1-1-2012

A framework for personalized dynamic cross-selling in e-commerce retailing

Arun K. Timalsina
Wayne State University,

A FRAMEWORK FOR PERSONALIZED DYNAMIC CROSS-SELLING IN E-COMMERCE RETAILING

by

ARUN KUMAR TIMALSINA

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2012

MAJOR : INDUSTRIAL ENGINEERING

Approved by :

____

Advisor Date

____

____

____

DEDICATION

This dissertation is dedicated to

my parents, Shree Bodhnath Timalsina & Smt. Sabitri Timalsina

and

everyone who cared me and influenced my life in a major way.

ACKNOWLEDGEMENTS

I would like to express my profound gratitude and indebtedness to my advisor Dr. Ratna Babu Chinnam for his outstanding mentoring during my entire Ph.D. study at Wayne State University. Without his invaluable guidance and continuous support, this research would not have been possible. I am also very much thankful to my committee members Dr. Darin Ellis and Dr. Alper Murat for their continuous help and constructive comments to make this work a success. I am equally indebted to another committee member, Dr. Chandan Reddy, for his invaluable advice and input throughout this research study.

I am highly grateful to Dr. Kenneth Chelst and Dr. Leslie Monplaisir, Department Chairs, for taking care of my funding through assistantship throughout my Ph.D. program. ISE department officials, Rita Coyne and Margaret Easley deserve special appreciation for their excellent support.

I would like to express my special thanks to my dear friend Dr. Bimal Nepal for his exceptional encouragement and arrangement in initiation of my Ph.D. study at WSU. I am also thankful to Dr. Shashidhar R. Joshi, Nripa D. Khadka, and Dr. Sanjeeb P. Panday for their relentless support for me from Institute of Engineering, Pulchowk Campus, Tribhuvan University. I would also like to thank Shreeram N. Regmi for keeping me always motivated for work. My friends both in the ISE department and in the Nepalese community have played a big role to make my stay at Wayne State enjoyable. I offer my sincere gratitude to all of my friends and their families for their help in the successful completion of my study.

Last but not least, I would like to thank my family, particularly my wife Pragya, daughter Apala, my parents and other family members back at home, for their sacrifice and perseverance throughout this study. Thank you for your love and encouragement throughout this ordeal, which has been hard on all of us.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANOVA	: Analysis of Variance
AUC	: Area Under ROC Curve
CBF	: Content-based Filtering
CF	: Collaborative Filtering
COW	: Customer Oriented Weight
CRM	: Customer Relationship Management
FW	: Full Weight
LF	: Loyalty Frequency
LM	: Loyalty Monetary
LR	: Loyalty Recency
LSA	: Latent Semantic Analysis
MDP	: Markov Decision Process
MF	: Matrix Factorization
NB	: Non-Buyer
NNMF	: Non Negative Matrix Factorization
OCCF	: One Class Collaborative Filtering
PB	: Potential Buyer
PCA	: Principal Component Analysis
PDBP	: Personalized Dynamic Bundle Pricing
PLC	: Product Life Cycle
POW	: Product Oriented Weight
PPH	: Price Product Hierarchy
RFM	: Recency Frequency Monetary
ROC	: Receiver Operating Characteristic
SKU	: Stock Keeping Unit
SVD	: Singular Value Decomposition
TCOW	: Temporal Customer Oriented Weight
TPOW	: Temporal Product Oriented Weight
UW	: Uniform Weight
ZW	: Zero Weight

CHAPTER 1

INTRODUCTION

1.1 Background

These days, it is typical to see customers purchasing particular items being recommended with additional items, often at a discount. This is a common practice in many business sectors. Examples include travel packages (e.g., hotel reservations with airline ticket), insurance services (e.g., home insurance with auto insurance), apparel (e.g., tie with shirt), restaurants (e.g., soda with sandwich), and consumer goods (e.g., memory card with digital camera). Cross-selling has emerged as an essential means for realizing higher sales without incurring additional business investment. Yadav and Monroe (1993) find that the additional savings offered under such bundle packages have a greater relative impact on buyers' perceptions of transaction value than savings offered on individual items. Various synonyms of the similar concept are bundling, cross-selling, and upselling, which have been considered as interesting research issues in economics, marketing, and operations management (Adams and Yellen, 1976; Venkatesh and Mahajan, 1993; Salinger, 1995; Stigler, 1963). The central theme of any related study is either to identify the optimal packaging complement or to derive the formulation for fixing the optimal discounted price of the packaged bundle, or both. Depending on the time of the decision for the optimal package components and the optimal price, research can be categorized into two types: dynamic and static. Under static cross-selling (or bundling), the discounted price and the components of the bundles or packages are already fixed before customer arrival (Schmalensee, 1984; Hanson and Martin,1990; Harlamet al., 1995; McCardle, Rajaram, and Tang, 2007). Whereas, under dynamic cross-selling, both actions are optimized only after the customer initiates the purchasing process (Aydin and Ziya, 2008; Netessine, Savin, and Xiao, 2006; Elmaghraby and Keskinocak, 2003; Gallego and Ryzin, 1994), which makes the problem very challenging and complex.

Traditional brick-and-mortar shopping malls offer only static and pre-packaged bundles because of the high implementation cost and incurred time delay for making dynamic packaging and arriving at optimized pricing decisions (Elmaghraby and Keskinocak, 2003). In order to enable dynamic cross-selling, both of the decisions, associate item finding for bundling and discount calculation, have to be started only after a particular customer initiates the purchasing process for the first item and have to be completed before the payment (or check-out) process. Hence, we need a suitable environment to carry out this in a timely fashion. The whole process of understanding customer interest and then building personalized bundles necessitates sales environments beyond traditional brick-and-mortar retail store. The e-commerce-based online shopping set-up offers a much better and implementable environment for the application of dynamic packaging as well as dynamic pricing.

1.2 Research Motivation

E-commerce (including m-commerce) is fundamentally changing the overall economy and business practices. While one out of ten people were using mobile-cellular devices and one out of twenty were using the Internet in the year 2000, in

2011, almost nine out of ten people used mobile devices, and every third person used the Internet (source: ITU)[†]. These enabling technologies are having a tremendous impact and reshaping every business from their traditional set-up of operation.

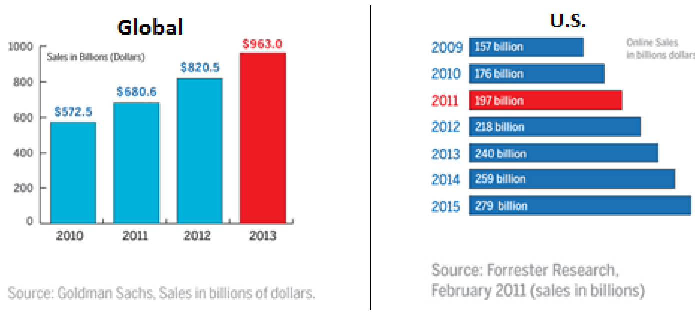

Figure 1.1: Global and U.S. e-retail sales growth forecast

According to a National Retail Foundation report, the U.S. Department of Commerce estimates almost two-thirds of the U.S. GDP comes from retail consumption[‡]. As shown in Figure1.1, Goldman Sachs estimates that global e-retailer sales is growing at the rate of 19.4% per year and will reach almost a trillion dollars by 2013. Similarly, Forrester forecasts that U.S. e-retailer sales will grow 10% every year with a volume of 279 billion dollars by 2015[§]. All these trends affirm the huge opportunity for retailers and e-retailers.

With shrinking profit margins and competition not only from neighboring retailers but from overseas e-retailers as well, e-retailers always need to look for various ways to boost their sales without incurring extra costs. Dynamic cross-selling is one of the most promising tools that can cater to the needs of competitive e-retailers. The two tasks of finding an appropriate second item as an

[†] The World in 2011: ICT Facts and Figures ; http://www.itu.int/ITU-D/ict/
[‡] (http://www.nrf.com/modules.php?name=Pages&sp_id=1215)
[§] (http://www.internetretailer.com/trends/sales/)

extra cross-sell item and offering an optimal discount amount on that dynamically formed bundle package are complex and time consuming. The first task of finding a complementary item for a particular customer is a personalized recommendation problem (Su and Khoshgoftaar, 2009; Sarwar et al., 2000; Kim et al., 2002; Linden, Smith, and York, 2003; Liu , Lie, and Lee, 2009), whereas the second task of fixing an optimized price for that unique bundle is a dynamic pricing problem (McCardle and Rajaram, 2007; Aydin and Zia, 2009). Instead of offering bundles with random components as an extra cross-selling item, personalized bundles with customer interest matching the associated item offer will improve the business in three ways by increasing the 1) conversion rate of simple browsing sessions to transaction sessions, 2) average basket size or the number of products purchased in a single session, and 3) customer retention level, avoiding the higher cost of a new entrant. Similarly, dynamic pricing also increases the revenue from the extra sales generated through adjusting the product price with discounts, which lowers the product valuation, matching more the customer's valuation of the product (Keeney, 1999). However, the limited experiment with a dynamic programming-based formulation (Netessine, Savin, and Xiao, 2006) currently used is not adequate to handle the complexity generated through user interactions even with thousands of SKUs from any moderate retail store.

There are very few studies (Netessine, Savin, and Xiao, 2006; Aydin and Ziya, 2009) in the area of personalized dynamic cross-selling, and those that do exist are myopic, drawn either from the aspect of data mining or marketing or inventory, which is not directly applicable for retail businesses because of the class differences in recommendation (Pan et al., 2008; Sindhwani et al., 2009) and dynamic pricing to be differentiated at the level of customer product pair. The basic assumption in other studies is that there is no computational limitation on identifying the right combination of products for bundling, which is far from the reality of any retail store operation. For example, a Wal-Mart store maintains 40,000 to 80,000 SKUs, which results in 800 million to 3.2 billion bundle packages with 2-SKUs. Online market leader *Amazon.com*[**] maintains 10 million SKUs, which results in 50 trillion bundle packages with 2-SKUs. Hence, the significance of any formulation, without the capacity to reduce the complexity before doing any further calculations, is questionable for real world applications.

The lack of current research in the literature and in industry practice is another issue regarding dynamic discounting policies for bundle packagingin cross-selling. *Amazon.com* attempts to address the issue by providing recommendations for additional items based on past transactions and other similar customer purchases; however, their price is still static without any offer of incentives for additional purchases. Similarly, another online retail company *Newegg.com*[††] provides a '*combo*' sale with prepackaged, cross-selling items with a price break. However, their prepackaged bundle discount can be considered as closer to the dynamic price practice but not the dynamic cross-selling feature.

[**] http://www.internetretailer.com/trends/e-retailers/ (access on Nov 2011)

[††] http://www.newegg.com/ (access on Nov 2011)

To the best of our knowledge, there is no earlier study on a real-time personalized discounting policy for dynamic cross-selling in the e-commerce environment. In this work, we develop an integral policy to design personalized complementary products to offer as a bundle package with optimized discount pricing as a dynamic cross-selling feature within the e-retailing environment.

1.3 Research Objectives

The objective of our study is to develop cross-selling models for the e-commerce-based market environment with dynamic interaction with online customers. The specific objectives are as follows:

1) Develop a model for recommending complementary SKUs to form candidate bundles for a customer based on the customer’s preference. Here, each customer is offered a list of unique personalized cross-selling packages formed with the consideration of the customer and their collaborative transaction history.
2) Develop personalized dynamic price discounting policies to promote cross-selling packages.

The first objective of recommending a complementary item is realized with the matrix factorization method. An individual customer preferred product list is estimated through implicit behavior extraction from historical transactions using the proposed temporal weight incorporated One Class Collaborative Filtering (OCCF). Similarly, the second objective of developing a discounting policy is realized with another proposed methodology, which uses the customer and product, price hierarchical pair information from historical transactions to set the optimal discount to be passed onto the customer.

1.4 Organization of Thesis

The remainder of this thesis is organized as follows. Chapter 2 details the simulation platform developed to generate e-retailer transactions. In chapter 3, we discuss the product recommendation problem and present various weight-based collaborative filtering (CF) methods. A model is proposed with the processes of temporal information extraction from transactional records, which are primarily based on PLC and CRM attributes, followed by integration into the one-class problems of such CF model in the form of weights. Results comparing the proposed model with other baseline methods are also presented. Similarly, in chapter 4, the process of personalized discount setting methodology is first discussed. Later, the proposed method with evaluation results is compared with other loyalty-based methods. Chapter 5 concludes the thesis by summarizing the contributions of the research and with a discussion of possible future research endeavors.

CHAPTER 2

SIMULATION PLATFORM FOR GENERATION OF E-RETAILER TRANSACTIONS

2.1 Introduction

Online retailers, also known as e-retailers, are almost omnipresent, available at any time, any location, and in no time to an online customer, whereas a brick-and-mortar retailer is only at a physical location, operating within fixed times with various queue delays. In addition to these benefits, e-retailers can even provide differentiated and personalized service and products to each customer based on their profiles. Profiles can be in explicit records, such as gender, age, address, etc. as entered by the customer during registration or billing. Implicit profiles of past behaviors are captured through historical transactions, which is more challenging to work out but promising in terms of system performance. This type of datamining of customer profiles offers various opportunities for revenue management for the retail industry, such as up-selling, cross-selling, dynamic pricing, etc. However, the retail data available for research purpose is always a big question. We observe that real transactional data non-availability is very common, and in this highly competitive market, it is rational to be reluctant to share business model secrets through the easy availability of transaction data. The problem of finding the balance between the level of confidentiality maintained from disclosed data and the legitimate needs of the data users is still a research problem to be further explored as pointed out by Dasseni (Dasseni, et al., 2001). If one has to pre-process the real data and perform data masking or modification with to preserve confidentiality, it may have an impact on various latent associations within the data. The resulting quality of such data may become questionable. To overcome these problems, this paper illustrates a novel framework for the generation of a realistic synthetic online transactional record set.

Agrawal and Srikant (1994) proposed an efficient algorithm for the market-basket-association problem, which made dramatic reduction in the search space and they gained a very high level of acceptance for their work. They used their own synthetic data generator (IBM Quest) for algorithm verification. The IBM synthetic data generator basically generates a list of items or products as a single transaction with a varying number of items. Only item list information is enough for market-basket-association at SKU level analysis; however, to extend the analysis to the product category, sub-category level, or to the personalized customer level, additional information is needed. Our attempt is to provide a synthetic data generator framework for the online shopping-based e-retailer which meets all of these requirements. Our framework even provides the product life cycle features for temporal analysis and market experimentation.

2.2 Model Environment and Specifications

Any e-commerce transaction-based simulator should capture and address a range of information related to the product, customer and transactions. The primary objective of the simulation is to mimic the market dynamics and customer behavior through retail transaction activities. Product-related information is already available to e-retailers, but the problem of extraction (or estimation) of implicit customer information, such as individual customer reservation on price and customer preference (or interest) for a particular product or product group, becomes very interesting and challenging. Unlike traditional retailers, e-retailers benefit from ubiquity, but the transaction activities transform into heavy-tailed distribution as argued by Anderson (2006). Therefore, working only at the level of a single SKU and individual customer, even in a moderate-sized retail environment, the data sparsity problem dominates and masks the huge amount of latent but meaningful information related to the customer and product. Similar processing but at a different hierarchy, such as product categories, customer types, and price range levels, will be able to reveal such meaningful and associative information in aggregate form. Thus, the proposed framework maintains products and customers with their corresponding hierarchical category or class level information. In addition, various temporal aspects of transactional sequences are also retained for later analysis. Product life cycle activity is also modeled through beta-distribution with product launch and product launch-off information and processes. Every product launch price is preserved for later price- related experimentation.

2.2.1 Products and Product Categories

For illustrative purposes, we have chosen the computer/laptop market segment within consumer electronics e-retailing sector for this study. The basic products and product categories have been selected after visiting various e-retailer vendors such as *Amazon.com*, N*ewegg.com*, *Microcenter.com*, and *CircuitCity.com*, as depicted in Figure 2.1 and detailed in Table 2.1. Three categories of product samples have been selected for the proposed simulator as also given in Table 2.1.These settings are only for illustration purposes; the model is flexible enough for any number of product categories.

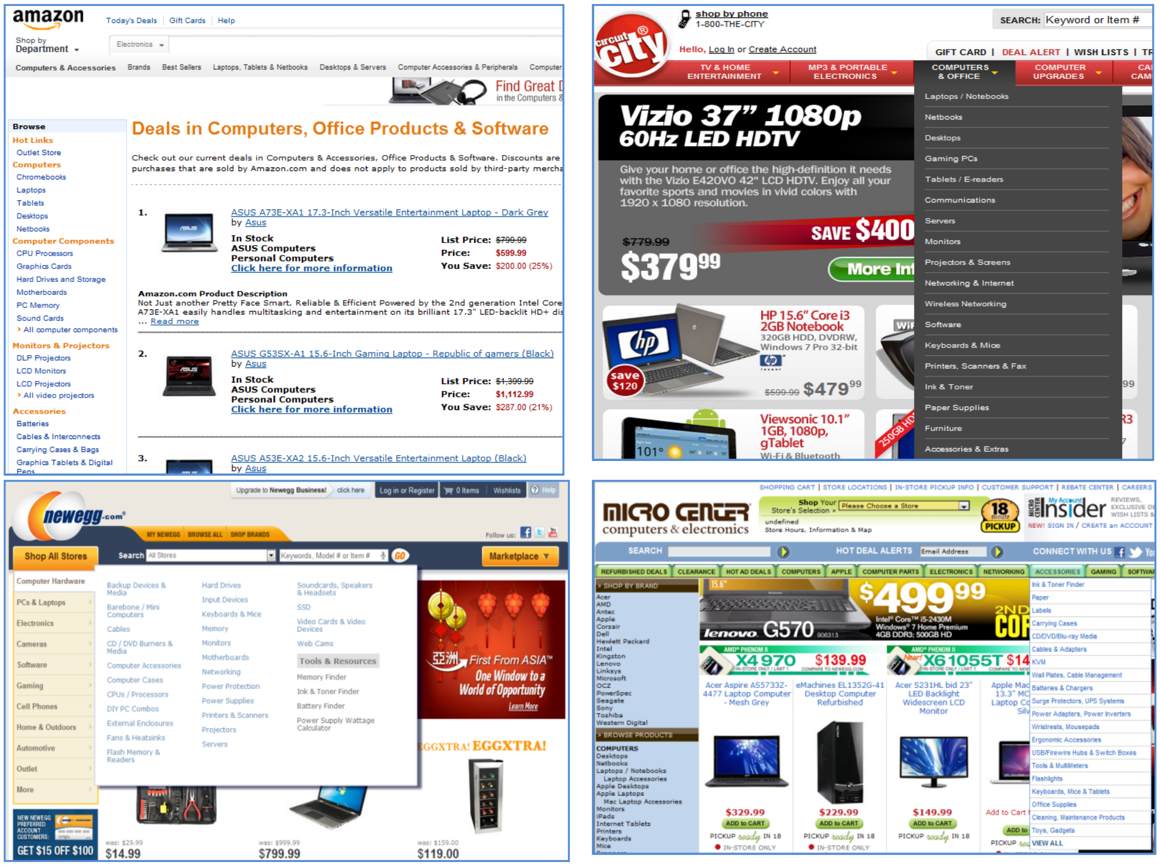

Figure 2.1: e-Retailer websites: *Amazon*, *Newegg*, *Microcenter* and *CircuitCity*

There exists a certain level of association among the products and product purchase probabilities. A regular customer might make multiple purchases with computer category products as principle items with follow-up purchases of computer parts and accessories. In addition to the previous general example of regular customer purchases, extreme cases of only a purchase of either accessories category items or computer parts item is also possible either for any occasional (including first-time purchase) customers or any particular bargain hunting customers.

Table 2.1: Sample product type categorization

Category	Products
Computer / Laptop (**CL**)	Desktops,Laptops / Notebooks, Netbooks, Apple Desktops / Notebooks, Tablet PCs
Computer Peripheral (**CP**)	Monitors, Printers,Processors, Motherboards, Hard Drives, System Memory / RAM, Graphics Cards, Power Supplies, Computer Cases, Optical Drives, Air & Liquid Cooling, Sound Cards,Controller Cards, Case Accessories
Computer Accessories (**CA**)	Printer Ink & Toner, CD/DVD/Blu-ray Media, Cables & Adapters, Batteries, Chargers, UPS, Surge Protection, KVM, Wrist rests, Mouse pads, Speakers, Microphones, Carrying Cases, Office Supplies, Cleaning,Paper, Labels

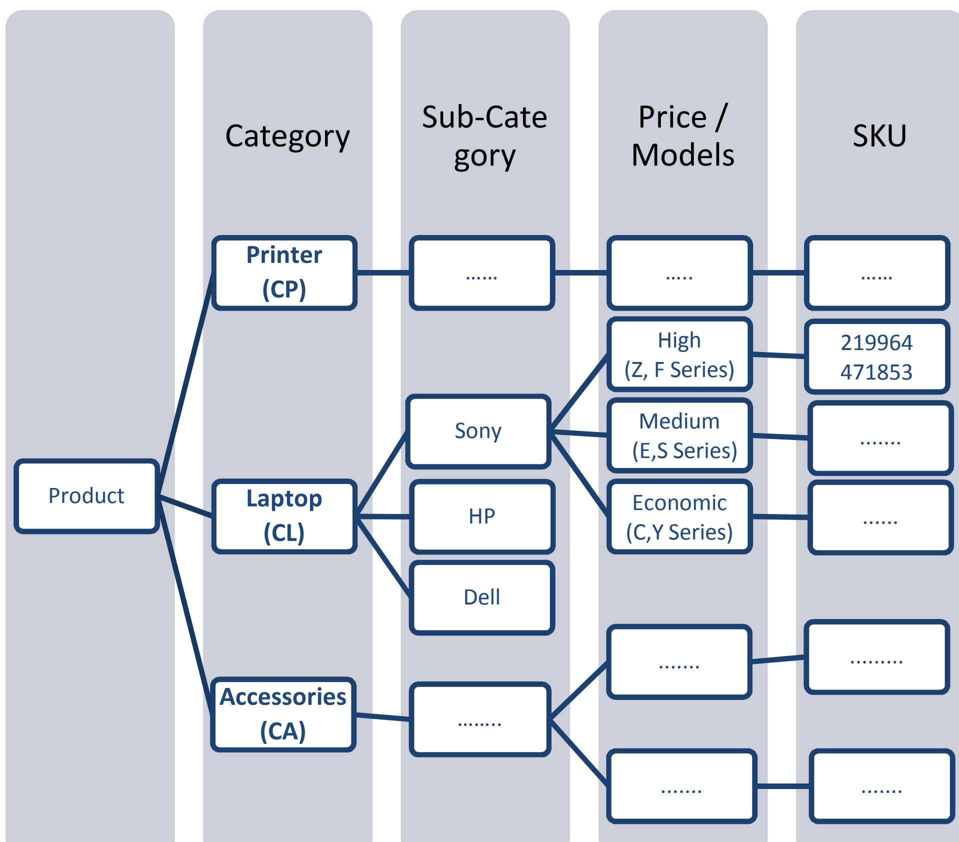

Figure 2.2: Products with their hierarchical information illustration

The basic product categories are further classified into various sub-categories of similar product types. Even within a sub-category, there are different models, and different models have different levels of performance, quality, and of course, prices. In our simulator model, we encoded such price level differences also, which is directly related to customer type and their preference towards the product types. Such hierarchical categorization is similar to the illustration in Figure 2.2.

