* **9**, pp. 447–462.

Tunga, M. A. & Demiralp, M., 2006. Hybrid high dimensional model representation (hhdmr) on the partitioned data. *Journal of Computational and Applied Mathematics* **185(1)**, pp. 107–132.

Tunga, M. A. & Demıralp, M., 2008. A new approach for data partitioning through high dimensional model representation. *International Journal of Computer Mathematics* **85(12)**, pp. 1779–1792.

Tunga, M. A. & Demiralp, M., 2008. A new approach for data partitioning through high dimensional model representation. *International Journal of Computer Mathematics* **85(12)**, pp. 1779–1792.

Véras, D., Prota, T., Bispo, A., Prudêncio, R., & Ferraz, C., 2015. A literature review of recommender systems in the television domain. *Expert Systems with Applications* **42(22)**, pp. 9046–9076.

Wang, S.-W., Georgopoulos, P. G., Li, G., & Rabitz, H., 2003. Random sampling-high dimensional model representation (rs-hdmr) with nonuniformly distributed variables: application to an integrated multimedia/multipathway exposure and dose model for trichloroethylene. *The Journal of Physical Chemistry A* **107(23)**, pp. 4707–4716.

Yaman, İ. & Demiralp, M., 2003. High dimensional model representation applications to exponential matrix evaluation. *Math. Res* **9**, pp. 463–474.

Yaman, I. & Demiralp, M., 2004. High dimensional model representation approximation of an evolution operator with a first order partial differential operator argument. *Applied Numerical Analysis & Computational Mathematics* **1(1)**, pp. 280–289.

Zeng, C., XING, C.-X., & Zhou, L.-z., 2003. A personalized search algorithm by using content-based filtering [j]. *Journal of Software* **5**, p. 018.

Other Publications

Altin, E. M. & Tunga, B., 2014. High dimensional model representation in image processing. In *Proceedings of the 2014 International Conference on Computational and Mathematical Methods in Science and Engineering*. pp. 55–64.

Basu, C., Hirsh, H., Cohen, W. et al., 1998. Recommendation as classification: Using social and content-based information in recommendation. In *Aaai/iaai*. pp. 714–720.

Breese, J. S., Heckerman, D., & Kadie, C., 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pp. 43–52.

Condli, M. K., Lewis, D. D., Madigan, D., & Posse, C., 1999. Bayesian mixed-e ects models for recommender systems. In *ACM SIGIR*, vol. 99.

Cremonesi, P., Koren, Y., & Turrin, R., 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, pp. 39–46.

Demiralp, M., 2006. Transformational high dimensional model representation. In *ICCMSE-2006 (International Conference of Computational Methods in Sciences and Engineering)*.

Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F., Reiterer, S., & Stettinger, M., 2014. Basic approaches in recommendation systems. In *Recommendation Systems in Software Engineering*. Springer, pp. 15–37.

Ghazanfar, M. A. & Prugel-Bennett, A., 2010. A scalable, accurate hybrid recommender system. In *Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on*. IEEE, pp. 94–98.

Ho, Y., Fong, S., & Yan, Z., 2007. A hybrid ga-based collaborative filtering model for online recommenders. In *Ice-b*. pp. 200–203.

Karaca, E. & Tunga, M. A., 2016. Interpolation-based image inpainting in color images using high dimensional model representation. In *Proceedings of the 24th European Signal Processing Conference (EUSIPCO 2016)*. p. In Press.

Karatzoglou, A., Baltrunas, L., & Shi, Y., 2013. Learning to rank for recommender systems. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, pp. 493–494.

Konstas, I., Stathopoulos, V., & Jose, J. M., 2009. On social networks and collaborative recommendation. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 195–202.

Koren, Y. & Sill, J., 2011. Ordrec: an ordinal model for predicting personalized item rating distributions. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, pp. 117–124.

Lee, D. H. & Brusilovsky, P., 2010. Social networks and interest similarity: the case of citeulike. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*. ACM, pp. 151–156.

Maneeroj, S. & Takasu, A., 2009. Hybrid recommender system using latent features. In *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on*. IEEE, pp. 661–666.

Pazzani, M. J. & Billsus, D., 2007. Content-based recommendation systems. In *The adaptive web*. Springer, pp. 325–341.

Ren, L., He, L., Gu, J., Xia, W., & Wu, F., 2008. A hybrid recommender approach based on widrow-hoff learning. In *Future Generation Communication and Networking, 2008. FGCN'08. Second International Conference on*, vol. 1. IEEE, pp. 40–45.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J., 1994. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. ACM, pp. 175–186.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J., 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. ACM, pp. 285–295.

Sarwar, B. M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., & Riedl, J., 1998. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work*. ACM, pp. 345–354.

Shardanand, U., 1994. *Social information filtering for music recommendation*. Ph.D. thesis, Massachusetts Institute of Technology.

Tunga, B., 2014. Logaritmic high dimensional model representation in image processing. In *AIP Conference Proceedings 1637 (ICNPAA 2014 World Congress: 10th International Conference on Mathematical Problems in Engineering, Aerospace and Sciences)*. pp. 1120–1126.

Tunga, M. A. & Demiralp, M., 2009. Computational complexity investigations for high-dimensional model representation algorithms used in multivariate interpolation problems. In *Advances in Numerical Methods*. Springer, pp. 15–29.

Van Meteren, R. & Van Someren, M., 2000. Using content-based filtering for recommendation. In *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*. pp. 47–56.

CURRICULUM VITAE

Name Surname : Özge Yücel Kasap

Date and Place of Birth : 19.01.1988 - Altındağ

M.S.: Bahçeşehir University, Software Engineering

B.S. : Bahçeşehir University, Software Engineering

Ph.D. : Bahçeşehir University, Computer Engineering

Publications :

- Özge Yücel KASAP, M.Alper TUNGA, A Polynomial Modeling Based Algorithm in Top-N Recommendation, Expert Systems With Applications (2017)

Work Experience :

- Cybersoft, Ongoing since Dec. 2015
- Bahçeşehir University, 2011-2015
- Kafein, 2010-2011