2.2.2 Customer and Customer Segments

Another important aspect of customer behavior is mainly driven by customer sensitivity towards price. In the pricing literature, cost-based and competition-based pricing strategies are the two most prevalent strategies for a traditional retail setup. Customer value-based pricing is another promising strategy, found to be superior as a result of experimental studies (Ingenbleek et al., 2003). Recency, frequency, and monetary value (RFM) have remained the primary parameters for customer segmentation in the literature for many years (Kohavi and Parekh, 2004). Based on the prevalent existence of differing customer need, the customer segment is divided into three types (Hinterhuber, 2008) as follows:

- price-driven segment of customers (**PS**) (aka bargain hunters, late adopters, end-of-season-shoppers)
- mainstream segment of customers(**MS**) (aka regular, general customers)
- sophisticated segment of customers(**SS**) (aka early adopters, brand-loyal customers)

Sensitivity towards the price of a product can be assigned differently to these different types of customers on the basis of their customer type. The brand-driven or early-adopters are nearly insensitive to price, whereas bargain hunters and late-adopters are highly sensitive to price.

2.2.3 PLC and Temporal Features

As indicated previously regarding the general case of regular customer purchase behavior, a customer needs computer peripherals and accessories only when she has already purchased a computer or laptop. However, a customer can purchase a computer from one vendor and parts and accessories from other vendor, but we consider there to be a very small number of such customers within our system. Therefore, in a huge customer transaction dataset, most of the accessories and computer-parts sales are follow-up sales of primary computer/laptop sales for the same customer.

Every product has unique product life cycle-based sales characteristics following introductory phase, with phases of minor sales, major sales, and then again reduced sales caused by the market saturation effect. In order to have a proper analysis of transactions, a fixed number of periods (time-window) based transaction cases are considered so that the new product will be launched periodically and old products are launched off from the system (removed from the shelf). The length of the product life cycle varies across different products based on their attributes. A particular laptop model may be removed earlier than another new technology-based monitor model due to its sales pattern.

Each product has own its product life characteristics, which may be similar to product family characteristics or even unique one among all of the SKUs. The new product models with additional features built into advanced technologies will be offered with a high price tag into the current market, but previous models have to be offered with discounts to clear the stock. This characteristic of the hi-tech market (consumer electronics) is not much different than that of a retail market where the product is seasonal (like summer versus winter products) and time sensitive item markets (such as food and pharmaceutical products that also have expiration dates). Thus, depending on how long that product is in the market, offering varying discounts on the corresponding products will have an impact on sales and the overall improvement of the total revenue of the vendor.

2.3 Model Notations

Selling Periods

τ_k	a particular period (selling season) :$\in [1, K]$
Δ	average number of customer arrival per period, which follows Poisson distribution
Δ_k	number of customer arrivals during period *k*

Products

c	product category:$\in [1, C]$
b	product sub-category or brand of a product :$\in [1, B]$
r	price ranges of products within the level of product sub-category:$\in [1, R]$
ψ_C	product category proportion vector :$\in (0,1)$
$P_j^{c,b,r}$	SKU or a product with *j* as product–id and falling in the class of *c* category, *b* sub-category, and *r* price-range

t_j^l	product launch period of product *j* with the value *l* :∈ $[1, k]$
t_j^{lo}	product launch-off period of product *j* with the value *lo* :∈ $[1, k]$
δ_C	default saturation period counts for launch-off decision of products within the product category *c*: ∈ $[1, k]$
$\beta_j(al, be)$	a beta-distribution of a product life cycle with parameter *al* and *be* and selling distribution for product *j*
σ_j^l	is price-tag of the product *j* during the launch period *l*

Customers

s	customer segment :∈ $[1, S]$
$\mathcal{C}_i^s$	a customer with *i* as customer–id and falling in *s,* the customer segment
ω_s	customer segment proportion vector :∈ $(0,1)$
λ_s	customer arrival proportion based on customer segment, *s* :∈ $[0,1]$
π_i	reservation threshold (WTP) of individual customer*i* :∈ $(0,1]$
$\pi_s^{'}$	price-based average purchase threshold for *s,* the customer segment :∈ $(0,1]$
$\upsilon_{S,R}$	customer segment preference rate over product price ranges: ∈ $(0,1)$

Transaction Pattern

Ω_{m}	mth seed pattern set (based on all combinations of product categories)
Φ_{m}	size of mth seed pattern(number of products within a pattern, *m*)
$\Phi_{\max}$	maximum size of seed pattern(maximum number products in a pattern)

Assumptions :

- Infinite inventory available.
- At the time of transaction generation, customers are indifferent towards the price of the products.

- Products are launched on and off regularly, so transactions are generated only from available products list within a particular period, following corresponding seed pattern, which is formed of from the different combination of product categories.

2.4 Market Simulation

In order to mimic the real market scenario, we must consider a certain number of periods (or selling seasons) for transactions. The simulation starts with the generation of products. A new product launch into the market is random, but the product and all of the associated product attributes should be properly entered into the system before any transaction occurs. Each product with a unique productID falls into a particular class of product categories. Each product is further categorized based on brand specificity or the price range of the product, which might be something like high, medium, low, etc. Each product is launched at a particular period, and some of the products are launched at another period. Product launching is done at the start of the period so that the product transaction may happen during the same period. Similarly, the launching off process is also done at the start of the period, and there will be no transaction during and after that period. We consider that each product life cycle (sell-pattern) follows a typical beta-distribution with particular parametric α (*al*) and β (*be*) values.

Similarly, we also generate the member customers. We consider the customers to be pre-registered into the system before making any transactions, but we also allow for their arrival process as a completely random process. Each customer has a unique customer-ID and falls into one of these particular segments. We consider each customer has a reservation price threshold (π_i), which is randomly generated with consideration for bias from the purchase threshold (π_s') of the customer segment, i.e. any price below this threshold is considered for purchasing. These two latter parameters influence a customer in making the decision to purchase any product in that particular period with a particular price tag. We also consider the customer segment-based preferences ($\upsilon_{S,R}$) on different price ranges within certain product categories.

2.4.1 Customer Generation Process

We initialize the simulator with customers and products first, before generating any transaction dataset. As the customers are pre-registered to the system, customers are generated and assigned sequential integer numbers based on customer-IDs. Then whole customers are divided into a certain number of segments as specified by proportion parameter, ω_s. Each customer has a unique price reservation threshold, also known as willingness to pay, π_i, which is a random value based on the segment-based reservation threshold (π_s').

Customers = $\mathbb{C} = \{\mathbb{C}_i^S, \omega_s, \lambda_s, \pi_i\}$

2.4.2 Product Generation Process

After the customer registration process, we generate the product list. As with customers, products also need to be registered into the system before any transactions can occur. For the sake of processing, we generate all products before transactions; however, the product launch is purely random during the market activities, which is not algorithmically different from the scenario of new products entering the market continuously. Each product is again assigned to one of the different product categories, which broadly classifies the generic product types (e.g. computer/laptop, computer peripheral, accessories, etc.). To distinguish among different brands, high-end products, and premium priced products, we again classify the products by assigning level-wise categorical values as sub-categories (e.g. high, medium, low, etc.).

Products $=\mathbb{P}=\{P_j^{c,b,r}, c, b, r, \psi_C, t_j^l, t_j^{lo}, \delta_C, \beta_j(al, be), \sigma_j^l\}$

Each product is also assigned with random launch period values. Similarly, default launch-off values for each product are calculated using a default launch-off period of product category.

Default Launch-off ➔ $t_j^{lo} = t_j^l + \delta_C$

Similarly, the price of each product is randomly set within the pre-specified different price ranges within a product sub-category during the launch period.

In order to model the product life cycle patterns of product sales, we considered modeling the demand pattern following the beta distribution. Thus, we assigned random values for alpha and beta parameters of beta distribution for every product. The cumulative probability distribution (ξ_j^n) up to the period *n* for a product *j* is calculated using the below formula.

The cumulative distribution (ξ_j^n) up to the period *n*

$$\xi_j^n = \widetilde{\xi_j^x} = F(x, \alpha, \beta) = I_x(\alpha, \beta) = \frac{B(x; \alpha, \beta)}{B(\alpha, \beta)}$$

where beta function, $B(\alpha, \beta)$, is evaluated as

$$B(\alpha, \beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}\, dt; for \alpha, \beta > 0$$

and the number of the period, *n,* should be converted into the proportional rate, *x*, to calculate the cumulative probability of the beta-distribution.

Similarly, $I_x(\alpha,\beta)$ is also called the *regularized incomplete beta function* and $B(x;\alpha,\beta)$ is also termed the *incomplete beta function* and evaluated accordingly. Any incomplete beta function is evaluated as

$$B(x;\alpha,\beta) = \int_0^x t^{\alpha-1}(1-t)^{\beta-1}\,dt;\, for\ \alpha,\beta > 0\ and\ x \leq 1$$

Once we have the cumulative probability, we can calculate the probability density of any product for each period as the difference between two consecutive periods as formulated below.

The probability density for period *n*, $p_j^n = \xi_j^n - \xi_j^{n-1}$

We can define the market saturation (t_j^m) as the product demand saturation in the market. The saturation may happen at any stage of PLC, i.e. introduction, growth, maturity, and decline. Market saturation condition at different stages can be modeled as listed below:

(a) introductory stage saturation: the difference between two consecutive period demands is less than a value (e.g. 0.0001), if the trend remains the same for a fixed number of multiple periods and the cumulative demands up to this period are much less (e.g. 0.0500)

(b) growth stage saturation: the difference between two consecutive period demands is less than a value (e.g. 0.0001), if the trend remains the same for a fixed number of multiple periods and the cumulative demands up to this period are less (e.g. 0.2000)

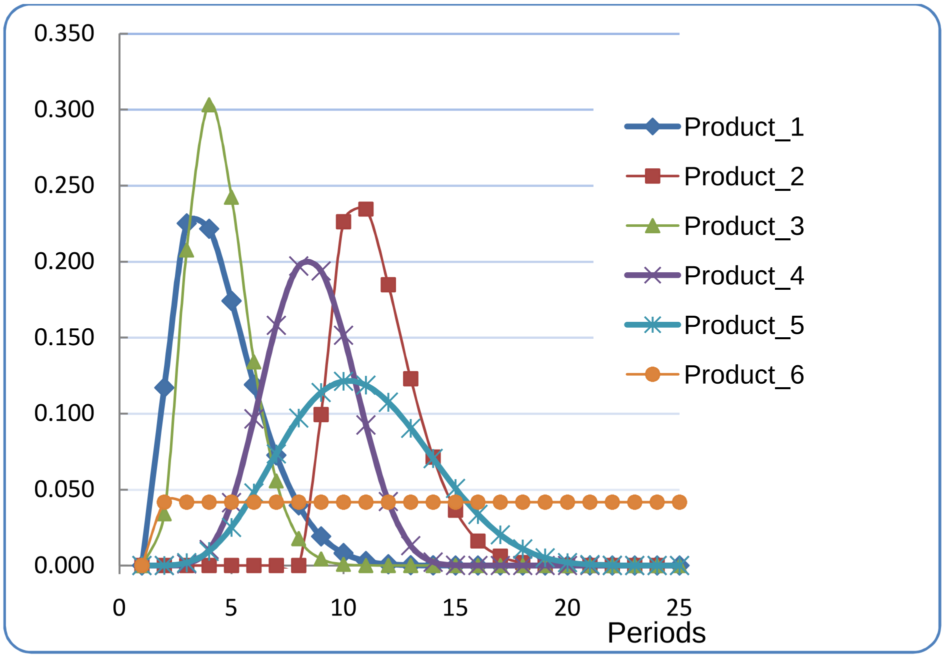

Figure 2.3: Sample PLC of different category products

(c) maturity stage saturation: the difference between two consecutive period demands is less than a value (e.g. 0.0001), if the trend remains the same for a fixed number of a few periods and cumulative demands up to the period are moderate (e.g. 0.5000)

(d) decline stage saturation: the difference between two consecutive period demands is less than a value (e.g.0.0001) with cumulative demands up to the period moderate (e.g. 0.9500)

For example, the market saturation (t_j^m) for decline stage formulation is as shown below:

$$t_j^m = \begin{cases} 1 & if\ \xi_j^{n-1} > 0.9500\ and\ \ p_j^n < 0.0001 \\ 0 & otherwise \end{cases}$$

$$t_j^m = 1 \Rightarrow t_j^{lo} = \tau_n$$

Samples of product life cycles based on their observed transactions are depicted in Figure 2.3. For comparison, product_3 lasted around 8 periods in the market with high sales rate before saturation, whereas product_6 was almost uniformly selling with a low sales rate for all 25 periods. Similarly, product_1's sales picked up from the very first week of the launch whereas product_5 sales were not even 5% after 5 periods though both of the products were launched together in the market.

2.4.3 Current Period Available Product List Generation

After the customer and product initialization processes, the actual transactions are generated. The transaction process starts with the setting of a current period or a particular selling season. Once the current period is set, then a current period available product list (at the SKU level) is generated from the master product list, $\mathbb{P}$, with the following conditions:

Current period: $\tau_k = n$

Then, the current period available products (at the SKU level) is the subset of the complete product list, where the product launch period of each product is less than or equal to the current period and the launch-off period is higher than the current period.

Thus, the product set is

$=\mathbb{P}' \subseteq \mathbb{P}\{\forall\mathbb{P} :\ t_j^l \leq \tau_k \ \ and \ \ t_j^{lo} > \ \tau_k\}$

with their corresponding demand densities

$p_j^n = \xi_j^n - \xi_j^{n-1}$

Now once the selected products are within $\mathbb{P}'$, each product demand density is calculated as mentioned previously. Finally, the normalized density of the products within the same sub-category and price range (one level higher in hierarchy) are calculated as below

$$normalized\ density\ = \Theta_j^n\ = \frac{p_j^n}{\sum_{b,r} p_j^n}$$

After this step, every product category will have a current period available product list with the corresponding cumulative demand density values of each product.

2.4.4 Master Seed Pattern Generation

Let us suppose our three product categories are L, P, and A. Then all combinations of these three categories will be {L, P, A, LP, LA, PA, LPA}, which is considered to be a transaction seed pattern, Ω set. The size of the first three patterns (Φ) is just 1; the size of the next three patterns is 2, and the last one is 3. From our historical transaction analysis of the ψ_C, product category proportion vector, it is trivial to get the proportion of such pattern probabilities. Considering such seed pattern probabilities and currently available products within a particular category and their normalized probabilities, we can generate transactions with different products and their combinations.

2.5 Transaction Generation

This is the main process step, in which we generate the transactions for the entire current period. The transaction involves only those products which have already been launched and not yet launched off. Product launching is done at the start of the period so that the product transaction may happen during the same period. Similarly, the launching off process is also done at the start of the period, and there will be no transaction during and after that period.

2.5.1 Customer Arrival Process

Following Δ_k, the number of customer arrivals during period *k* , the customers will be entered into the system one by one. A random customer entering the system will be assigned to a particular customer segment based on the λ_s customer arrival proportion, which in turn is based on the particular customer segment.

2.5.2 Product Selection Process

Once the customer with the segment is identified, we select the probable seed-pattern from the pattern set Ω and denote as Ω'.

1. Read the number of total seeds of Ω'.
2. Randomly pick one of the seed from Ω'.
3. Record the product count of the selected seed pattern as the number of products to be generated.
4. Randomly generate the price range of the product subcategory of the selected seed pattern according to $\upsilon_{S,R}$.
5. With this price range and product sub-category, match one among the available products of this particular product subcategory and price range.
6. Repeat the same procedure up to Φ times as per the selected pattern product count.

A flowchart of the transaction generation process is shown in Figure 2.4. After properly initializing all the required variables, the various processes are sequenced as shown in the following diagram. Each shaded process is advanced with a corresponding assignment based on a randomly generated specific number.

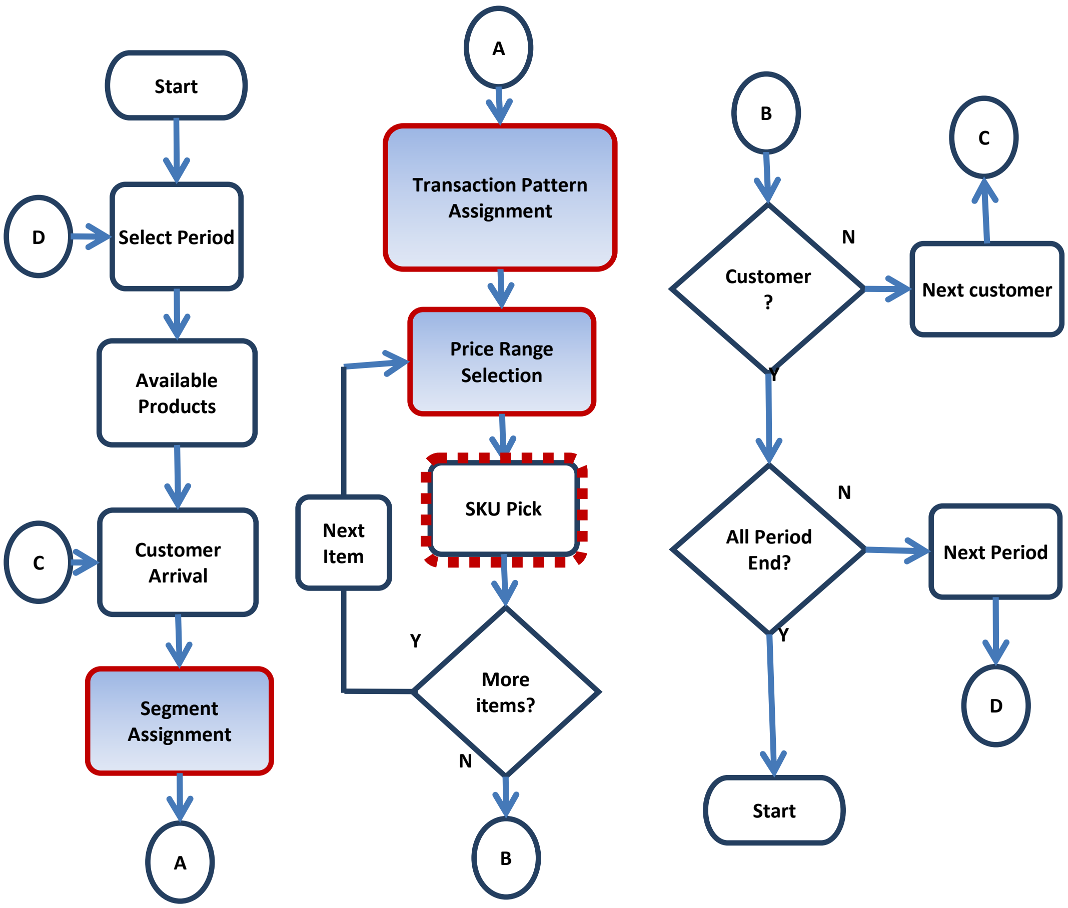

Figure 2.4: Flowchart of transaction generation processes

The most important process, “SKU Pick”, does the final product (SKU) assignment of the customer. During this process, a group of SKUs that belongs to same product sub-category, price range, and availability in the period, competes with each other. Any customer picking probability is cumulatively distributed based on their demand distribution estimated through beta-distribution.

2.6 Validation

Here we present some of the simulation run snapshots. We ran the simulator with 5,000 products and 5,000 customers for 24 periods with Poisson distributed 10,000 as the average number of transactions per period for transaction generation. In Figure 2.5, product category based transaction counts are shown. The number of total transactions per period is closer to 10,000, and as the product category SKUs are distributed as CL(25%), CP(35%), and CA(40%), randomly distributed, but a much closer number of transactions are observed. Initial few periods, during the simulator warm-up periods, transient response is observed. After the warm-up periods, steady state response is observed.

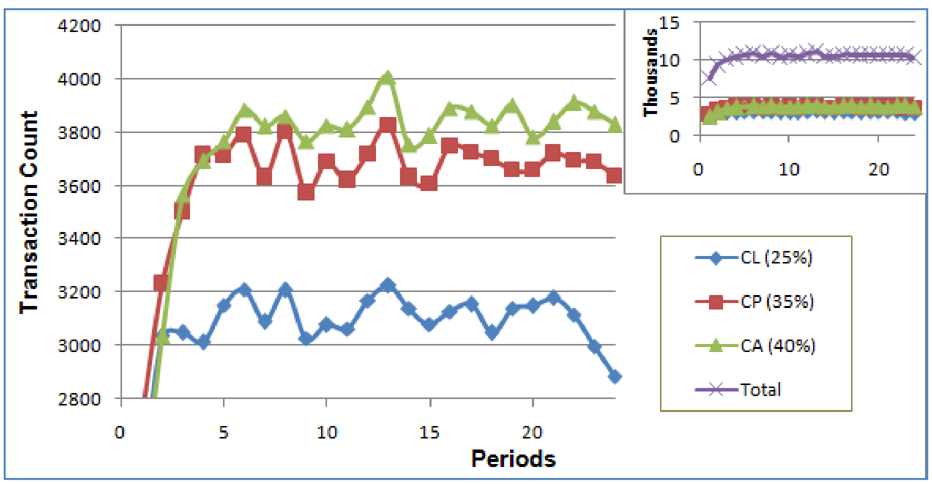

Figure 2.5: Transaction counts of different product categories

Similarly, in Figure 2.6, we show the number of transactions per period for different customer segments. Different customer segment-based arrival rates control the transactions per customer segment types. Even though it is random, it still follows overall very closely to the rates allocated for each customer segment.

Table 2.2: Customer segment-based price-range preference and arrival rate

Price-Range → Preference ↓		Price Ranges			Customer Arrival Rate
		High	Mid-Range	Economic	
Customer Segments	SS	0.6	0.3	0.1	0.15
	MS	0.1	0.6	0.3	0.45
	PS	0.1	0.3	0.6	0.40

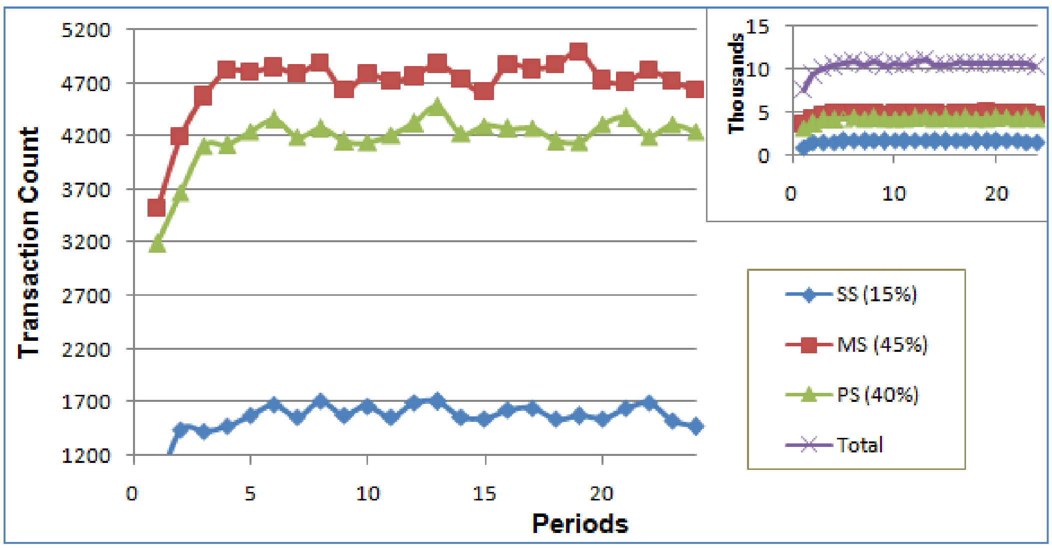

Figure 2.6: Transactions from different customer segments

In Figure 2.7, we show the transaction counts which are grouped based on the product price ranges, which are classified as three price ranges, i.e., high, mid-range, and economic. The average distribution of the transaction count is actually derived through the illustrative example case values as listed in Table 2.2.

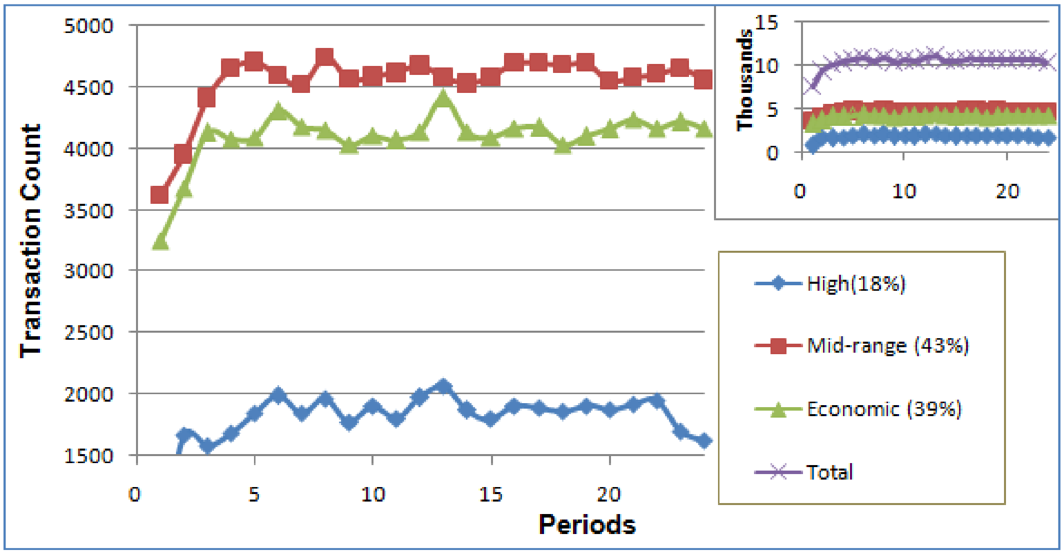

Figure 2.7: Transactions with different price range preferences

2.7 Conclusion

In this chapter, we have presented our simulation framework model for transaction record generation. Utilizing the higher level hierarchical information of products and customers with broader level association information, we presented an efficient framework for synthetic data generation for e-retailers. The framework generates multi-periodic transactions consisting of the number of products and customers that fall within their categorical classes. The framework is even suitable for product life cycle analysis as it maintains the product launch, launch price, margin and other related information.

CHAPTER 3

TEMPORAL INFORMATION INTEGRATION IN OCCF FOR PRODUCT RECOMMENDATIONS

3.1 Introduction

The primary challenges of any retail industry are estimating customer interest/perception/preference over range of products and estimating the exact valuation of a product from the market perspective. At the level of the transaction event, these challenges translate into the single task of finding a perfect match between an individual customer and a single product. These days, traditional brick-and-mortar retail businesses are transforming to an e-retailer (e-commerce or m-commerce-based retailer or online retailer) mode at a faster rate than ever. As per the 2008 U.S.Census Bureau report[‡‡], whole U.S. retail sales increased from $2.58 trillion in 1998 to $3.95 trillion in 2008 with a CAGR (cumulative average growth rate) of 4.3%. During the same decade, online trading increased from 0.2% ($4.98 billion) to 3.6 % ($141.89 billion) of total sales with a CAGR of 39.7%. Interestingly, total retail sales from 2007 to 2008 decreased 1.1% because of unfavorable economic conditions; however, in the same period online sales increased 3.3%.

Despite such a huge opportunity with e-commerce, e-retailers have to deal with a higher level of complexity due to the increase in the number of customers and products as a result of the physical limitations of brick-and-mortar based

[‡‡] http://www.census.gov/retail/, accessed on May 2011

retailers. When there is an enormous range of options available for selection, the traditional recommender system generates a targeted list of potential choices. It can be either a list of items for a target user or a list of users for a target item. A personalized recommendation for a typical user is generated through associative exploitation of the explicitly expressed interests or extracted from the implicit behaviors of similar users; this method is known as collaborative filtering (CF). One of the most studied problems of this research domain is the Netflix movie recommendation task, where users express their movie interests in the form of a wide range of rating values.

Given the available dataset of transaction records of purchased items, news item recommendation records with recommended sites, or social bookmarking lists with bookmarked tags, the problem turns into a one-class collaborative filtering (OCCF) type. This OCCF is a relatively less studied problem because of the poor performance of the CF-based traditional recommendation system on one-class problems. In addition to the universal CF problem of unbalanced or sparsely labeled datasets representing positive interest, the efficiency of OCCF-based methods depends on the treatment and consideration of the unlabeled or missing dataset but also has these highly confounding datasets that have both negative interest and soon-to-be positive interest. There are some recent experiments on such OCCF problems applying weight-based non-negative matrix factorization techniques. Their results are based on frequency-based information like customer count, product count, product popularity, etc.

In today's aggressive market, product life is becoming short and product portfolios are constantly changing. Similarly, purchasing preference for a customer might be event-dependent and vary over time. Also, product perception is heavily influenced by evolving selections, which are available in the market at a particular time.

The major contribution of this work is developing a methodology to incorporate the different sets of temporal information to improve the quality of recommendation for OCCF domain problems in the e-retailer business. In particular, both product life cycle (PLC)-related product launch information and customer relationship management (CRM)-related customer recency information are used as captured temporal information. After an empirical evaluation of several simulations of synthetic e-commerce datasets generated through the model framework, as explained in chapter 2, we compared the proposed method with other industry standard techniques. The promising results confirm the efficacy of temporal information on OCCF.

The remainder of the chapter is organized as follows: In section 3.2, the literature review is laid out as a related research discussion. In section 3.3, our proposed methodology with formulation is presented in detail. The various methods of evaluation are compared by using the same e-retailer transaction records, the results of which are presented in section 3.4. Section 3.5 concludes the work with a discussion of future research.

3.2 Literature Review

Before making any decision, it is natural to seek corroboration with other sources of information, ranging from other people's opinions, references, recommendations and comments on any related news through different media sources. The goal of such verification is to achieve higher confidence in moving from solitary to mass knowledge. In today's inundated market, customers are presented with a myriad of options for products and services. Similarly, for a vendor in the e-retailer mode of operation, there is no limiting factor in reaching any global customer. However, the growth of a vendor's business is proportional to the level of personalization that they can offer to an individual customer. A recommender system is the answer for such a huge task of information filtering.

3.2.1 Collaborative Filtering

Collaborative filtering (CF) is one of the most widely used techniques in designing recommender systems. CF exploits the associative interest that emerges from the known interests of other similar users. CF provides recommendations for previously unknown user-product pairings based on the associative interest of known user-product pairings. Content-based filtering (CBF) is another class of recommender systems, where the content information (such as customer profile, product options etc.) is utilized for recommendation, unlike in CF, where ratings or other numerical values are used. Goldberg et al. (1992) defined the term "collaborative filtering" for the first time, while making one of the earlier recommender systems, *Tapestry*, which was different from existing basic content-based filtering (CBF) and rule-based recommender systems. In

Tapestry, annotations contributed from early readers were collaborated and used for filtering the streaming documents within the newsgroup members. *GroupLens* (Resnick et al., 1994) was another system developed for the news-item filtering task similar to*Tapestry*. However, *GroupLens* was the first system that introduced the rating scores as a measure of user interest towards news items. Using the same approach of ratings-based scores, the same *GroupLens* group later started the *MovieLens* project and advanced from the news item-filtering problem to the movie recommendation problem. As a score rating-based approach, the basic assumption of CF is that if two users X and Y have rated *Z* number of items (historical basis) with very close scores; then they rate the remaining items (future prediction) with similar scores. A comprehensive list of several recommender systems built with CF or CBF are compared in Montaner et al. (2003). Similarly, Sindhwani et al. (2009) provides an overall detail survey of various recommender systems.

The two broad categories of CF systems (excluding CBF and hybrids) are based on the different processing techniques that are either memory-based, which are also termed as neighborhood methods, or model-based. *GroupLens* was the first to use one of the popular memory-based techniques. This technique uses the Pearson correlation-based neighborhood measure in its automated CF system (Herlocker et al., 1999). Another memory-based method is the item-to-item-based top-N recommendation technique, which has a wider acceptance among e-retailers including the market leader, Amazon (Linden, Smith, and York, 2003). Product recommendation from Amazon, as shown on Figure 3.1, is generated with the item-to-item recommendation technique, which follows the simple rationale that people who buy X also buy Y.

Figure 3.1: Snapshot of product recommendation for a customer from Amazon

There are a few improvement techniques for memory-based methods, such as *Inverse User Frequency*, *Case Amplification*, *Imputation-Boosting*, *Weighted Majority Prediction*, *Default Voting,* etc.(Su and Khoshgoftaar, 2009; Breese, Heckerman, and Kadie, 1998). These memory-based CF methods are popular because of the improved recommendation with faster and off-line calculation of correlation and other similar measures. However, with an increased level of sparsity, the performance of memory-based CF methods deteriorates because of the over dependence on common items among users for similarity measure calculation (Su and Khoshgoftaar, 2009; Adomavicius and Tuzhilin, 2005). Lack of emergence of general insight because of not having any learning component and resulting suboptimal accuracy are some other limitations of the memory-based methods (Hofmann, 2004).

Various model-based CF methods have been proposed and found to overcome the many limitations of memory-based CF methods. All model-based CF methods, at the first stage, learn to recognize the complex patterns that are present within the user and item, and their explicit and/or implicit preferences. In the second stage, after learning through historical data, the model-based CF methods provide recommendations, which are in fact the model-based predictions. Various predictive models, such as Bayesian models (Breese, Heckerman, and Kadie, 1998, Miyahara and Pazzani, 2000), dependency network based models (Heckerman et al., 2001), clustering models (Ungar and Foster, 1998), and MDP based models (Shani, Heckerman, and Brafman, 2005) are well documented in building model-based CF systems with promising performance. Recently, due to the Netflix movie recommendation prize competition[§§], there has been a surge of research on building efficient model-based recommendation systems. As a result, matrix factorization-based dimensionality reduction methods, such as singular value decomposition (SVD) (Sarwar et al., 2000; Sarwar et al., 2002), principal component analysis (PCA) (Goldberg et al., 1991), and probabilistic latent semantic analysis (pLSA)(Hofmann, 2004) are gaining popularity. In fact, that competition has already demonstrated that the latent factor-based matrix factorization models are superior to classic memory-based and other model-based techniques among CF

[§§]http://www.netflixprize.com/ , accessed on September2011 methods for recommendation tasks (Koren, Bell, and Volinsky, 2009). In choosing a particular method, generally, classification algorithm-based models are suitable to user preferences coded in categorical data type and regression and latent factor-based models are suitable to user preferences coded in numerical data type (Su and Khoshgoftaar, 2009).

3.2.2 One Class Collaborative Filtering (OCCF)

Most of the CF related research has used either *MovieLens* datasets, still maintained at *GroupLens*[***] or the *Netflix* competition[†††] datasets, which are no longer publicly available. In these datasets, users express their interest in the movies in the form of ratings with a wide range of scores, like 1-to-5, where 1 means they did not like it at all and 5 means they liked it the most; of course, 0 is set aside for unrated movies. In other words, the rate-based dataset has all three distinct categories of data: positive label (user's high rating on particular movies), negative label (user's low rating on particular movies), and unobserved or missing (no ratings yet). However, in many real world scenarios, users may have to either accept or discard as choices between two binary decision options. Such examples include purchasing an item from a retailer or an e-retailer, clicking on the linked webpage for more information, bookmarking a website for later reference, sharing a news-item on social media, etc. Though it seems all of the above problems are very similar to the recommender system point of view, these latter problems have only positive label data for model learning, whereas the former problems (rating based movie data) have both positive and negative label

[***] http://www.grouplens.org/ , accessed on September2011
[†††] http://www.netflixprize.com/ , accessed on September 2011

data for model learning. The CF problem with only positive examples has recently been termed as a one-class collaborative filtering (OCCF) problem (Pan et al., 2008). Such OCCF problems are less studied than existent rating score-based CF problems. OCCF problems have become harder due to the fact that in addition to the problem of few positive label datasets (the sparsity problem), the other two other categories of datasets, negative and missing, are confounded.

The Netflix award-winning algorithm-based model (Koren, Bell, and Volinsky, 2009) intuitively learns through latent factor-based matrix factorization to allocate the wide range of rating scores into different parts as global average, user bias, item bias, and user-item interaction. Unlike the rating scores dataset, there is no grading information in the OCCF dataset to partition the user and item biases. With the high level of sparsity of positive label data, the treatment of the remaining ones, which are confounded negative label and unobserved data, becomes critical in the OCCF problem. Pan et al. (2008, 2009) proposed different weight assignment schemes for learning through latent factor-based matrix factorization models for OCCF problems. A similar weight assignment technique was proposed as a weighted low rank approximation to improve the recommendation for unobserved data in rating the score-based CF problem (Srebro and Jakkola, 2003). In OCCF problems, the differentiated initial weight assignment of the unobserved dataset resulted in the significant improvement of the model performance (Pan et al., 2008) in comparison to the undifferentiated single weight assignment of the unobserved dataset as in (Srebro and Jakkola, 2003). The primary objective of such different weight assignments in OCCF is to provide a relative measure of unobserved data to be closer to the negative label or to the missing label. Sindhwani et al. (2009) further simplified weight assignment schemes with compact formulation and proposed the addition of another optimization variable for the OCCF problem.

3.2.3 Temporal Aspect Experimentation

Most of the models proposed for the well-known CF problem of rate-based movie recommendations consider user behavior as stationary, as in one who buys X also buys Y. There is very limited research on CF with temporal information despite the continuous updates on product popularity and regular changes in customer preferences. Koren (2009) proposed various temporal models for rate-based Netflix recommendations with reporting of improved performance. Similarly, some recent research considering the temporal aspect consideration is discussed in (Lu, Agarwal, and Dhillon, 2009; Xian and Yang, 2009; Xiong et al., 2010), and the dataset considered for all these studies are either *Movielens* or *Netflix*, which are standard CF problems but not OCCF problems.

Previous work on the OCCF domain only considers frequency-based information. Similarly, there are few studies with experimentation involving temporal information on to wide range of rate based collaborative filtering problem. To the best of our knowledge, there is no research reporting on experiments of a temporal information application on OCCF problems. We believe this paper is the first to integrate product launch and customer recency information, which are some of the temporal components of product life cycle (PLC) and customer relationship management (CRM) into a CF-based recommender system, which results in more robust and accurate recommendations for business processes like cross-selling, up-selling etc.

3.3 OCCF Model and Formulation

In this section, we first discuss the basic latent factor models that utilize matrix factorization. The OCCF problem-solving models that employ the differentiated weight assignment and low rank approximation are presented in detail. The proposed models are then formulated and explained.

3.3.1 Notation

A basic non-negative matrix factorization method was first presented by Paatero and Tapper (1994) and was documented as positive matrix factorization. Interestingly, they were working with huge environmental data and trying to explain the data with a few prominent factors list. Each factor is a positive combination of the basic variables. In other words, either the particular variable is present with a certain degree of positive effect or completely absent in the final result. In their model, there is no consideration of any negative effect of any variables, which is very practical in many application domains.

For the convenience of discussion, we will first introduce the annotation, but we will define a few matrices first.

X: actual transaction matrix with binary data, {1 : purchase; 0 : no purchase}

U: user feature matrix with latent features of customers (non-negative entries)

V: product feature matrix with latent features of products(non-negative entries)

Y: resultant matrix recommendation(non-negative entries)

$\mathcal{W}$: weight matrix (explained below): $\in [0,1]$

3.3.2 MF based OCCF

Suppose there are *m* customers and *n* products. Thus, in X, the actual transaction matrix or customer-product matrix, there will be *m* rows and *n* columns. The entry of 1 in $X$is to indicate the customer *i* out of *m* purchased a product *j* out of *n*. Similarly, the 0 entry is for the no purchase indication of that particular customer-product pair. This X is large but sparse and unbalanced: many zeroes and very few 1s.

Dimensionality reduction is the primary power of every matrix factorization method. Here the transaction events, which are the interactions between customers and products, are also mapped into the new joint latent factor space formed by latent customer features and latent product features. Let us consider U is a matrix representation of customer features. As, $U = [u_1 , u_2 , \dots\dots, u_m]^T$ is *m* × *r* matrix, the i^{th} row of U is a customer, and $u_i$who is represented in the *r*-dimensional customer feature space. Similarly, let us suppose V is a matrix representation of product features. As, $V = [v_1 , v_2 , \dots, v_n]$ is *r* × *n* matrix, the j^{th} column of V is a product, v_j that is represented in the *r*-dimensional product feature space. Here, this *r* is termed as rank of the factorization, which is the number of latent features to be analyzed. In general, the relations *m, n* >> *r* and *m* × *n* >> *(m + n)* × *r* verify the dimensionality reduction and processing efficiency through matrix factorization methods. The dot product ${u_i}^T v_j$ captures the closeness observed by user u_i towards the product v_j in the joint latent feature space. Let us suppose $Y = UV$. Now, the Y estimation task turns to a simple optimization problem as shown below:

$$\underset{U \geq 0, V \geq 0}{\arg\min} \sum \mathcal{L}(X, Y) \tag{1}$$

Here $\mathcal{L}$ is a squared error function or any other loss function as listed below:

$$\text{squared error} : \mathcal{L}(X, Y) = ||X - Y\,||^2 = \textstyle\sum_{i=1,j=1}^{m,n} (X_{i,j} - Y_{i,j})^2 \tag{1a}$$

$$\text{KL divergence loss} : \mathcal{L}(X, Y) = D(X||Y) = \textstyle\sum_{i=1,j=1}^{m,n} (X_{i,j} log \frac{X_{i,j}}{Y_{i,j}} - X_{i,j} + Y_{i,j}) \tag{1b}$$

And to overcome the over-fitting problem, we have to add regularization term with the multiplication parameter λ, which modifies our optimization model as below:

$$\underset{U \geq 0, V \geq 0}{\arg\min} \lambda(||U||_F^2 + ||V||_F^2) + \sum \mathcal{L}(X, Y) \tag{2}$$

In this equation, $||U||_F^2$ and $||V||_F^2$ are Frobenius norms of the corresponding U and V matrices. Note that the Frobenius norm is one of the simplest matrix norms. The Frobenius norm for matrix A is evaluated as:

$$||A||_F^2 = \sqrt[2]{\sum \sum \left\| a_{i,j} \right\|^2}$$

Our primary goal is to provide the recommendation of similar products to customers, which is derived through the implicit collaborative behavior of customers. The transaction matrix X is large with a high value of *m* and *n* but is highly unbalanced or sparse (mostly zeros with very few ones). As the above optimization formulation mainly considers only ones or positive (customer-product-transact-pair) entries of X, this basic formulation is not sufficient for our purpose (or good result). In any collaborative filtering model the positive labeled data (or 1 entry in retail transactions) is the primary fuel for the system and also calibrates the system performance, whereas in one-class filtering, the strength of the model will be determined by all those zero entries. However, there is a need for consideration on top of the positive labeled data because of the fact that there will be more zeros (unlabeled data) than ones (labeled data). In either the e-retail or traditional retail set-up, the zero entry (no-transaction, NT) of a customer product pair consists of any customer between both extremes of customers. For example, one is not going to purchase that particular product if she has recently purchased a substitute product (i.e. no purchase intension for that particular product even in future, absolute-negative, non-buyer, NB). Moreover, the same applies if another is considering purchasing a product very soon but has not yet purchased the product (non-negative case, potential buyer, PB).

Simple weight assignment is a technique that introduces the relative degree of importance of different entries or different groups of data sets formed due to the difference in implicit customer behavior towards the various product and product categories.

Let us modify our optimization model with weight assignment.

$$\underset{U \geq 0, V \geq 0}{\arg\min}\, \lambda(||U||_F^2 + ||V||_F^2) + \sum \mathcal{W}\mathcal{L}(X, Y) \tag{3}$$

Srebro and Jakkola (2003) applied weight-based low rank approximation in collaborative filtering with a simple model of assigning two extreme weights:the highest weight (1) on positive entries and the lowest weight (0) on other entries.

Suppose X^1 is a set that contains only the pairs (*i,j*) of all 1 entries in the actual transaction matrix X. In other words, X^1 = { (*i,j*): $X_{i,j} = 1$ } . Similarly, let

X^0represent another set, which contains only the pairs (*i,j*) of all 0 entries (no purchase) in X. So, $X^0 =$ { (*i,j*):$X_{i,j} = 0$ } .

Following Srebro and Jakkola's (2003) weight assignment, our formulation will be the following:

$$\underset{U \geq 0, V \geq 0}{\arg\min}\ \lambda(||U||_F^2 + ||V||_F^2) + \sum_{i,j \epsilon X^1} \mathcal{W}_{i,j}\, \mathcal{L}\big(X_{i,j}, u_i^T v_j\big) \qquad (4)$$

where $\mathcal{W}_{i,j} = \begin{bmatrix} 1 & \forall\ (i,j) \epsilon X^1 \\ 0 & \forall\ (i,j) \epsilon X^0 \end{bmatrix}$

Such a model is biased towards the potential buyer group or PB cases, and ignores the non-buyer group or NB cases. The models with the range of different weight assigning schemes were recently proposed (Pan et al., 2008; Sindhwani et al., 2009), and they presented a dramatic improvement in performance compared to only two extreme weight assignment models. In order to accommodate the assignment of differing weights on both types of entries for the one-class filtering model, our formulation would be modified from (4) to the following (5):

$$\underset{U \geq 0, V \geq 0}{\arg\min}\ \lambda(||U||_F^2 + ||V||_F^2) + \sum_{i,j \epsilon X^1} \mathcal{W}_{i,j}\, \mathcal{L}\big(1, u_i^T v_j\big) + \sum_{i,j \epsilon X^0} \mathcal{W}_{i,j}\, \mathcal{L}\big(0, u_i^T v_j\big) \qquad (5)$$

This formulation provides a way to consider all types of customers: perfect-buyer, potential buyer and non-buyer. The different forms of customer deliberation are modeled through the value of the weight assigned for $\mathcal{W}_{i,j}$.

The term $\sum_{i,j \epsilon X^1} \mathcal{W}_{i,j}\, \mathcal{L}\big(1, u_i^T v_j\big)$ is for positive labeled data (1) entries in the transaction (already purchased cases), which are for perfect-buyers; the weight assigned for these data should always be the highest value.

The last term $\sum_{i,j\epsilon X^0} \mathcal{W}_{i,j}\, \mathcal{L}(0, u_i^T v_j)$ is for all the unlabeled data (0) entries in the transaction (no-purchase cases). The value of the weight assigned for these datasets will influence the buyer type considerations. If the highest value is assigned, which is the same as a perfect-buyer, the remaining zero entries of customer-product pairs are also that of perfect non-buyer (NB) cases (i.e. the customers are not interested in purchasing that product in the future). This model considers the other extreme of a perfect non-buyer. Similarly, if the least value or zero is assigned as weight, then all those remaining zero entries of customer-product pairs are potential-buyer (PB) cases (i.e. the customers has strong interest and will be purchasing that product in the future). This model also considers such cases. In reality, there will always be a mix of these two types of customers, i.e. both perfect non-buyers (NB) and potential buyers (PB). The model based on formulation (5) is flexible enough to assign any weight value between these two extremes.

After having the proper assignment of $\mathcal{W}_{i,j}$ for the corresponding $X_{i,j}$, formulation (5) can be rewritten in a simplified and compact way, which is very closely related to the model of Sindhwani et al. (2009):

$$\underset{U\,\geq\,0, V\,\geq\,0}{\arg\min}\ \lambda(||U||_F^2 + \,||V||_F^2) + ||\Omega \otimes (X - UV)||_F^2 \qquad (6)$$

In this formuation $\Omega_{\mathrm{i,j}} = \sqrt{\mathcal{W}_{\mathrm{i,j}}}$, and $\otimes$ is used to denote an element wise product operation.

Lastly, following the steps explained in Lee and Seung (1999; 2000) the final optimized solution will be obtained after following the below two alternating multiplicative update steps:

$$V = V \otimes \frac{U^T(\Omega \otimes X)}{U^T(\Omega \otimes (UV)) + \lambda V} \quad (7)$$

$$U = U \otimes \frac{(\Omega \otimes X)V^T}{(\Omega \otimes (UV))V^T + \lambda U} \quad (8)$$

3.4 Weight Assignment Schemes

Every customer-product pair-related weight, $\mathcal{W}_{i,j}$ can have any non-negative (zero or any positive) value. The seven methods of weight assignment are outlined in Table 3.1. The different choices of $\mathcal{W}_{i,j}$, which were proposed in Pan et al. (2008) and Sindhwani et al. (2009) with promising results, are considered as baseline methods for comparing the results from our proposed method of temporal information integration.

3.4.1 Baseline Methods

Each baseline method is outlined below with the rationale of initialization values and other considerations.

1. Zero Weight (ZW): In this case, we assign *One*as the weight for all transaction entries and *Zero* for all no-transaction (NT) entries. Thus, the weight matrix is exactly the same as the transaction matrix. From our model perspective, the loss function calculates the error only on purchased customer-product pairs. In other words, the model considers that virtually all customers are potential buyer (PB) for all products on the shelf.

2. Full Weight (FW): In addition to all transactions entries, we also assign *One* as the weight for all no-transaction entries. As a result, the whole weight matrix is full of ones. The loss function on our formulation now calculates maximum error on both transactions as well as no-transaction entries. In this case, the model considers all the zero entries of the customer-product pair as the customer having the intention to not buy that particular product (NB).

3. Uniform Weight (UW): In contrast to the last two weight assignment schemes of two extremes, either *Zero* or *One* assignment, a small weight, $\delta \in (0, 1)$ is assigned as a weight to all those NT entries in this uniform weight scheme. The 'uniform' is to indicate that the weight assigned to all the NT entries are same, unlike other methods where every customer or product may be assigned with different weights. The rationale of this weight is to show that the confidence of a positive label, being a PB (a perfect buyer) case, is higher than the confidence of an unlabeled case (NT) being a NB. In our all experiments, we took a positive-label-rate (ratio between positive label counts and total counts) as a uniform weight value for evaluation purposes.

4. Customer-Oriented Weight (COW): In this scheme, the non-uniform weight assigned to NT cases is proportional to the customer transaction counts. The corresponding customer weight is calculated as follows:

$$\delta_{i,j} \propto \frac{\sum_j^n x_{i,j}}{\left[\max\left(\sum_j^n x_{i,j}\right)\right]},$$

Here, the division is to satisfy $\delta \in [\,0, 1]$. The rationale here is if a customer has a history of heavy purchases of many items, i.e. having many labeled data, the NT, or the unlabeled data, will bear high confidence as a NB case for this particular customer.

5. Product Oriented Weight (POW): This method is similar to the COW method with non-uniform weight assignment. However, the proportional relation is based on the transacted product count in this POW method, unlike the customer count in COW method. The corresponding product weight is calculated as follows:

$$\delta_{i,j} \propto \{1, \frac{\sum_i^m x_{i,j}}{[\max(\sum_i^m x_{i,j})]}\},$$

   Here also the count is turned into a fraction first; then only subtraction is done so as to limit the range, $\delta \in [\,0, 1]$. The rationale here is if a product observes fewer transaction counts, then most of the NT or missing data will bear high confidence in NB cases for that particular product-customer pair.

3.4.2 Proposed Methods: Temporal Weight Assignments

We propose two simple methods that capture the temporal information from the system:

1. Temporal Customer Oriented Weight (TCOW): In addition to the transaction data set, here we maintain customer recency vector, ζ^{CR}, with the record of every customer's most recent visit. This customer based temporal weight assignment is also proportional to the difference in time

periods from the current (recommendation or evaluation) period and each customer's most recent visit period. This is written below:

Let ζ^{CR} is a customer recency vector for all customers

$\Delta^{\mathrm{CR}} = \tau_c - \zeta^{\mathrm{CR}}$ where τ_c iscurrent or evaluation period

$$\delta_i^{CR} \propto [\, 1 - \frac{\Delta_i^{CR}}{\max(\Delta^{\mathrm{CR}})}\,] \qquad \{\text{Hence } \delta \in [\,0, 1]\}$$

In this formuation, δ_i^{CR} is the temporal customer recency based weight for customer *i*.

Table 3.1: Different methods of weight assignment

Weight Scheme	**Code**	**Transaction (1)**	**No-Transaction (0)**
Baseline Methods			
Zero	ZW	$\mathcal{W}_{i,j} = 1$	$\mathcal{W}_{i,j} = 0$
Full	FW	$\mathcal{W}_{i,j} = 1$	$\mathcal{W}_{i,j} = 1$
Uniform	UW	$\mathcal{W}_{i,j} = 1$	$\mathcal{W}_{i,j} = \delta(\, 0 < \delta < 1\,)$
Basic Customer Oriented	COW	$\mathcal{W}_{i,j} = 1$	$\mathcal{W}_{i,j} \propto \sum_j X_{i,j}$
Basic Product Oriented	POW	$\mathcal{W}_{i,j} = 1$	$\mathcal{W}_{i,j} \propto (m - \sum_i X_{i,j})$
Proposed Methods			
Temporal Customer Oriented	TCOW	$\mathcal{W}_{i,j} = 1$	$\mathcal{W}_{i,j} \propto \delta_i^{CR}$
Temporal - Product Oriented	TPOW	$\mathcal{W}_{i,j} = 1$	$\mathcal{W}_{i,j} \propto \delta_j^{PL}$

Each customer temporal weight is divided by the maximum weight so as to limit the range as $\delta \in [\,0, 1]$. Again, the rationale here is if this customer has recently visited the e-retailer, the confidence on his returning to shop is higher than for another customer who has not visited recently. This customer recency information is endogenous and can easily be tracked for record keeping.

2. Temporal Product Oriented Weight (TPOW): In a similar customer recency- based method, we also maintain a product launch vector, ζ^{PL} , with every product (at SKU level) launch period information. Temporal weight assignment is proportional to the difference in time: from the current (recommendation or evaluation) period, as well as each product launch period. This is presented below:

   Let ζ^{PL} is a product launch vector for all products

   $\Delta^{\mathrm{PL}} = \tau_c - \zeta^{\mathrm{PL}}$ where τ_c is current or evaluation period

   $\delta_j^{PL} \propto \frac{\Delta_j^{PL}}{max(\Delta^{\mathrm{PL}})}$ { Hence $\delta \in [\,0, 1]$}

   In this formulation, δ_j^{PL} is a temporal product launch-based weight for product *j*.

   The calculation follows the simple rationale that the longer the product is in the market, the more the product is observable to customers; thus, there will be a high confidence of NT or missing data as with PB cases. The rationale we considered is reasonable for consumer electronics, laptops and the computer market, where new models and updated versions are continuously entering into the market. However, the rationale is not perfect for the products with very long product life cycle characteristics. In any retail setup, the product launch record is endogenous and also trivial information for gathering and maintaining.

3.5 Empirical Evaluation

For all evaluations, we used the transaction record set generated through the synthetic data generator for e-commerce, which was explained in detail in chapter 2. The simulation environment is comprised of Intel PCs with Windows 7 OS with a Quad Core i-7 (2.2 GHz) processor and 8GB RAM. We have set various parameters as default values for various comparisons, and they are listed in Table 3.2. We have chosen the area-under-ROC curve (AUC) as the standard measure to compare the recommendation quality of different methods. AUC measure and associated formulation is detailed in appendix. Within a single comparison, a random dataset of transaction record set was produced and used for each method. For each run of the simulation, the latent user and product feature matrices, *U* and *V*, are initialized with random entries.

Table 3.2: Default values of different parameters for evaluations

Parameter details	**Value**
No. of Customers	5000
No. ofProducts	5000
Average no. of Transactions per period	7200
No. of Latent Factors (Rank)	3
Evaluation at Period	16
Error Tolerance Level	10^{-8}
Maximum Iteration	500

Figure 3.2 is of a typical run result for each method with the same transaction dataset with all default parameters. It is evident that the ZW method is the most inferior method with the ROC-curve being almost diagonal and not

offering any classification quality. In all other baseline methods, the ROC shapes are almost similar with some degree of improvement on performance. One of our proposed methods, the TPOW, depicted the most superior ROC-curve performance of all. Another proposed method, the TCOW, is also slightly better than other baseline methods.

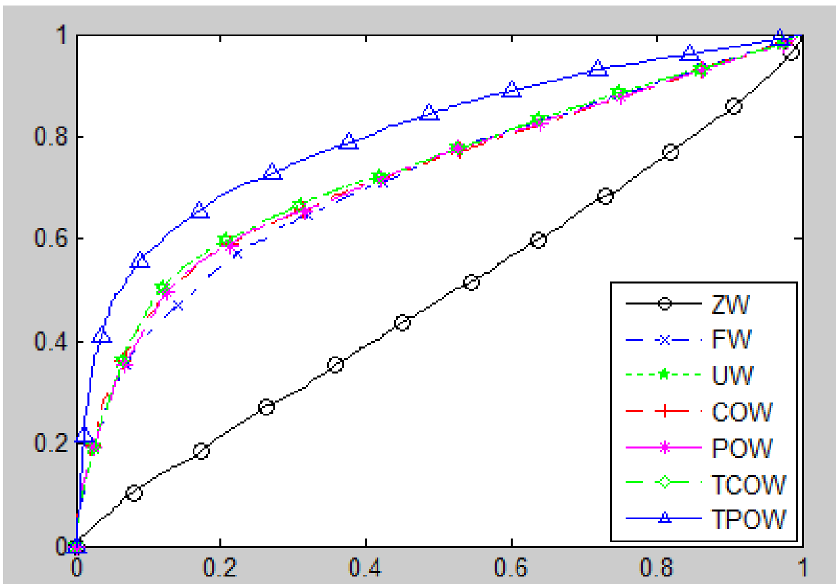

Figure 3.2: AUC comparison for different methods in a single typical run

In order to have proper comparisons among different methods, we ran 10 replicate runs for each method with all default parameters. With one single random set of transactional dataset, we evaluated area under ROC curve (AUC) of the recommendations of each method for that particular run. Repeating the procedure for 10 times with different random transactional dataset, we got the mean and standard deviation as depicted in Table 3.3. Figure 3.3 is the AUC mean value plot of the Table 3.3 dataset.

Table 3.3: Mean and s.d. of AUC of different methods under different ranks

		Ranks (Latent Factors)															
		1		2		3		4		6		9		12		15	
		μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Methods	ZW	0.517	0.035	0.505	0.005	0.498	0.0071	0.497	0.009	0.501	0.007	0.495	0.007	0.493	0.01	0.491	0.01
	FW	0.701	0.021	0.701	0.024	0.7	0.0213	0.698	0.022	0.695	0.021	0.692	0.022	0.691	0.02	0.689	0.02
	UW	0.699	0.022	0.715	0.022	0.722	0.0217	0.718	0.021	0.715	0.021	0.71	0.022	0.707	0.021	0.703	0.019
	COW	0.697	0.022	0.706	0.023	0.703	0.024	0.701	0.023	0.699	0.024	0.697	0.024	0.694	0.022	0.696	0.019
	POW	0.699	0.021	0.705	0.024	0.703	0.0224	0.704	0.022	0.7	0.024	0.7	0.024	0.698	0.021	0.696	0.019
	TCOW	0.698	0.022	0.715	0.022	0.721	0.0218	0.718	0.021	0.715	0.021	0.71	0.022	0.707	0.021	0.703	0.019
	TPOW	0.786	0.016	0.79	0.017	0.788	0.0164	0.787	0.016	0.785	0.016	0.783	0.016	0.784	0.014	0.78	0.015

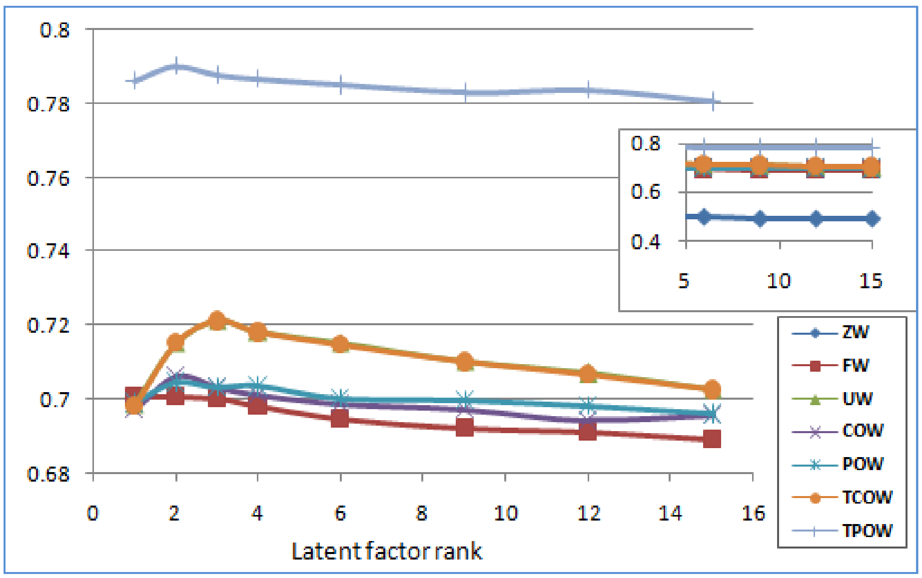

Figure 3.3: Mean AUC measures of different methods under different rank values

From the mean AUC values of Table 3.3 and Figure 3.3 plot, it is evident that method ZW is having very poor performance for recommendation purpose. Proposed methods, TPOW and TCOW exhibit better performance than others. Similarly, on the rank (number of latent factors) based evaluation, there is gradual improvement on performance as we increase the rank up to a certain point, followed by saturation and then degradation stage. In order to find the statistical significance of various methods, ranks and the possible interaction effect between method and rank are also tested with ANOVA analysis. Table 3.4 lists the ANOVA results with key indicators for the random 3 replicated runs with consideration of different methods and ranks as factor levels. Different methods demonstrate very high significance with a p-value of almost zero, whereas rank has small significance, which is also evident from the Figure 3.2 plot. We do not observe any interaction effect between method and rank. Similarly, on the same data, we did a *Tukey* comparison test, as depicted on Table 3.5 results. On this tests also, the method 1(ZW) is significantly different and the most inferior compared to other methods. FW, UW, COW, and POW (corresponding methods 2, 3, 4, and 5 on table list) method results are statistically insignificant. On the contrary, both of our proposed temporal methods, TCOW and TPOW, have a strongly significant performance compared to other methods.

Table 3.4:TWO-Way ANOVA: AUC vs. Method, Rank

General Linear Model: AUC versus Method, Rank

Factor	Type	Levels	Values
Method	fixed	7	1, 2, 3, 4, 5, 6, 7
Rank	fixed	8	1, 2, 3, 4, 6, 9, 12, 15

Analysis of Variance for AUC, using Adjusted SS for Tests

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Method	6	1.046438	1.046438	0.174406	260.64	0.000
Rank	7	0.004455	0.004455	0.000636	0.95	0.471
Method*Rank	42	0.004014	0.004014	0.000096	0.14	1.000
Error	112	0.074944	0.074944	0.000669		
Total	167	1.129850				

S = 0.0258678   R-Sq = 93.37%   R-Sq(adj) = 90.11%

After having few comparisons with all default parameters, we investigated various parameter effects on these methods with the same AUC measure. Each time we repeated the simulated runs, keeping all other parameters at default values and changing only one parameter at a time.

Table 3.5 : Statistical comparison among different methods

Tukey Simultaneous Tests
Response Variable AUC
All Pairwise Comparisons among Levels of Method

Method = 1 subtracted from:

Method	Difference of Means	SE of Difference	T-Value	Adjusted P-Value
2	0.1989	0.007467	26.64	0.0000
3	0.2110	0.007467	28.26	0.0000
4	0.2015	0.007467	26.98	0.0000
5	0.2026	0.007467	27.13	0.0000
6	0.2583	0.007467	34.59	0.0000
7	0.2358	0.007467	31.57	0.0000

Method = 2 subtracted from:

Method	Difference of Means	SE of Difference	T-Value	Adjusted P-Value
3	0.012071	0.007467	1.6165	0.6720
4	0.002545	0.007467	0.3408	0.9999
5	0.003615	0.007467	0.4841	0.9990
6	0.059347	0.007467	7.9475	0.0000
7	0.036823	0.007467	4.9312	0.0001

Method = 3 subtracted from:

Method	Difference of Means	SE of Difference	T-Value	Adjusted P-Value
4	-0.009526	0.007467	-1.276	0.8616
5	-0.008456	0.007467	-1.132	0.9167
6	0.047276	0.007467	6.331	0.0000
7	0.024752	0.007467	3.315	0.0205

Method = 4 subtracted from:

Method	Difference of Means	SE of Difference	T-Value	Adjusted P-Value
5	0.001071	0.007467	0.1434	1.0000
6	0.056803	0.007467	7.6068	0.0000
7	0.034279	0.007467	4.5905	0.0002

Method = 5 subtracted from:

Method	Difference of Means	SE of Difference	T-Value	Adjusted P-Value
6	0.05573	0.007467	7.463	0.0000
7	0.03321	0.007467	4.447	0.0004

Figure 3.4 is the result of runs with a varying number of customers. The TPOW method always is superior with a very wide gap of performance difference over other methods. As the number of customers is increasing, we observe the performance degrading for all of the methods. However, in each run, the number of products and transactions per period is random; it is very close to the default value in the average sense. Hence, an increasing number of customers makes the transaction data matrix sparser. The sparse data problem is a general problem for any matrix factorization-based method. As a result, sparse data degrades the performance of all the methods.

Similarly, in another set of runs, we increased the number of products, keeping all other parameters at default values. Figure 3.5 shows the corresponding effect of changing number of products. Here also, the reason for degradation of the performance of each method is primarily the sparsity problem. However, both of our proposed methods observe the rate of degradation as lower than other baseline methods.

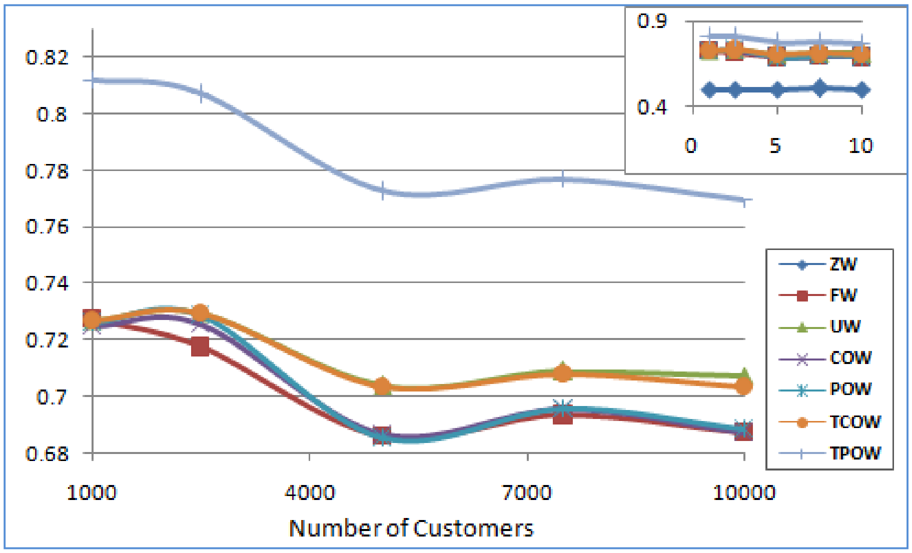

Figure 3.4: AUC of different methods with varying number of customers

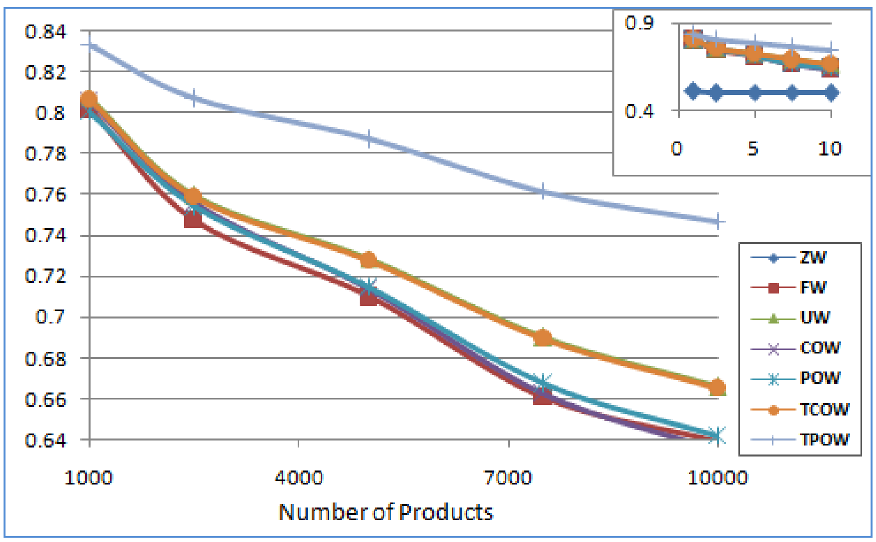

Figure 3.5: AUC of different methods with varying number of products (SKU)

We also experimented with various transaction rates. We considered 2 weeks as a single period. Considering 16 hours in a day as an average transaction activity time, the corresponding conversion rate is calculated, such as 0.2 transactions per minute equals 2880 transactions per period and 0.5 transactions per minute equals 7200 transactions per period. Figure 3.6 shows the increasing transaction rate effect on different methods. Here every method performance is increased with a higher transaction rate, which means more transactions, resulting in more labeled datasets for a better level of model learning.

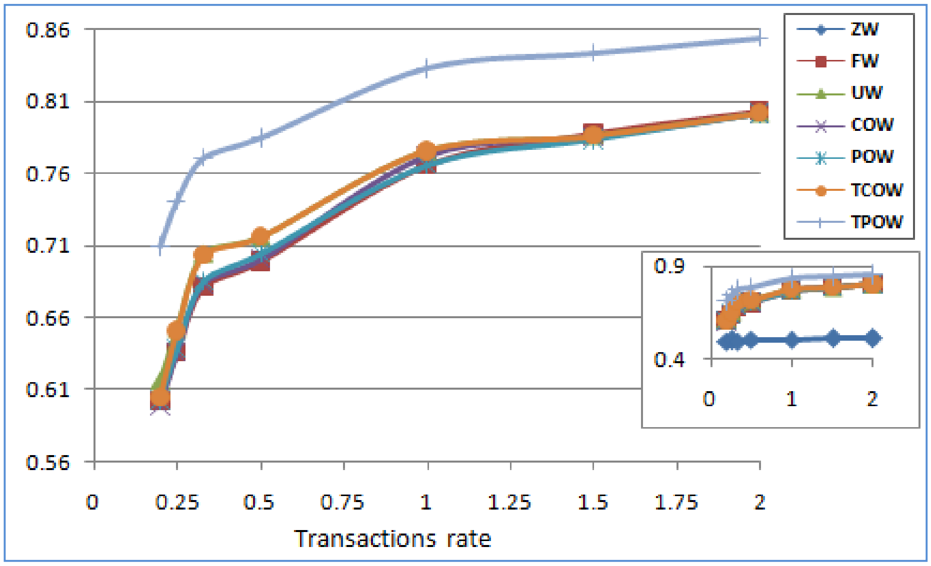

Figure 3.6: AUC of different methods under different transaction rates

Similarly, the graph in Figure 3.7 shows the evaluation at different periods. In each run, for the N^{th} period evaluation, all historical transactions up to (N-1) period transactions are available for model learning. All method performance is increasing with the availability of more history; however, the performance gap difference depicts the robustness of our proposed methods even with less history.

We also investigated various model performances from model approximation time requirements. As depicted in Figure 3.8, most of the models reached at the pre-specified error tolerance level of 10^{-8} within 100 iterations, so we set 500 as the maximum number of iterations to perform.

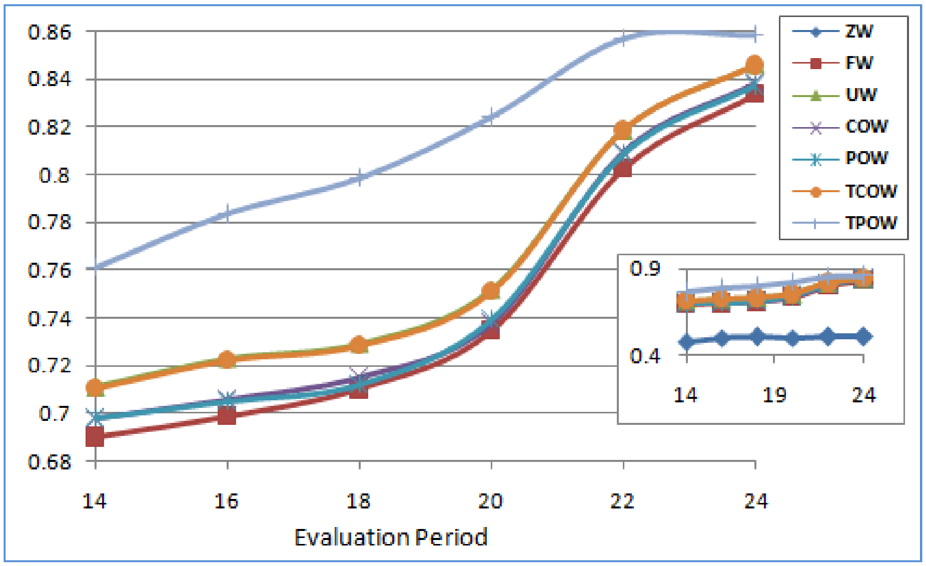

Figure 3.7: AUC of different methods at different periods

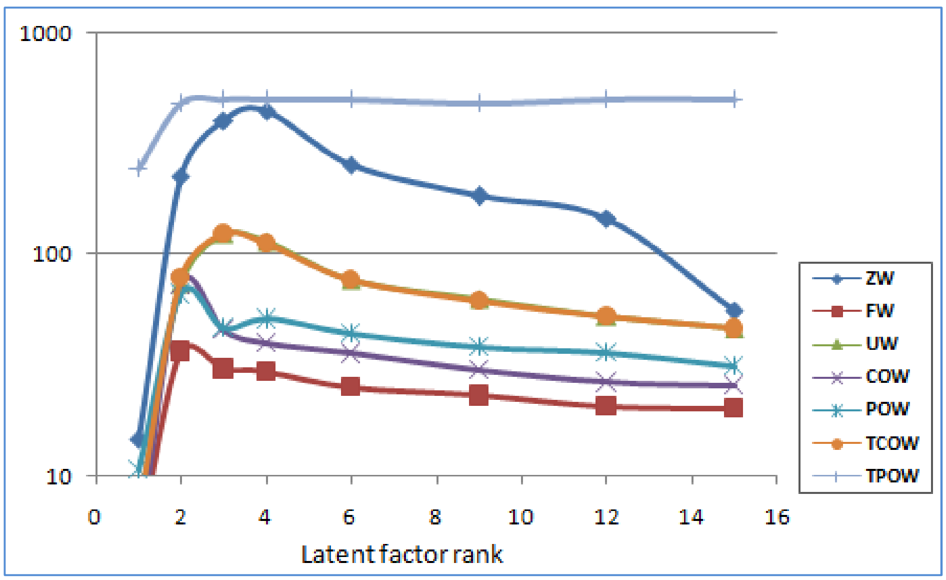

Figure 3.8: Average no. of iterations to achieve fixed error tolerance level

Our proposed method, TPOW, is going under around 200 iterations, irrespective of the number of ranks, to always arrive at superior results. Even the TCOW method is giving good results in less than 100 iterations. Overall, the number of iterations required, to achieve the fixed error tolerance level, is slightly decreasing, with the number of rank increment, which is obvious that this model gets more flexible in achieving fixed level tolerance with more factors.

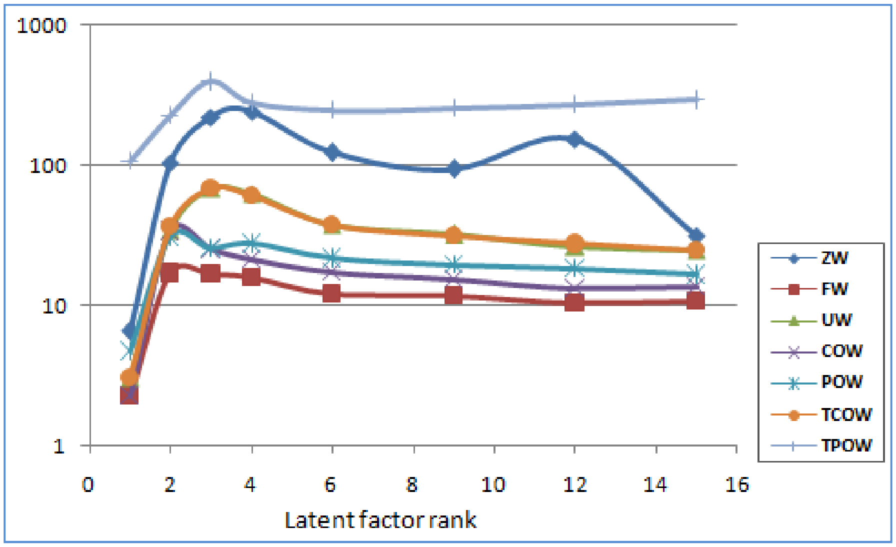

Figure 3.9: No. of seconds required to achieve fixed error tolerance level

With consistent results, Figure 3.9 depicts the average number of seconds required for different methods in achieving the fixed error tolerance level. One of the inferior methods, ZW, is taking similar time as our proposed method, TPOW, which provides the best performance among all. The TCOW and UW methods provide second best results while taking less time for the convergence of the model.

Table 3.6: Different PLC rates for different product categories

	PLC_Rate			
Product Category	**1**	**2**	**3**	**4**
CL	4	8	16	24
CP	8	16	20	24
CA	12	24	24	24

The results are illustrated on Figure 3.10. Here also our proposed TPOW and TCOW method results are better than other baseline methods. When the market is changing from shorter life cycle products to longer life cycle products, product launch-related information is less effective, but it still produces efficient results compared to other methods, as seen in Figure 3.10.

A few more simulation runs were conducted to evaluate the effect of different PLC rates, i.e. the number of periods the products will be on the shelf for display and purchase Similar to what is detailed in Table 3.6, PLC rates are fixed for comparison of different methods. The shortest PLC periods are listed as rate value 1, where CL category products are on the shelf for only 4 periods, CP category products last for 8 periods, and CA category products last for 12 periods. On the other hand, the case of products available at all times is modeled with the highest rate of 4, where every product is always available after the product launch.

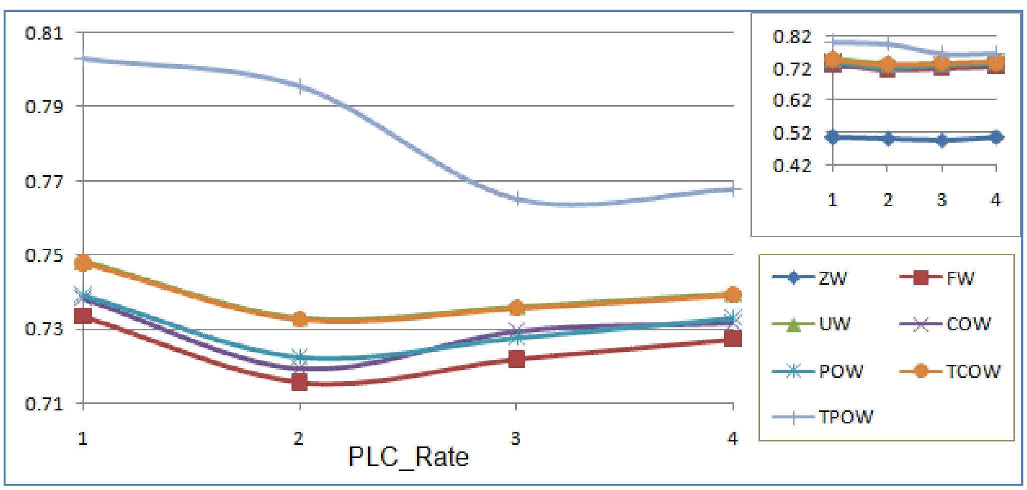

Figure 3.10: AUC of different methods under different PLC rates

3.6 Conclusion

In this chapter, we reported the development of an integrative method to capture and use temporal information through differentiated weight assignments on matrix factorization based low rank approximated methods for OCCF problems. Very limited experimentation of temporal information is being carried out in the CF domain, and that is also only on rank score-based movie recommendation problems. Two proposed methods with customer recency and product launch information are tested on synthetic e-retailer transaction record sets, with promising results. This novel technique and performance improvement on OCCF problem makes a contribution to the field and will be applicable to cross-selling, up-selling, and personalized and targeted selling within the e-retailer business domain.

CHAPTER 4

PERSONALIZED DYNAMIC BUNDLE PRICING

4.1 Introduction

Cross-selling has emerged as a key issue in contemporary business products or services. These days, during the process of purchasing a particular item, the customer is recommended to purchase additional items with or without some discounts. It is a very common practice in almost every sector of business. Examples include travel packages (hotel with air ticket), insurance services (home with life), apparel (tie with shirt), restaurants (soda with sandwich),and consumer goods (memory card with digital camera). Similar concepts are bundling, cross-selling, and up-selling, which is considered to be an interesting research issue in economics, marketing, and operations management. The central theme of any related study is either to identify the optimal packaging complement or to derive the formulation for fixing the optimal discounted price of the packaged bundle, or both. In fact, bundles might offer added value through product bundling with product integration or price bundling with discounts. Similarly, depending on the time of the decision for the optimal package components and the optimal price, research studies are categorized into two types: static and dynamic. The discounted price and the components of the bundles or packages are already fixed before the customer arrival process in static cross-selling (or bundling), whereas both actions are optimized only after the customer initiates the purchasing process in dynamic cross-selling, which makes the problem very challenging and complex.

Traditional brick-and-mortar shopping malls offer only static and pre-packaged bundles because of the high implementation cost and incurred time delay for dynamic packaging and pricing decisions. In order to provide dynamic cross-selling, both of the decisions, associate item finding and discount calculation, have to be started only after the customer initiates the purchasing process for the first item and have to be completed before the payment (or check-out) process. In order to carry out the whole calculation in a brief duration of time, this type of dynamic nature of the problem requires a suitable environment. Similarly, the customer should be presented with personalized bundles based on her interest and her first item selection, which also necessitates offering more than a traditional brick-and-mortar retail store. The e-commerce-based online shopping setup offers a much better and implementable environment for the application of dynamic packaging as well as dynamic pricing.

There was an 8-fold increment of mobile-cellular device use globally from the year 2000 to the year 2011, and there was a 6-fold increment of Internet use during the same period[‡‡‡]. According to a National Retail Foundation report, the U.S. Department of Commerce estimates almost two-thirds of the U.S. GDP comes from retail consumption[§§§]. There is a forecast of almost a trillion dollar volume of global business through e-retail, and almost a quarter of it will be conducted within the U.S. by 2013[****]. With shrinking profit margins and competition not only from neighboring retailers and overseas e-retailers as well,

[‡‡‡] The World in 2011: ICT Facts and Figures ; http://www.itu.int/ITU-D/ict/

[§§§] (http://www.nrf.com/modules.php?name=Pages&sp_id=1215)

[****] (http://www.internetretailer.com/trends/sales/)

e-retailers always need to look for various ways to boost their sales without incurring extra costs. One such technique is cross-selling.

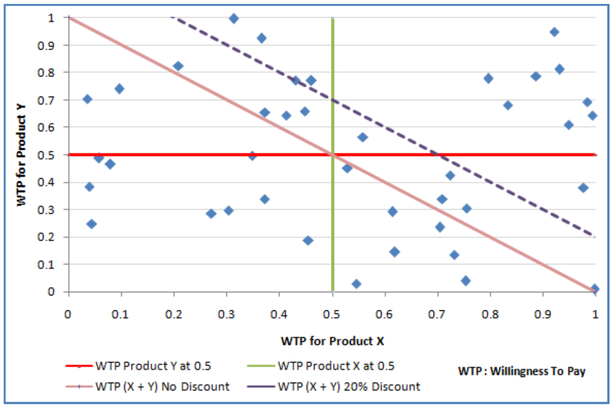

Figure 4.1: Raised WTP with bundle discount on dynamic cross-selling

As shown in Figure 4.1, consider there are two products, X and Y. Based on each customer's needs and evaluation, they purchase a certain product only at a certain price, which is also termed as willingness to pay (WTP) in some literature and also reservation price in other literature. For two products, there are four quadrants formed with 0.5 WTPs for each product. Customers in the lower left quadrant are those who will purchase both products at the maximum price at 0.5 units. Similarly, in the upper left quadrant are the customers who will pay more than 0.5 units for product Y but not for product X. On the other hand, customers in the the lower right quadrant are willing to pay more than 0.5 for product X but not for product Y. Customers in the upper right quadrant are willing to pay more than 0.5 for both products. The solid diagonal line denotes the WTP of 1 for a bundle purchase, which consists of both products X and Y. Customers below this line are ready to pay, in total 1, for a bundle purchase. Similarly, customers falling below the dashed line are willing to pay the maximum of 1.2. An e-retailer providing a 20% discount on a bundle purchase is equivalent to virtually raising the WTP of customers from 1 to 1.2. One can observe the extra customers between these two diagonal lines, bringing extra revenue from additional sales generated through these additional customers.

In fact, dynamic cross-selling with a price discount is an effective strategy to entice spontaneous and extra purchases in addition to planned purchases, which is also defined as an impulse purchase in marketing literature. Hausman (2000) summarized his results of research on impulse purchases with a claim that almost ninety percent of people make occasional impulse purchases, and he also found that purchasers considered almost fifty percent of their purchases to be impulse purchases. Similarly, Beatty and Ferrell (1998) listed situational variables (including time and price) and various personalized variables (including interest, enjoyment and buying tendency) as primary factors, which influence impulse buying.

In this chapter we propose a personalized dynamic bundle pricing (PDBP) model which generates the optimized amount of discounts based on the transaction history of each customer with consideration for product hierarchy and the price consciousness of the customer. The model generated discount is fully dynamic and unique at the level of a particular customer and a particular product. We consider the e-retailer environment, where the real-time bundle formed by an appropriate cross-selling follow-up product with a derived bundle price using a model-based discount, could be offered for online customers. The model performance in revenue rate is compared with other baseline methods including static and various loyalty-based methods on a synthetic dataset generated through an e-commerce simulator, which was explained in detail in chapter 2. The proposed model generates higher revenue than any other methods.

The rest of the chapter is organized as follows. In section 4.2, we present and review the related literature. Similarly, in 4.3, the proposed PPH model with other loyalty methods and their detail formulations are discussed. Various simulated runs and comparative results are illustrated in 4.4, with the conclusion in 4.5.

4.2 Literature Review

In this section, we provide the review of research, mostly related to bundling, cross-selling and personalization. Among the various degrees of price discrimination, the most complex but equally promising is the third degree of price discrimination. Personalization is central on recommendation system in identifying customer product pairs, whereas dynamic bundle pricing extends even one step further requiring differentiation among the relations formed by all customers, products and prices.

4.2.1 Bundling and Dynamic Pricing Discrimination

One of the very first concepts of dynamic cross-selling that is very closely related to our study is bundling, a common practice in marketing. In fact, bundling strategies have been widely practiced mainly because of the gains to both parties: savings for the customers (Yadav and Monroe, 1993; Estelami, 1999) and extra revenue with increased demand for the sellers (Lawless,1991). In addition, customer benefits extend to reduction in time and cognitive effort required on unfamiliar products (Moriarty and Kosnik, 1989) and fewer hassles due to consolidation of overall activities. In addition, the vendor benefits through reduced logistics costs (Eppen et al., 1991), differentiation among peer competitors, and building new markets (Ovans, 1997). Among all, we mainly consider the two streams of literature; the first one focuses on customer reservation price or WTP, and the second one focuses on the correlation among component demands. Stigler (1968) presents bundling as a price discrimination tool. Adams and Yellen (1976) consider the following three different sales strategies with price discrimination: pure or unbundled components (separately priced and sold), pure bundle (either sold together or not at all), and mixed bundle (sold separately as well as bundled). Schmalensee (1984) and Venkatesh and Mahajan (1993) compare the mixed bundle strategy with pure-bundled and unbundled strategies and conclude that the mixed bundle strategy is superior to other strategies in terms of overall profit. McCardle et al. (2007) investigate bundling profitability over other three parameters: individual product demand, bundle cost, and the nature of the relationship between the two products to be bundled. They also show some typical cases with negative profit. Overall, the opportunity of generating higher profit with improved efficiency in logistic-related costs and resources are realized even with static bundling.

Another aspect of cross-selling is the discounted price of the bundle, termed as dynamic pricing, which is also one of the widely practiced revenue management techniques in the airline and hospitality-related service industries (McGill and Van Ryzin 1999; Bitran and Caldentey 2003). There are also static pricing models for bundling in the marketing literature (Rao, 1993), but there are no dynamic models dealing with cross-selling issues. Elmaghraby and Keskinocak (2003) present a detailed survey of dynamic pricing practices in operations management with inventory considerations. They cite many applications with the conclusion that there is no literature about customized pricing; thus, we consider that dynamic cross-selling is one example to fill the gap. Other publications within this stream typically consider single-product dynamic pricing (Gallego and Van Ryzin, 1994;Aviv and Pazgal, 2005) but neglect dynamic pricing of bundles formed after cross-selling.

4.2.2 Personalized Price Discrimination

The attempt to use cross-selling without consideration of customer needs and interest matching may turn out to be counter-productive because of careless pushing attempt for more products. On the other hand, even with unsuccessful transactions, the personalized cross-selling process exploits the opportunity to detail the range of products and services to the target customer.

There are various industry practices which imitate the dynamics of cross-selling. One of the very common techniques is offering free shipping on a specified purchase amount. If a product order totals $120 and an inflated shipping charge of $25 is free on orders of over $150, customers have the incentive to find an extra item to purchase to take advantage of this free shipping offer. Though the technique is for boosting sales, it is not dynamic cross-selling because it lacks personalization and the mode of offer is static and prefixed. Similarly, another popular technique is through discount coupons, which may fall into the broader category of personalizing, but it is still static with a prefixed value of discounts.

Before formulation and implementation of such personalized discriminatory pricing, two natural questions arise. The first one is whether such a discriminatory pricing policy is legal, and the second one is whether customer response remains the same after they become aware of such discrimination.

The company, VS and Katzman case ruling is that it is sufficient for any retailer or e-retailer to practice personalized dynamic pricing (Weiss andMehrotra, 2001). There are a few reported attempts of practicing price discrimination based on the location of customers, and another few are based on the number of visits to e-commerce sites (Aydin and Ziya, 2009).

From Amazon’s dynamic pricing experiments (Streitfeld, 2000), it appears that the real challenge for e-retailers is to come up with some rationale for their dynamic pricing, which should not be perceived as unfair treatment to their loyal customers. According to a survey by Turow et al. (2005), less than 30% of customers know that it is legal for both offline and online store to charge different people different prices at the same time of day. In fact, there is already personalized pricing practice based on demographics like student and senior discounts, which are commonly accepted as fair practice. Similarly, these days, two different customers flying on the same flight and staying in similar hotel rooms are likely to pay different airfares and different hotel bills, yet no one objects. The airline industry was the first to have a computerized ticket system in the 1950s, and it started experimenting with dynamic price discrimination in the

1980s (McAfee 2007). It is likely that the airlines and hospitality business sector has already convinced their customers about the fairness of price discrimination, but the retail sector has yet to come out of the coupon periphery and start personalized price discrimination experiments. We believe that this research attempts to fill this gap.

4.3 Proposed Model: Price Product Hierarchy

In this section we explain the proposed price product hierarchy (PPH) model and show the step-by-step formulation and model learning process following the novel and clean optimization technique of matrix factorization. We also provide the details of the discount allocation process. The static method and three loyalty-based methods are considered as baseline methods for comparing the proposed method performance. Each of these methods with discounting steps is also explained.

4.3.1 Notations

For the convenience of discussion, we will first introduce the annotation. Let us define a few matrices first.

X : full transaction matrix with binary data,{1 : purchase; 0 :no purchase }

$\mathcal{H}$: PPH matrix with normalized count fraction (maximum 1)

U : user feature matrix with latent customer features

V : PPH feature matrix with latent features of hierarchical groups

$\mathcal{I}$: resultant, learnt matrix with customer interest on PPH groups

D : discount vector with fixed increasing steps,

where $\forall D \in [0, D_{max}]: D_i < D_{i+1}$

All of the above matrices have non-negative entries.

4.3.2 MF based Customer Interest Learning

Let us suppose there are *m* customers and *n* products. Thus, in X, the actual transaction matrix or customer-product (at the SKU level) matrix, there will be *m* rows and *n* columns. The entry of 1 in X is to indicate the customer *i* out of *m* purchased a product *j* out of *n*. Similarly, the 0 entry is for the no purchase indication of that particular customer-product pair. This X is large but sparse; there are many zeroes and very few ones, which is also termed as unbalanced. The attempt of approximation on customer interest towards each product at the SKU-level was thoroughly worked out in the last chapter as a task of customer wise product recommendation. Our objective here is to approximately match the price discount to be offered on products with a customer interest on potential cross-selling products in order to boost the sales and the overall profit for an e-retailer.

Full binary purchase matrix with each customer-product at the SKU level pair X, is collapsed into a product-price-hierarchy, PPH based count matrix.

Table 4.1: Sample PPH indexing

Product Category	Price range	PPH Index
Desktop	High	1
Desktop	Medium	2
Desktop	Low	3
Laptop	High	4
Laptop	Medium	5
Laptop	Low	6
Accessories	High	7
Accessories	Medium	8
Accessories	Low	9

Sample PPH grouping and indexing with three product categories and three price ranges is illustrated in Table 4.1. This aggregation through PPH collapse trades off the sparsity problem of the full purchase matrix to arrive at meaningful learning.

Individual customer purchase counts of a particular PPH group are maintained in the $\mathcal{H}$ matrix. With this PPH grouping, *n* products are grouped into only *g* groups, and *g*<<*n* .The rows of the $\mathcal{H}$ matrix remains the same as $X$the matrix whereas columns of $\mathcal{H}$ matrix reduce to only *g* from the very large value of *n* of column of matrix X. Customer interests on PPH groups are actually distributed over the $\mathcal{H}$ matrix and maintained as the interactions between customers and products. In fact, from the model aspect $\mathcal{H}$ is the interaction of customers and PPH groups, which are not just products or product groups; rather, it captures the price groups also. Using the matrix factorization method, the interactions between customers and PPH groups are mapped into the new joint latent factor space formed by latent customer features and latent PPH features. These PPH features are the composite features representing product and price hierarchy information. Let us consider U as a matrix representation of customer features, following the usual notations, $U = [u_1, u_2, \dots., u_m]^T$ is $m \times k$ matrix, where the i^{th} row of U is a customer, u_i , who is represented in the *k*-dimensional customer feature space. Similarly, consider V as a matrix representation of PPH features. Then again, $V = [v_1, v_2, \dots., v_g]$ is $k \times g$ matrix, where the j^{th} column of V is a product, v_j , which is represented in the *k*-

dimensional PPH feature space. Here, this *k* is termed as the rank of the factorization, which is the number of latent features to be worked with.

The dot product $u_i{}^T v_j$captures the degree of interest observed by the user u_i towards the PPH group v_j in the joint latent feature space. Let us suppose this product as interest, $\mathcal{I} = UV$.Now, estimating the degree of customer interest on PPH groups, $\mathcal{I}$, turns into a simple optimization problem as shown below.

$$\underset{U\geq 0, V\geq 0}{\arg\min} \sum \mathcal{L}(\mathcal{H}, \mathcal{I}) \qquad (1)$$

Here $\mathcal{L}$ is either a squared error function or any other loss function as listed below,

With a measure of squared error:

$$\mathcal{L}(\mathcal{H}, \mathcal{I}) = ||H - \mathcal{I}\,||^2 = \sum_{i=1,j=1}^{m,g} \left(H_{i,j} - \mathcal{I}_{i,j}\right)^2 \qquad (1a)$$

With a measure of KL divergence loss:

$$\mathcal{L}(\mathcal{H}, \mathcal{I}) = D(\mathcal{H}||\mathcal{I}) = \sum_{i=1,j=1}^{m,g} (H_{i,j} log \frac{H_{i,j}}{\mathcal{I}_{i,j}} - H_{i,j} + \mathcal{I}_{i,j}) \qquad (1b)$$

And also to overcome the over-fitting problem, we have to add regularization terms with multiplication parameter, λ, which modifies our optimization model as follows:

$$\underset{U\geq 0, V\geq 0}{\arg\min} \lambda(||U||_F^2 + ||V||_F^2) + \sum \mathcal{L}(\mathcal{H}, \mathcal{I}) \qquad (2)$$

where $||U||_F^2$ and $||V||_F^2$ are Frobenius norms of the corresponding U and V matrices.

Note that the Frobenius norm is one of the simplest matrix norms and is defined for a matrix A as:

$$||A||_F^2 = \sqrt[2]{\sum\sum\left\|a_{i,j}\right\|^2}$$

Last, following the steps explained as in (Lee and Seung, 1999; Lee and Seung, 2000) the final optimized solution will be obtained after using the two alternating multiplicative update steps written below, where $\otimes$ is to denote as element wise product operation.

$$V = V \otimes \frac{U^T \mathcal{H}}{U^T(UV)+\lambda V} \tag{3}$$

$$U = U \otimes \frac{\mathcal{H} V^T}{(UV)V^T+\lambda U} \tag{4}$$

Our primary goal here is to estimate the customer interest on various PPH groups, which is derived through learning from the past implicit collaborative behavior of customers expressed in the form of purchase actions of these PPH group products. The customer interest on PPH groups is now available in a matrix $\mathcal{I} = UV$. Each row provides the list of individual customer interests with PPH groups.

4.4 Dynamic Discount Assignment

Let D be a discount vector with different discount levels in increasing order with the maximum value of discount of D_{max}. Thus, the discount vector detail follows as below:

$$D = \left(D_1, D_2, \dots \dots, D_f\right); D_i < D_{i+1}; D_f = D_{max} \tag{5}$$

Similarly, on the interest matrix, $\mathcal{I}$,every row vector data is an individual customer's list of interest towards each PPH group. Following our previous notations, there are *m* customers and g PPH groups, so $\mathcal{I}_{i,j}$ represents the interest of customer *i* (out of *m* customers) towards the PPH group *j* (out of g groups).

Considering a single customer, *i* :

$$\mathcal{I}_{i,j} = (\mathcal{I}_{i,1}, \mathcal{I}_{i,2}, \mathcal{I}_{i,3}, \ldots . \mathcal{I}_{i,g}); \;\; \mathcal{I}_{i,max} = maximum(\mathcal{I}_{i,j}) \tag{6}$$

Let us define $\delta_{\mathcal{I}}$ as the conversion factor, which matches the degree of interest with the range of $[0,1]$ to the corresponding discount steps with the range of $[D_{min}, D_{max}]$. Let us make D_{min} be zero and the default discount level so that no discount case also turns to the D_{min} discount case. The discount to be offered to the customer for a particular PPH group product is directly relative to the customer interest on that PPH group, which is learned through collaboratively using the matrix factorization method as detailed above. A simple conversion from the interest value to the discount level offer is given as below.

$$D_{offer} = \frac{\mathcal{I}_{i,j}}{\mathcal{I}_{i,max}} * \delta_{\mathcal{I}} \tag{7}$$

The conversion from (7) results in not only different discount offers for different customers for the same product but also different discount offers for the same customer for different products, which is not possible in any of the other baseline methods as discussed below.

4.4.1 Baseline Methods

Dynamic price discounts are based on the concept of customer relationship management, where the customers that bring more value to an e-retailer are offered higher levels of discounts. We consider the current industry standard for customer loyalty measures, RFM (recency, frequency, and monetary), as the baseline method for comparing our proposed method. Bult and Wansbeek (1995) use RFM ranks as measures for customer valuation for the task of making optimal decisions on whether to send promotion mail to a particular customer or not. The RFM scores are very useful for the task of estimating a particular customer falling into one among multiple segments. By dividing customers into various groups, retailers can provide personalized promotion offers to those customers who are more likely to respond to such offers.

Table 4.2: Different methods with their limitations on the dynamic discount offer

Methods	**Dynamic**	**Personalization**	**Product Level Distinction**	**Collaborative Effect Consideration**	**Price Consideration**
Static	No	No	*Possible*	No	No
Loyalty Recency	Yes	Yes	No	No	No
Loyalty Frequency	Yes	Yes	No	No	No
Loyalty Monetary	Yes	Yes	No	No	No
PPH (Proposed)	Yes	Yes	Yes	Yes	Yes

The RFM measures are based on past histories. Customers who purchased recently are more likely to buy again versus customers who have not purchased in a while. Similarly, customers who purchase frequently are also more likely to buy again versus customers who do so only occasionally. Customers who spend more money while purchasing may come for another purchase soon. In all three cases, the valuable customers, with high scores, tend to continue to become even more valuable.

Table 4.2 lists the different methods and their limitations on discrimination granularities on dynamic discounting. The static method lacks all features. In brick-and-mortar stores, certain products or product group items can be offered with discounts, which are done in a static way and one by one, which is indicated as *possible* in the table. All three loyalty-based methods are dynamic, and they discriminate at the customer level. The proposed method, PPH, intuitively captures product and price level hierarchical information and also provides discrimination. The latent feature learning process utilizes the collaborative information from the transaction histories.

Each baseline method and their discount offer policy is explained as below.

1. Static: In this method, the e-retailer chooses one of the various pre-fixed discount level values from the discount vector, D , and applies it to all of the customers. The policy applies on either an “all” or “nothing” basis, which is one of the very simple methods to apply and a de-facto common practice in all brick-and-mortar retailers as well as the e-retailer business. In our evaluation of the simulation runs, the selected discount level is applied to all of the customers and all of the products. As the policy is

neither able to make any distinction between customers nor the products, the policy is termed as static.

2. LR (Loyalty-Recency): One of the techniques among loyalty-based methods evaluates the recency measure for distinguishing among customers. Using the historical transaction records, the latest visit period of each customer is considered as the recency measure data from all of the customers. All of the non-negative recency data is converted into the range of [0, 1] with division from the maximum recency value and recorded as the recency index for later use. The mean and standard deviation from all of the recency indexes are also calculated. Now, let us suppose $\mathcal{R}_i$ is the recency index of customer *i*, and $\mathcal{R}_\sigma$ is the mean and standard deviation of the recency indexes. The recency score, $\mathcal{R}_z$, is calculated as below, and the $\delta_{\mathcal{R}}$, as a conversion factor, which converts the recency score; most of them fall within the range of $[-3,3]$ to the corresponding discount steps with the range of $[D_{min}, D_{max}]$ through the below relation.

$$D_{offer} = \mathcal{R}_z * \delta_{\mathcal{R}} = \frac{\mathcal{R}_i - \mathcal{R}_\mu}{\mathcal{R}_\sigma} * \delta_{\mathcal{R}} \quad (8)$$

This method provides different discount offers to different customers even for the same product, depending on how recently the customer has done previous transactions with the e-retailer. The customer recency score maintains the same irrespective of the target cross-selling product.

3. LF (Loyalty-Frequency) : Frequency count is another popular technique in measuring customer loyalty towards the retailer. How frequently a customer does business with a retailer is considered a measure of loyalty. From past historical transactions, the greater the purchase count (frequency) of a customer, the greater the loyalty to the retailer. Similar to the recency index calculation, each customer visit frequency is divided by the maximum frequency value to make each customer frequency index to be within [0, 1]. Finally, considering all frequency indexes, the mean and standard deviation values are calculated. Let us suppose $\mathcal{F}_i$ is the frequency index of customer *i*, and $\mathcal{F}_\mu$ and $\mathcal{F}_\sigma$ are the mean and standard deviation of the frequency indexes. The frequency score, $\mathcal{F}_z$, is calculated as below, and the $\delta_\mathcal{F}$,as a conversion factor, which converts the frequency score, with most of them falling within the range of $[-3\,,3]$to corresponding discount steps with the range of $[D_{min}\,, D_{max}]$ through the below relation.

$$D_{offer} = \mathcal{F}_z * \delta_\mathcal{M} = \frac{\mathcal{F}_i - \mathcal{F}_\mu}{\mathcal{F}_\sigma} * \delta_\mathcal{F} \qquad (9)$$

Depending on the frequency of purchases done from a customer to the e-retailer, this method also provides personalized and different discount offers for different customers even for the same product. The product level differentiated discount offer to the same customer is not possible through this method.

Table 4.3 : Different methods of comparison

Methods	**Code**	**Measure Index**	**Discount Distinction**
Static	Static	None	None
Loyalty Methods			
Recency	LR	$\mathcal{R}_i$	Customer
Frequency	LF	$\mathcal{F}_i$	Customer
Monetary	LM	$\mathcal{M}_i$	Customer
Proposed Method			
Price Product Hierarchy	PPH	$\mathcal{I}_{i,j}$	Customer, Product

4. LM (Loyalty-Monetary): In this method, the monetary amount of the transactions from the customer is considered as a loyalty indicator towards the e-retailer. Similar to the other two loyalty-based methods, here also, first, the individually contributed monetary amount is summed up, from the historical transactional records, followed by dividing by the maximum value to all, forcing the individual monetary index value to be within [0, 1]. Using these monetary indexes, the mean and standard deviation values are also calculated. Here also suppose $\mathcal{M}_i$ is the monetary index of customer *i*, and $\mathcal{M}_\mu$ and $\mathcal{M}_\sigma$ are the mean and standard deviation of the monetary indexes. Similarly, the monetary score, $\mathcal{M}_z$, is calculated as below, and the $\delta_\mathcal{M}$, as a conversion factor, which converts the monetary score, with most of them falling within the range of $[-3,3]$ to

corresponding discount steps with range of [D_{min} , D_{max}] through the below relation.

$$D_{offer} = \mathcal{M}_z * \delta_{\mathcal{M}} = \frac{\mathcal{M}_i - \mathcal{M}_\mu}{\mathcal{M}_\sigma} * \delta_{\mathcal{M}} \tag{10}$$

Similar to the previous two loyalty methods, this method also makes a distinction among customers with their differing monetary scores and offers personalized discounts to the customers even for the same product but lacks in making distinctions among products for a single customer.

4.5 Empirical Evaluation

In this section, we present the evaluation results of both the proposed and baseline methods. For all evaluations we used the transaction record set generated through the synthetic data generator for e-commerce, which is explained in detail in chapter 2. The simulator run environment is Intel PCs with Windows 7 OS, with, Quad Core i-7 (2.2 GHz) processor and 8GB RAM. We have set various parameters as default values, which are listed in Table 4.4.

Table 4.4 : Default parameters for simulation runs

Parameter details	**Value**
No. of Customers	1000
No. of Products	1000
Average no. of Transactions per period	1500
Transaction History Periods	20
No. of Evaluation Period	1
Discount (percent)	20
Customer Segment	3
Product Category	3
Price-Levels	3

On each simulated run, using all of the default parameters, complete N period transactions are generated. During the transaction generation process, we do not consider the reservation threshold (or willingness to pay, WTP) of the customer, so there is no discarding of any product on the basis of price. We consider the price of the product, discount offer to customer for any potential cross-selling product and customer reservation threshold, which is a random value assigned while customer member generation process, only at the evaluation stage. We discard the first two period transactions as simulation warm-ups. We consider everything up to period K as history and evaluate the $(K+1)^{th}$ period transaction record set. In this evaluation period, we also consider only multi-product transaction cases as cross-selling. In our simulation runs, such cross-selling potential, multi-product transactions are almost one-third the total numbers of transactions within that particular test period. Among the products within two-product transactions, the higher cost product is considered as the first product, and the lower cost product is considered as the follow-up product of that transaction. Different methods, as listed below, considering the follow-up product as the potential cross-selling product, generate the discount offer to the customer on such a product. Every customer will decide whether to purchase that extra cross-selling product based on her reservation threshold (also termed as willingness to pay, WTP) and the offered discount level on the regular price of that particular product. If the discount offered meets the reservation threshold value, the cross-selling happens; otherwise, it is considered as product discards from the customer and no purchasing happens, which makes no cross-sell revenue on such events. In order to have a proper comparison among multiple and randomly simulated run results, all absolute counts, or dollar amounts either in revenue or in customer savings, are first converted into a ratio. Ideally, if everyone also purchases the follow-up products without any discount consideration and their WTP or reservation threshold, the revenue ratio turns to 100% or fractional revenue to 1. If a customer reservation threshold is lower than the offered discount, we consider such cases as the customer discarding the product because of the price and calculate it as revenue loss with the amount of the regular price of that particular product. On the other hand, if a customer reservation threshold is higher than the discount, the transaction happens with the offer of a method specific discount. In such a case, the revenue is calculated after subtracting the discount amount of the original price of that particular product. Most of our evaluations based on different methods are compared on the realized (achieved) fractional revenue.

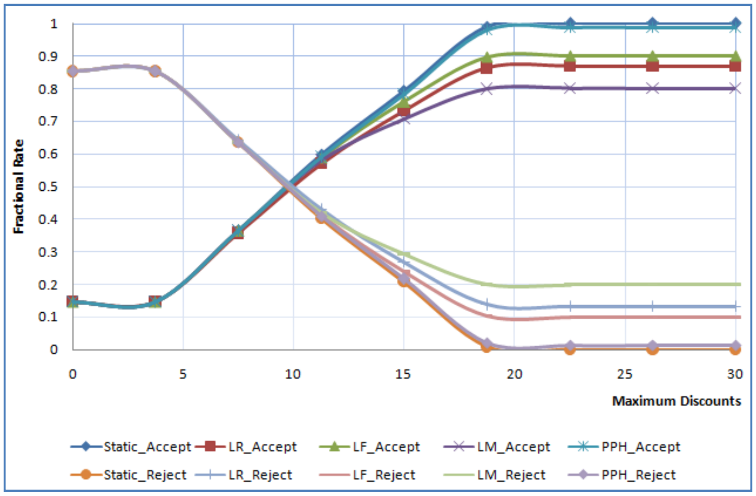

Figure 4.2 : Acceptance – Rejection rate versus maximum discounts

Each potential cross-selling product is offered with a certain amount of discount level ranging from zero to thirty percent. However, for clarity, our evaluations are compared with pre-fixed discount levels; the model is flexible enough for experimentation of continuous discounts within certain ranges. Figure 4.2 depicts the overall customer response to cross-selling products with accept-reject ratio measures, which are the ratio of accepted and rejected product counts to the total number of cross-sell potential products, at various discount levels. At lower discount levels, all methods have similar results. Differing performances are observed only on higher value discount offers. As we restricted the reservation threshold of customers between 0.7 and 1.0, with a 30% discount offer all the product were sold, and the static method achieved 100% acceptance rate.

In the static method, all customers are offered an equal amount of discount on all of the products, so one could achieve a full acceptance rate and a zero rejection rate. The remaining methods provide differentiated discount offers based on their corresponding measure, such as recency, frequency, monetary, and interest indexes, so there are also some rejections. However, the normal trend of the higher the discount, the higher the acceptance rate and the lower the rejection rate follows on all four methods, i.e. LR, LF, LM, PPH, with the proposed PPH method showing a better acceptance rate. In order to achieve the same level of fractional revenue, how much the net average discounts have to be offered to the customers is illustrated in Figures 4.3.a and 4.3.b.

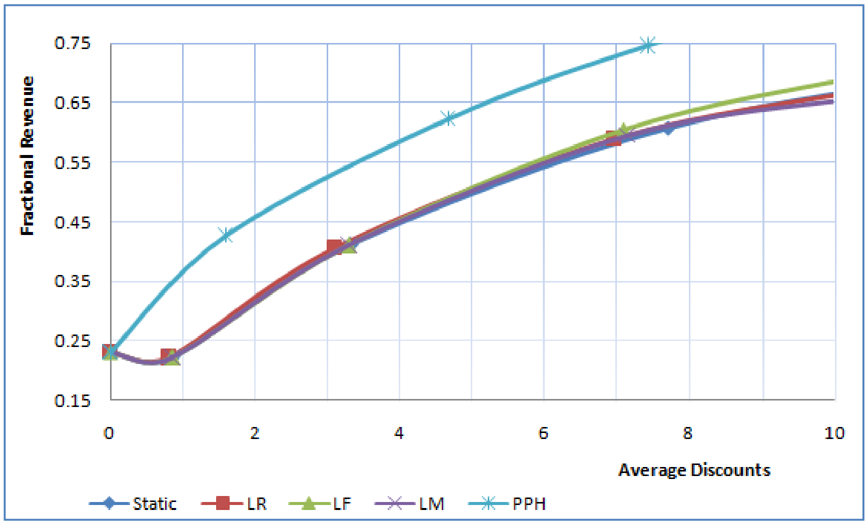

Figure 4.3.a : Fractional revenue versus average discounts (up to 10 % discount)

More than 10% of the extra revenue could be generated with the proposed method in comparison to the other methods, with the same amount of average discounts being passed on to customers. This positive difference even grows beyond the average of 5% discount levels, as depicted in Figure 4.3.a.

Figure 4.3.b clearly demonstrates the difference and significance of static versus dynamic pricing policy benefits. These differences are not observable at the lower values of discount levels. All five methods are evaluated with maximum of 30% discount offers to the customers. The static method, which is the current de-facto standard method for all retailers and e-retailers, lacks in differentiating among customers and passes the same maximum discounts to all customers. Loyalty based methods (LR, LF, and LM) and the proposed PPH method can offer personalized and different discount levels (up to maximum discounts) to different customers. There are two straight insights emerging from Figure 4.3.b. First but most important, in the static method, the e-retailer announcing maximum discounts on products is the exactly same as average discounts that have been

passed onto the customer. However, in other methods with the possibility of offering personalized discounts, the average discounts passed on to customers are far less than the e-retailer announced maximum discount. Lowering this average discount passed onto customers actually boosts the revenue without any extra effort. The second one is a general insight that revenue can be increased with discount offered only up to a certain point.

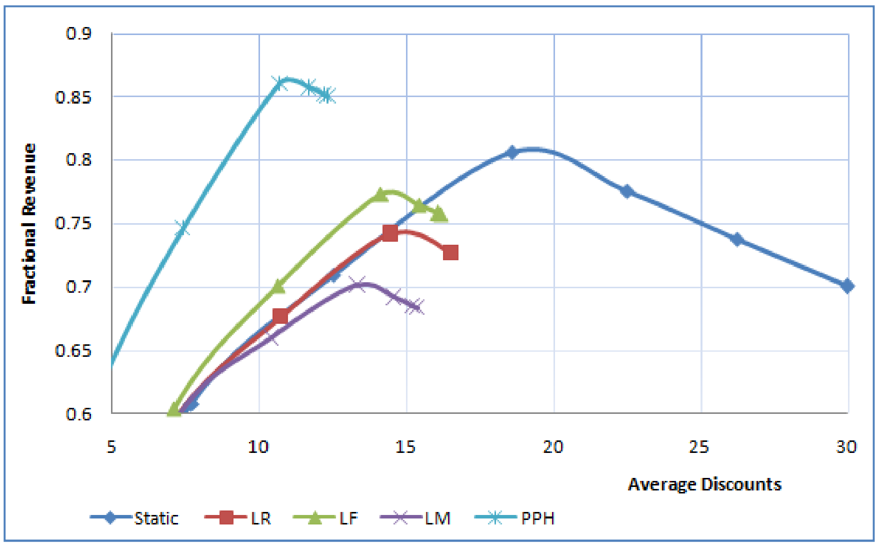

Figure 4.3.b : Fractional revenue versus average discounts (up to 30 % discount)

Each method has its own peak point, which corresponds to optimum revenue achievement. Before the peak, extra revenue generated is more than discounts offered to customers, and after the peak, discounts offered to customers are more than extra revenue generated.

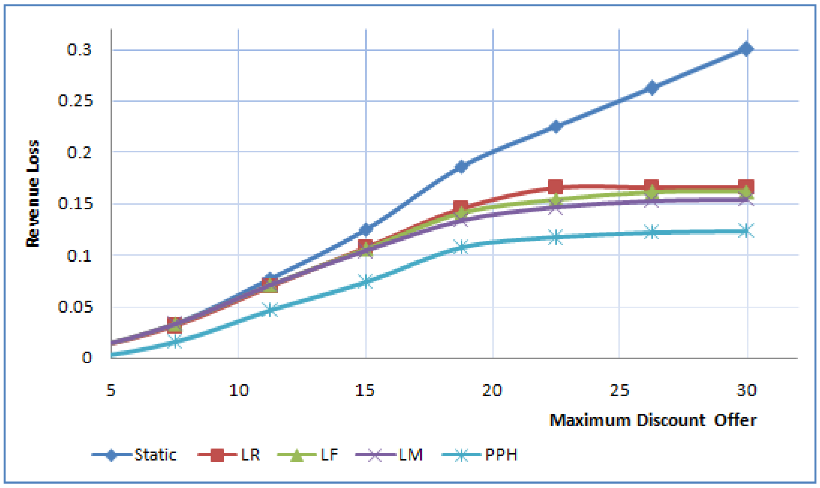

Figure 4.4: Revenue Loss versus discount offer

The proposed method achieves more than 85% revenue and passes nearly 10% average discounts to customers. Similarly, around 75% of the revenue is generated with loyalty-based methods, with around 15% of average discounts passed on to customers. The static method achieves around 80% revenue with 20% of discounts passed to customers. A similar conclusion can be seen in Figure 4.4, with maximum discount offer versus revenue loss for different methods.

As in Figure 4.5, ANOVA results depict the statistical significance of different discount levels and different methods, with an acceptable R^2 value. No significant interaction effect was observed on the model. We further ran a Tukey simultaneous test for method-by-method comparison, as seen in Figure 4.6. From these test results, the proposed method, PPH, is confirmed as significant method to every other method with high statistical significance.

```
General Linear Model: Revenue versus Discount, Method

Factor    Type   Levels  Values
Discount  fixed       9  1, 2, 3, 4, 5, 6, 7, 8, 9
Method    fixed       5  1, 2, 3, 4, 5

Analysis of Variance for Revenue, using Adjusted SS for Tests

Source            DF   Seq SS   Adj SS   Adj MS       F      P
Discount           8  6.34589  6.34589  0.79324  310.75  0.000
Method             4  0.11870  0.11870  0.02968   11.63  0.000
Discount*Method   32  0.08352  0.08352  0.00261    1.02  0.451
Error             90  0.22974  0.22974  0.00255
Total            134  6.77785

S = 0.0505236   R-Sq = 96.61%   R-Sq(adj) = 94.95%
```

Figure 4.5 : ANOVA test results

```
Tukey Simultaneous Tests
Response Variable Revenue
All Pairwise Comparisons among Levels of Method
Method = 1  subtracted from:

        Difference       SE of            Adjusted
Method    of Means  Difference  T-Value    P-Value
2         -0.00673     0.01375   -0.489     0.9882
3          0.00846     0.01375    0.615     0.9724
4         -0.02774     0.01375   -2.017     0.2662
5          0.06131     0.01375    4.459     0.0002

Method = 2  subtracted from:

        Difference       SE of            Adjusted
Method    of Means  Difference  T-Value    P-Value
3          0.01518     0.01375    1.104     0.8039
4         -0.02101     0.01375   -1.528     0.5471
5          0.06804     0.01375    4.948     0.0000

Method = 3  subtracted from:

        Difference       SE of            Adjusted
Method    of Means  Difference  T-Value    P-Value
4         -0.03620     0.01375   -2.632     0.0730
5          0.05286     0.01375    3.844     0.0021

Method = 4  subtracted from:

        Difference       SE of            Adjusted
Method    of Means  Difference  T-Value    P-Value
5          0.08905     0.01375    6.476     0.0000
```

Figure 4.6: Tukey comparison among different methods

In order to test the statistical significance of the proposed method, we conducted an ANOVA test. Three replicates of random experiment results are recorded keeping all parameters at default, discounts levels ranging from [0, 30] and equal 8 incremental steps of 3.75 % , and all method (Static, LR, LF, LM, PPH) level.

We also investigated the effect of varying transaction rates (number of transactions per period) ranging from 1000 to 9000, at two levels of discount with all other parameters at default values for revenue generation performance. As depicted on Figure 4.7, at 22.5% of discount offer, PPH presents the best performance. LR, LF, and Static methods present a moderate performance, while the LM method presents an inferior performance. Similarly, at 30%, PPH retains the superior performance, and LR and LF are still the second best. At 30%, the Static method performance deteriorates and presents an inferior performance similar to LM. Increasing the number of transactions actually adds to the complexity; however, the augmented history available was exploited for better learning by the PPH, LR, and LF methods. Due to the lack of any model components on the Static method, the performance is always flat and degrading with higher discounts. On the other hand, the PPH method is best suited for the retail environment where the number of transactions is high and the e-retailer can announce the higher percentages of discounts as maximum offers for customers.

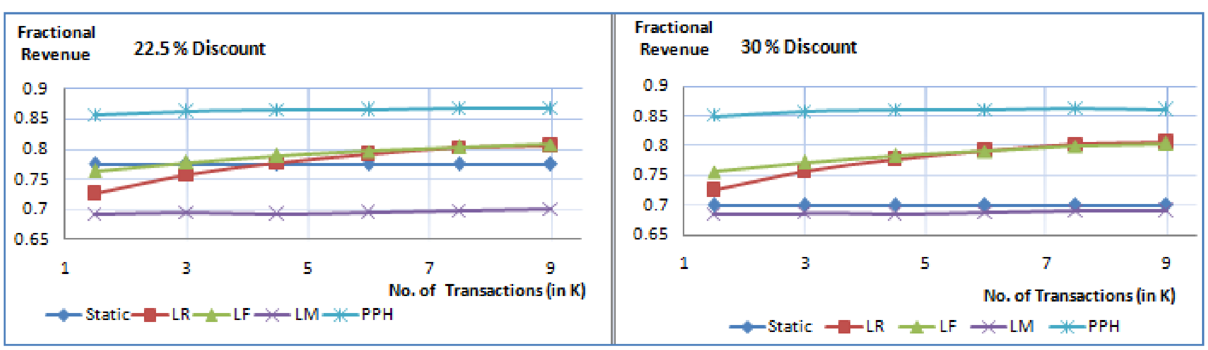

Figure 4.7 : Fractional revenue versus transaction rate at different discount levels

Another set of runs were evaluated to study the effect of a varying number of products from 1000 to 6000 SKUs with other fixed parameters at a default value. The wide performance gaps within the proposed method, PPH and the other methods are distinctly apparent as shown in Figure 4.8.

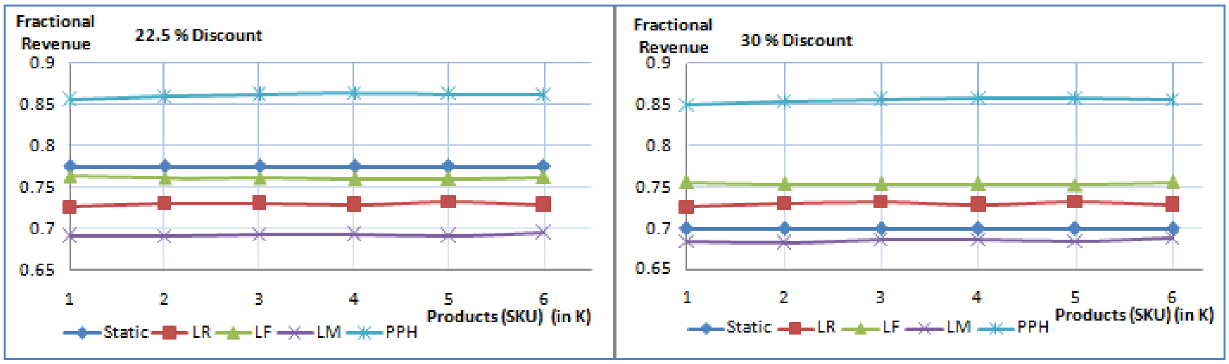

Figure 4.8: Fractional revenue versus no. of products at different discount levels

The Static method observes the degradation of performance at a higher discount level, while other methods are as affected either by more products or by more discount offers. In spite of increased complexity and sparsity with more products, the proposed method, PPH, generates slightly more revenue with more products.

4.6 Conclusion

In this chapter, we have presented the effects of implementing dynamic pricing strategies with different levels of discounts for the e-retail environment. We have proposed a novel method, a price product hierarchy (PPH)-based collaborative filtering method, as a personalized bundle dynamic pricing model for the problem of dynamic cross-selling. In addition to personalization, the method also provides discrimination at the level of product and price hierarchy groups. Unlike other loyalty-based methods, it can be concluded that the proposed method provides the most robust performance.

CHAPTER 5

SUMMARY & CONCLUSION

5.1 Summary

Cross-selling in the form of static bundling is widely practiced in almost every sector of retail business. Similarly, recommender systems with the feature of personalization are already prevalent in business set ups like movie rental portals, where customers reveal their items of interest through ratings. Given the transactional histories available with binary data type records, the personalized recommendation of various products for a target customer becomes a one-class problem, where traditional collaborative filtering techniques are not effective. Although there have been a few studies on one-class collaborative filtering, there are still gaps in the related literature of personalized and dynamic cross-selling, which needs not only targeted product recommendation, but also optimized dynamic price setting.

In this dissertation research, we tackled both issues of dynamic cross-selling in e-retailing. The first one involved generating a list of follow-up products as a personalized recommendation for a particular customer once the transactional process for the first product is initiated. PLC and CRM variables, such as product launch, market saturation and customer recency measures, are integrated into the matrix factorization based on OCCF methods. The efficiency of the proposed methods were found to be significantly higher than other existing methods. The second issue involved finding the optimal discount amount to pass on to a customer as an incentive for such cross-selling. The unique pair formed by a particular customer and a particular product group is considered in setting the dynamic price of the bundle through an optimized discount offer. With the consideration of product and price-based hierarchical groups, every customer's interests in recommended products are evaluated first through the regular matrix factorization-based collaborative filtering method. Finally, the dynamic price with different level of discount adjustment of the product is presented to the target customer based on the same customer's level of interest. The proposed method provided significantly better results in comparison to other static and loyalty-based methods.

This dissertation research also developed a simulation platform to generate e-retailer transactional records. Unlike currently available simulators (e.g., IBM Quest simulator) with bare products (SKU) and customer information, the proposed framework also maintains customer segments, product hierarchies (product category, sub-category, and price range), product prices, and other related information. The framework even provides the product life cycle features required for temporal analysis and market experimentation.

5.2 Research Contribution

Following are the specific contributions of our study for the task of product recommendation:

i) This is the first study that systematically accounted for temporal information related to the product and customer in one-class collaborative filtering problems. Product-based product life cycle and market saturation information are used to form a product-based temporal index. Customer visit information is used to form a customer-based temporal index.

ii) A thorough comparative study on various transactional parameters is carried out through varying number of customers, products, transactions, periods, and discount levels.

iii) Proposed methodology realizes the optimal revenue of the e-retailer by optimally managing the discounts offered to the customers. In other words, maximum discounts remain the same as static or other loyalty methods, but the revenue realization is significantly higher than other methods.

iv) Loyalty-based CRM techniques (RFM), a de-facto standard of current practices, provides an only unidirectional differentiate among customers, but the proposed framework, PPH, provides a bi-directional differentiation among customer-product combinations.

v) The proposed framework is flexible enough to incorporate various systematic variables and managerial decisions as listed below:

   a) The model is flexible enough to incorporate managerial decisions; for example, some product segments should be offered with heavy discounts, which could simply be achieved by resetting the product launch period correspondingly.

   b) Similarly, we considered infinite inventory for our model experiments, but the discount adjustment with inventory consideration can be carried out through the proposed model just by resetting the price hierarchy of a particular product at a particular inventory level.

c) Though the dynamic price experimentation was reported with single-shot discounts, the model can handle multiple periodic discounts.

5.3 Research Extension

Although we have already obtained encouraging results, some directions remain in which we can extend the research work further. One of the proposed methods uses product-based temporal information with the combination of a product launch period and market saturation information. Temporal tracking of customer behavior with a purchased product list-based analysis will be more promising as it may reveal customer status and transitions such as single versus family, student versus employee, person with or without children, etc. Similarly, another method uses customer recency information in the model as a CRM-based temporal loyalty variable. Other hybrid methods with the addition of various other CRM variables and their combinations are worth further experimentation. Another possible extension could focus on content boosting techniques with customer demographics and product hierarchical information.

Currently, we are assigning the discount level based on individual customer level preference for a particular PPH group product. One of the possible extensions in dynamic price setting is to augment the formulation with customer segment information. This is similar to product hierarchy consideration, classifying customers into a number of segments first, followed by individual customer based differentiation, which will be more effective in fixing the dynamic price for customer product pairs. Another possible extension is related to consideration of the temporal aspects of the problem. This extension has two parts. The first one is to consider the varying temporal behavior of the customer into the model. The second one is to consider product life cycle-based modeling, which will be very useful for products with a relatively short-life like consumer electronics.

APPENDIX

Area under ROC curve (AUC)

In a binary decision problem, a classifier categorizes example instances as either positive or negative. The decision made by the classifier can be represented in a structure known as a confusion matrix or contingency table as shown in Figure A.1. This matrix forms the basis for many common metrics.

Actual

Classifier

	+ve	-ve
+ve	TP	FP
-ve	FN	TN

Figure A.1: 2x2 Contingency Table (Confusion Matrix)

There are four possible outcomes for a binary classifier. If the instance is positive in actual and it is classified as positive, it is counted as a true positive (TP); if it is classified as negative, it is counted as a false negative (FN). If the instance is negative in actual and it is classified as negative, it is counted as a true negative (TN); if it is classified as positive, it is counted as a false positive (FP). TP and TN are good results whereas FP and FN are erroneous results.

TP : True Positive : Good	TPR = TP/(TP + FN) {aka Sensitivity and Recall }
TN : True Negative : Good	TNR = TN/(TN + FP) {aka Specificity}
FP : False Positive : Error (Type I)	FPR = FP/(FP + TN) = 1 – TNR = 1 – Specificity
FN : False Negative : Error (Type II)	Precision = TP /(TP + FP)

Figure A.2: Common metrics

There are various measures related to classifier performance exist in literature. Few of these measures are accuracy, TPR (true positive rate), FPR (false positive rate), PPV (positive predictive value), NPV (negative predictive value), precision, recall, sensitivity, and specificity.

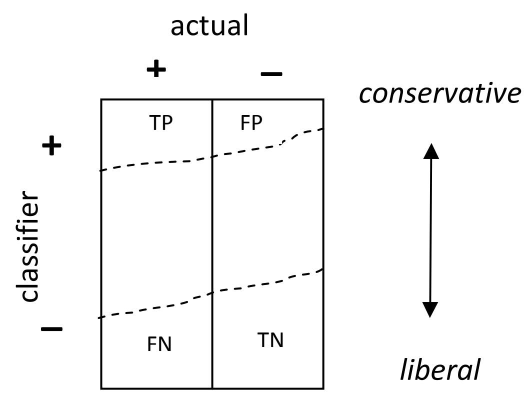

Figure A.3: Threshold effect on classifier

The threshold setting, the level of confidence in classifying an instance as a positive, is an important issue in any classification model. More '*conservative*' system (as demonstrated in Figure A.3) sets higher threshold and the classifier makes less error (FP) with a sacrifice on some of good instances (TP). In another extreme, the '*liberal*' system sets the lower threshold allowing all good instances (TP) but that comes only with more error (FP) acceptance. So, if one has to rank some classifiers by how good they are, the ranking might not remain same at different threshold values.

ROC (Receiver Operating Characteristic) curves plot TPR (also Sensitivity) versus FPR (also 1-Specificity) with the values ranging from 0 to 1 as shown on Figure A.4. In other words, the ROC curve provides the classifier performance plot ranging from *conservative* to the *liberal* thresholds.

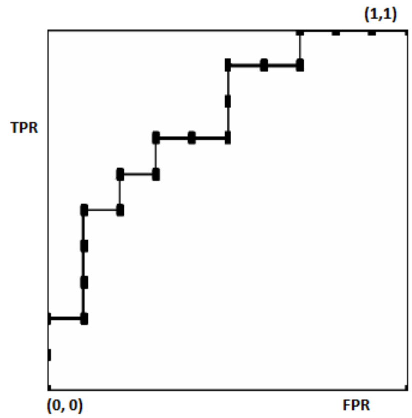

Figure A.4: Sample ROC curve plot

The area under the ROC curve (AUC) is one of the popular metrics that can be used to compare different classifier model performance in two-dimensional visualization space. In addition, AUC method provides a single scalar value that represents the overall expected performance of a classifier, which offers easier basis for comparing two or more classifier models. As the ROC curve is plotted within the area of the unit square, any classifier model that should perform better than random guess classifier (a diagonal from (0,0) to (1,1)) should have under the curve area (AUC) values in between 0.5 and 1.0. Compared classifier models are ranked based on the calculated AUC values.

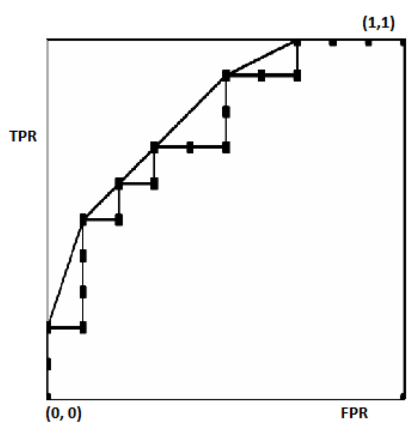

Figure A.5: Smoothing of a ROC curve plot

ROC plots as in Figure A.4 is reconstructed to smooth ROC curve as shown in Figure A.5 using a non-parametric method based on constructing trapezoids under the curve as an approximation of area.

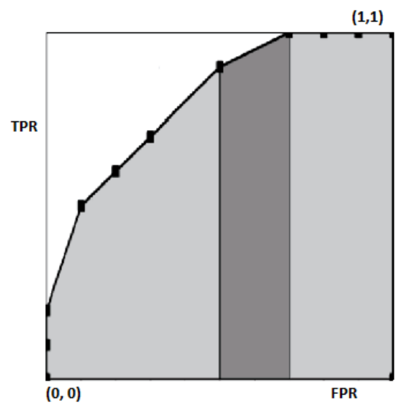

Figure A.6: Area under ROC curve

As shown in Figure A.6, one of the trapezoids, i^{th} trapezoid is shaded in dark, and the area is calculated as explained below:

Two X-axis points along FPR : F_{i-1} & F_i

Two Y-axis points along TPR : T_{i-1} & T_i

Then, $AUC_i = ½ * \{T_i + T_{i-1}\} * \{F_i - F_{i-1}\}$

Finally, sum of all the trapezoidal area provides the total AUC for a particular classifier model as shown in the Figure A.6 with shaded region.

REFERENCES

Adams,W.J., and J.L. Yellen,(1976).Commodity packaging and the burden of monopoly, Journal of Economics, 90, 475 - 498.

Adomavicius, G. and A. Tuzhilin, (2005). Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering,17 (6) 734-749.

Agrawal, R., R. Rakesh, T. Imielinski, and A. Swami,(1993). Mining association rules between sets of items in large databases, in: Proceedings of ACM SIGMOD22 (2) 207–216.

Agrawal, R., and R. Srikant, (1994). Fast Algorithms for Mining Association Rules, In Proc. 20th Int. Conference on Very Large Databases.

Anderson, C., (2006). The Long Tail: Why the Future of Business is Selling Less of More. New York, NY: Hyperion.

Ansari, A., S. Essegaier, and R. Kohli, (2000). Internet Recommendation Systems, Journal of Marketing Research, 37, 363-375.

Aviv, Y. and A. Pazgal, (2005). A partially observed Markov decision process for dynamic pricing, Management Science, 51, 1400-1416.

Aydin, G. and S. Ziya,(2008). Pricing promotional products under upselling, Manufacturing & Service Operations Management,10(3) 360-376.

Aydin,G. and S. Ziya,(2009).Personalized dynamic pricing of limited inventories, Operations Research, 57 (6) 1523–1531.

Beatty, S. E. and Ferrell M. E. (1998), Impulsive buying: modeling its precursors, Journal of Retailing, 74 (2) 169–191.

Bertsimas, D. J.,A. J. Mersereau, and N.R. Patel, (2003). Dynamic classification of online customers, 3rd SIAM Conference on Data Mining, 107- 118.

Bolton, R. N. and C. O.Tarasi; Malhotra, N. K.(ed.), (2006). Managing Customer Relationships, Review of Marketing Research, 3, 3-38.

Breese, J., D. Heckerman, and C. Kadie, (1998). Empirical analysis of predictive algorithms for collaborative filtering, in: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 43-52.

Bitran, G., and R. Caldentey,(2003). An overview of pricing models for revenue management.Manufacturing Service Operations Management,5, 203 - 229.

Brijs, T.,B. Goethals, G. Swinnen, K. Vanhoof, and G. Wets,(2000). A data mining framework for optimal product selection in retail supermarket data: the generalized profset model,7th ACM*SIGKDD* Conference.

Brin, S., R., Motwani, and C. Silverstein,(1998). Beyond market baskets: Generalizing association rules to dependence rules, Data Mining and Knowledge Discovery, 2(1) 39–68.

Bult, J. R. and T. Wansbeek,(1995).Optimal selection for direct mail, Marketing Science, 14 (4) 378-394.

Bulut, Z., U. Gurler, and A. Sen,(2000). Bundle pricing of inventories with stochastic demand, European Journal of Operational Research. 197, 897-911.

Burke, R.,(2002). Hybrid Recommender Systems: Survey and Experiments, User Modeling and User-Adapted Interaction,12, 331-370.

Chen,L. S.,F. Hsu, M. Chen, and Y. Hsu,(2008).Developing recommender systems with the consideration of product profitability for sellers, Information Sciences,178, 1032–1048.

Dasseni, E., V. Verykios, A. K. Elmagarmid, and E. Bertino, (2001). Hiding association rules by using confidence and support. In Proceedings of the 4th International Information Hiding Workshop, pp. 369-383.

Elmaghraby, W. and P. Keskinocak, (2003). Dynamic pricing in the presence of inventory considerations: research overview, current practices, and future directions, Management Science, 49 (10) 1287-1309.

Eppen, G. D., W. A. Hanson, and R. K.Martin, (1991). Bundling: new products, new markets, low risk, Sloan Management Review, 32 (4), 7-14.

Estelami, H.,(1999). Consumer savings in complementary product bundles, Journal of Marketing , Theory and Practice, 7, 107-114.

Evfimievski, A., J. Gehrke, and R. Srikant.(2003), Limiting privacy breaches in privacy preserving data mining.In Proceedings of the 22nd Symposium on Principles of Database Systems, pp. 211-222.

Gallego, G. and G. V. Ryzin, (1994).Optimal dynamic pricing of inventories with stochastic demand over finite horizons Management Science,40 (8) 999 - 1020.

Goldberg, D., D. Nichols, B. M. Oki, and D. Terry, (1992). Using collaborative filtering to weave an information tapestry, Communications of ACM,35 (12)61–70.

Goldberg, K., T. Roeder, D. Gupta, and C. Perkins, (2001). Eigentaste: a constant time collaborative filtering algorithm, Information Retrieval, 4 (2) 133–151.

Hanson, W. and K. Martin, (1990).Optimal bundle pricing, Management Science, 36 (2) 155-174.

Harlam, B., A. Krishna, D. Lehmann, and C. Mela, (1995).The impact of bundle type price framing and familiarity on evaluation of the bundle, Journal of Business Research, 33, 57-66.

Hausman, A., (2000). A multi-method investigation of consumer motivations in impulse buying behavior, Journal of Consumer Marketing, 17,(5), 403 – 426.

Heckerman, D., D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie, (2001). Dependency networks for inference, collaborative filtering, and data visualization, Journal of Machine Learning Research 1(1)49–75.

Herlocker, J., J.Konstan, A. Borchers, and J. Riedl,(1999). An algorithmic framework for performing collaborative filtering, in: Proceedings of ACM Conference on Research and Development in Information Retrieval.

Hinterhuber, A., (2008).Customer value-based pricing strategies: why companies resist, Journal of Business Strategy, 29 (4), 41-50.

Hofmann, T., (2004). Latent semantic models for collaborative filtering, ACM Transactions on Information Systems 22 (1)89–115.

Ingenbleek, P., M. Debruyne, R. Frambach, and T. Verhallen,(2003), Successful new product pricing practices: a contingency approach, Marketing Letters, 14 (4), 289-305.

Keeney,R. L. (1999). The value of internet commerce to the customer, Management Science 45 (4) 533-554.

Kim, J. K., Y. Cho, W. Kim, J. Kim and J. Suh,(2002).A personalized recommendation procedure for Internet shopping support, Electronic Commerce Research and Applications,1, 301–313.

Kohavi, R. and R. Parekh, (2004).Visualizing RFM Segmentation, SIAM International Conference on Data Mining, 391- 399.

Koren, Y., R. Bell, and C. Volinsky, (2009). Matrix factorization techniques for recommender systems, IEEE Computer 42 (8) 30–37.

Koren, Y., (2009). Collaborative filtering with temporal dynamics, in: Proceedings of 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 447-456.

Lawless, M. W., (1991). Commodity bundling for competitive advantage: strategic implications. Journal of Management Studies, 28 (3) 267-279.

Lee, D. D.andH. S. Seung, (1999). Learning the parts of objects by non-negative matrix factorization, Nature, 401 (21)788–791.

Lee,D. D. and H. S. Seung, (2000).Algorithms for Non-Negative Matrix Factorization, in: Proceedings of 13th NIPS Conference, Advances in Neural Information Processing Systems, 2000, pp. 556-562.

Li, Y., P. Ning, X. S. Wang, and S. Jajodia,(2001). Generating Market Basket Data with Temporal Information, KDD Workshop on Temporal Data Mining.

Linden, G.,B. Smith, and J. York, (2003). Amazon.com recommendations item-to-item collaborative filtering, IEEE Internet Computing 7(1) 76-80.

Liu, D., C. Lie, and W. Lee, (2009). Hybrid of sequential rules and collaborative filtering for product recommendation, Information Sciences, 179, 3505-3519.

Lu, Z., D. Agarwal, and I. S. Dhillon, (2009). A spatio-temporal approach to collaborative filtering, In :Proceedings of 3rd ACM RecSys Workshop on Recommendation-based Industrial Applications.

McAfee, R. P. and Vera teVelde,(2007). Dynamic Pricing in the Airline Industry, Handbook on Economics and Information Systems, Elsevier Handbooks in Information Systems, 1.

McCardle, K. F., K. Rajaram, and C. S. Tang, (2007). Bundling retail products: models and analysis, European Journal of Operational Research,177, 1197-1217.

McGill, J. I. and G. Van Ryzin,(1999). Revenue Management: Research overview and prospects,Transportation Science,33, 233 - 256.

Miyahara, K., and M. J. Pazzani, (2000). Collaborative filtering with the simple Bayesian classifier, in: Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence, pp. 679–689.

Moe, W. W. and P. S. Fader,(2004). Dynamic conversion behavior at e-commerce sites. Management Science, 50, 326 - 335.

Montaner, M., B. López, and J.L. de la Rosa, (2003).A taxonomy of recommender agents on the internet, Artificial Intelligence Review,19 (4) pp. 285–330.

Moriarty, R. T., and T. Kosnik, (1989). High-tech marketing: concepts, continuity, and change. Sloan Management Review, 30 (4), 7-17.

Netessine, S., S. Savin, and W. Xiao, (2006).Revenue management through dynamic cross-selling in E-commerce retailing, Operations Research,54, 893-913.

Oliveira S. and O. Zaiane, (2003). Protecting sensitive knowledge by data sanitization, In Proceedings of the 3rd IEEE ICDM, pp. 211-218.

Ovans, A. (1997). Make a bundle bundling, Harvard Business Review, (Nov-Dec), 18-20.

Paatero, P. and U. Tapper, (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values, Environmetrics 5 (2) 111–126.

Pan, R., Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M Scholz, and Q.Yang, (2008). One-Class Collaborative Filtering, in: Proceedings of 8th IEEE International Conference on Data Mining, pp. 502-511.

Pan, R. and M. Scholz, (2009). Mind the gaps: Weighting the unknown in large-scale one- class collaborative.

Rao, V.R.(1993). Pricing models in marketing, Handbooks in Operations Research and Management Science: Marketing, 517 – 552.

Resnick, P.and H. Varian,(1997). Recommender Systems, Communications of the ACM, 40 (3), 56-58.

Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, (1994). Grouplens: an open architecture for collaborative filtering of netnews, in: Proceedings of the ACM Conference on Computer Supported Cooperative Work, pp. 175–186.

Salinger, M. A., (1995). A graphical analysis of bundling, The Journal of Business, 68 (1) 85-98.

Sarwar, B., G. Karypis, J. Konstan and J. Riedl, (2000). Analysis of recommendation algorithms for e-commerce, in: Proceedings of 2nd ACM conference on E-Commerce,pp. 158–167.

Sarwar, B. M., G.Karypis, J. A. Konstan, and J. Riedl, (2000). Application of dimensionality reduction in recommender system - a case study, in:Proceedings of ACM WebKDD Workshop.

Sarwar,B., G. Karypis, J. Konstan and J. Riedl,(2001).Item-based collaborative filtering recommendation algorithm, In: Proceedings of 10th International World Wide Web Conference, pp.285–295.

Sarwar, B. M.,G.Karypis, J. A. Konstan, and J. Riedl, (2002). Incremental singular value decomposition algorithms for higly scalable recommender systems.in: Proceedings of 5th International Conference on Computer and Information Science, pp. 27-28.

Schmalensee, R., (1984). Gaussian demand and commodity bundling, Journal of Business,57(1) S211 –S230.

Shani, G., D. Heckerman, and R. I. Brafman, (2005).An MDP-based recommender system, Journal of Machine Learning Research, 6, 1265–1295.

Sindhwani,V., S.S. Bucak, J. Hu, and A. Mojsilovic,(2009). A family of non-negative matrix factorizations for one-class collaborative filtering problems, In : Proceedings of 3rd ACM RecSys Workshop on Recommendation-based Industrial Applications.

Skillicorn, D., (2007). Understanding complex datasets – data mining with matrix decomposition, Chapman & Hall/CRC Taylor & Francis Group.

Srebro, N. and T. Jakkola, (2003).Weighted low rank approximations, In Proceedings of 20th International Conference on Machine Learning, pp. 720-727.

Stigler,G. J.,(1963). United States v. Loew's Inc.: a note on block-booking, The Supreme Court Review 1963,152-157.

Streitfeld*,* D.,(2000).On the Web, Price Tags Blur What You Pay Could Depend on Who You Are, Wednesday, September 27, 2000; Page A01; Washington Post.http://www.washingtonpost.com/ac2/wp-dyn/A15159-2000Sep25 .

Su, X. and T.M. Khoshgoftaar,(2009). A survey of collaborative filtering techniques, Advances in Artificial Intelligence, 2009, 1-19.

Turow, J., L. Feldman, K. Meltzer,(2005). Open to exploitation: American shoppers online and offline, A Report from the Annenberg Public Policy Center of the University of Pennsylvania

Ungar,L. H. and D. P. Foster, (1998). Clustering methods for collaborative filtering, in Proceedings of AAAI Workshop on Recommendation Systems, (1) ,pp. 1-16.

Venkatesh, R. and V. Mahajan,(1993).A probabilistic approach to pricing a bundle of products or services, Journal of Marketing Research, 30 (4)494-508.

Wang, K. and M. T. Su,(2002). Item selection by "Hub-Authority" profit ranking,In: Proceedings of 8thACM *SIGKDD* Conference, 652 – 657.

Weiss, R. M. and A. K. Mehrotra, (2001). Online Dynamic Pricing: Efficiency, Equity and the Future of E-commerce, Va. J.L. & Tech. 11(6) Virginia Journal of Law and Technology Association.

Wu, X., Y. Wu, Y. Wang, and Y. Li, (2005).Privacy-Aware Market Basket Data Set Generation: A Feasible Approach for Inverse Frequent Set Mining, SIAM Conference on Data Mining, pp. 103-114.

Xian,L. and Q. Yang, (2009).Time-dependent models in collaborative filtering based recommender system, in:Proceedings of2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, (1), pp. 450-457.

Xiong, L., X. Chen, T. Huang, J. Schneider, J. G. Carbonell, (2010).Temporal collaborative filtering with Bayesian probabilistic tensor factorization, in Proceedings of SIAM International Conference on Data Mining.

Yadav, M. S. and K. B. Monroe, (1993). How buyers perceive savings in a bundle price: an examination of a bundle’s transaction value, Journal of Marketing Research, 30 (3) 350-358.

ABSTRACT

A FRAMEWORK FOR PERSONALIZED DYNAMIC CROSS-SELLING IN E-COMMERCE RETAILING

by

ARUN KUMAR TIMALSINA

May 2012

Advisor: Dr. Ratna Babu Chinnam

Major: Industrial Engineering

Degree: Doctor of Philosophy

Cross-selling and product bundling are prevalent strategies in the retail sector. Instead of static bundling offers, i.e. giving the same offer to everyone, personalized dynamic cross-selling generates targeted bundle offers and can help maximize revenues and profits. In resolving the two basic problems of dynamic cross-selling, which involves selecting the right complementary products and optimizing the discount, the issue of computational complexity becomes central as the customer base and length of the product list grows. Traditional recommender systems are built upon simple collaborative filtering techniques, which exploit the informational cues gained from users in the form of product ratings and rating differences across users. The retail setting differs in that there are only records of transactions (in period X, customer Y purchased product Z). Instead of a range of explicit rating scores, transactions form binary datasets; 1-purchased and 0-not-purchased. This makes it a one-class collaborative filtering (OCCF) problem. Notwithstanding the existence of wider application domains of such an OCCF problem, very little work has been done in the retail setting. This

research addresses this gap by developing an effective framework for dynamic cross-selling for online retailing.

In the first part of the research, we propose an effective yet intuitive approach to integrate temporal information regarding a product's lifecycle (i.e., the non-stationary nature of the sales history) in the form of a weight component into latent-factor-based OCCF models, improving the quality of personalized product recommendations. To improve the scalability of large product catalogs with transaction sparsity typical in online retailing, the approach relies on product catalog hierarchy and segments (rather than individual SKUs) for collaborative filtering. In the second part of the work, we propose effective bundle discount policies, which estimate a specific customer's interest in potential cross-selling products (identified using the proposed OCCF methods) and calibrate the discount to strike an effective balance between the probability of the offer acceptance and the size of the discount. We also developed a highly effective simulation platform for generation of e-retailer transactions under various settings and test and validate the proposed methods.

To the best of our knowledge, this is the first study to address the topic of real-time personalized dynamic cross-selling with discounting. The proposed techniques are applicable to cross-selling, up-selling, and personalized and targeted selling within the e-retail business domain. Through extensive analysis of various market scenario setups, we also provide a number of managerial insights on the performance of cross-selling strategies.

Keywords: cross-selling, bundling, data mining, dynamic pricing, one-class collaborative filtering, factorization, simulator.

AUTOBIOGRAPHICAL STATEMENT

NAME: ARUN KUMAR TIMALSINA

EDUCATION

Ph.D. : Industrial Engineering, Wayne State University, Detroit, MI, 2012.
M.S. : Industrial Engineering, Wayne State University, Detroit, MI, 2010.
M.E. : Computer Science, Asian Institute of Technology, Bangkok, Thailand, 1999.
B. E. : Electronics Engineering, NED University of Engineering & Technology, Karachi, Pakistan,1994.

WORK EXPERIENCE

09/06 – 12/11	Part Time Faculty / Teaching Assistant, Industrial & Systems Engineering, WSU, Detroit, MI
04/01 – 08/06	Lecturer, ECE Department, IOE, Tribhuvan University, Kathmandu, Nepal
01/00 – 03/01	Systems Analyst, AASU, Regional Computer Center, AIT, Bangkok, Thailand
07/96 – 06/97	Systems Engineer, ComputerLand Communication Systems, Kathmandu, Nepal
07/94 – 06/96	Senior Teletext Administrator Wavetec Pvt. Ltd, Karachi, Pakistan

AWARDS

- Outstanding Teaching Assistant Award, College of Engineering, WSU, 2010
- Honorable Mention Award: 60th Anniversary History Competition CHC$_{60}$, IEEE-CS, 2006
- Honorable Mention Award: 5th International Design Competition CSIDC-2004, IEEE-CS, 2004
- GTA Award, WSU(PhD study); NORAD Scholarship Award, AIT (ME study); PTA Scholarship Award (BE study)

PROFESSIONAL MEMBERSHIP

- Professional Member – Association for Computing Machinery (ACM)
- Member - Institute of Electrical and Electronics Engineers (IEEE) and Computer Society

PUBLICATIONS

I am an author/co-author of about dozen publications. They include journal publications and publications in peer reviewed conference proceedings. This dissertation work contributed couple of presentations and currently three papers are ready for submission.