FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Service Recommendation

MASTER'S THESIS

In partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Business Informatics

by

ILKER BALTACI
Number 0526856

at
Faculty of Informatics at Vienna University of Technology

Approved:
Supervision: ao. Prof. Dr. Jürgen Dorn

Vienna, 24.06.2013

(Signature Student) (Signature Supervisor)

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Empfehlungsdienste für Dienstleistungen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

ILKER BALTACI
Matrikelnummer 0526856

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuer : ao. Prof. Dr. Jürgen Dorn

Wien, 24.06.2013 _______________________ _______________________
(Unterschrift Verfasser) (Unterschrift Betreuer)

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: İlker Baltacı

Signature:

Service Recommendation

İlker Baltacı

Abstract

The problem of information overload has been coped with through the use of recommendation systems in information retrieval for ages. During the era of Web 2.0 and e-commerce, recommendation systems have become more than information filtering and relevance matching tools for users and service providers. They are considered as powerful business and marketing tools from the perspective of management personnel and sophisticated sales assistance for online customers. Research in recommendation system literature proposes different recommendation techniques and similarity measures for various domains including textual documents, consumer goods and services. Service science clarifies service features, quality evaluation models and involvement of service customer and service employee, which makes services more than just immaterial goods. Due to unique characteristics of services, service recommendation requires explicit consideration of certain service features and dimensions that do not exist for goods. This study investigates services with their unique characteristics that imply explicit considerations during implementation of service-oriented recommendation systems.

Additionally, possible problems that service customers could face during purchase due to intangibility, inseparability, perishability and heterogeneity of services could be overcome with the help of recommendation systems. This study also investigates the additional roles of recommendation systems for service customers in connection to service properties denoted in service science.

The empirical part of the thesis proposes a generic recommendation model by considering service customer and system users' requirements. The generic service model is evaluated in the restaurant domain in the scope of an offline experiment with given user, restaurant and rating data to measure recommendation accuracy.

Keywords: recommendation systems, services, multi-dimensional ratings, service quality models, restaurant recommendation, content-based recommendation, collaborative filtering.

Empfehlungsdienste für Dienstleistungen

İlker Baltacı

Abstract

Das Informationsüberflutungsproblem im Informationsrückgewinnungsfachgebiet wird seit langem durch den Einsatz von Empfehlungssystemen überwindet. In der Epoche von Web 2.0 und nach dem Aufstieg von E-Kommerz, erhielten die Empfehlungssysteme zusätzliche Aufgaben außer dem Informationsfiltern und Relevanz-Rechnung. Sie werden als ein wichtiges Management- und Marketinginstrument betrachtet, die nicht nur sowohl für die Online-Kunden als auch für Dienstleister sehr viele Vorteile bieten können. Ein effizientes Empfehlungssystem kann fast alle Aufgaben von einem Verkaufsberater erfüllen. Die Literatur über die Empfehlungssysteme beschriebt unterschiedliche Empfehlungsalgorithmen und Ähnlichkeitsmaßnahmen für viele Einsatzgebiete wie Text-Dokumente, Konsumgüter und Dienstleistungen. Im folge der einzigartigen Eigenschaften von Dienstleistungen wie die Teilnahme der Kunden und der Dienstleistungsmitarbeiter an dem Service-Prozess sowohl die unterschiedlichen Dienstleistungsqualitätsabmessungsmodelle gelten die Services mehr als immaterielle Güter. Bei der Implementierung eines dienstleistungsorientierten Empfehlungssystems sind die speziellen Eigenschaften und Dimensionen von Dienstleistungen explizit zu berücksichtigen. Diese Masterarbeit befasst sich mit den generischen Dimensionen der Dienstleistungen, die sich bei den herkömmlichen gut-orientierten Empfehlungssystemen nicht befinden.

Potentielle Probleme der Dienstleistungskunden während des Kaufprozesses können mit Hilfe von Empfehlungssystemen beseitigt werden, die auf Grund der immateriellen, untrennbaren, verderblichen und variablen Natur von Dienstleistungen auftreten können. Diese Arbeit beschäftigt sich mit die zusätzlichen Rollen der Empfehlungssysteme für Dienstleistungskunden im Bezug auf die sogenannten Eigenschaften von Dienstleistungen.

Der empirische Teil der Masterarbeit beschreibt ein generisches Empfehlungsmodel, das gezielt auf die Anforderungen von Dienstleistungskunden und Empfehlungssystembenutzern implementiert wird. Das generische Modell wird in einer Restaurantempfehlung-Anwendung eingesetzt und mit Hilfe von einem vordefinierten Daten-Set evaluiert.

Schlüsselwörter: Empfehlungssystem, Dienstleistung, multi-dimensionale Bewertung, Qualitätsmodel für Dienstleistungen, Restaurantempfehlung, inhaltsbasierte Empfehlung, kollaboratives Filtern.

ACKNOWLEDGMENT

First of all, I would like to thank to Professor Jürgen Dorn, my supervisor at University of Technology in E-Commerce institute. Without your assistance and feedback, I could not have finished this master thesis as the final part of my studies. Thank you for your valuable time and all your advice that enlightened me during my research.

I would like to send my special thanks to my lovely parents who have supported me during my whole education life, beginning from the primary school. Without your support and affection it would not have been possible to study in Austria.

Additionally, I would like to thank my girlfriend, Ece Bellek, for her patience and understanding throughout my whole master studies. You have always kept me motivated and encouraged to finish this thesis through difficult times.

Table of Content

TABLE OF FIGURES

TABLE OF ABBREVIATIONS

CBRS	Content based recommender system
CF	Collaborative filtering
IR	Information retrieval
KBRS	Knowledge based recommender system
K-NN	k-Nearest Neighbor
MAE	Mean absolute error
NDPM	Normalized distance based preference measure
PCA	Principal component analysis
RMSE	Root mean squared error
RS	Recommender System
SVD	Singular value decomposition
UBRS	Utility based recommender system

1 Introduction

1.1 Background

Services are considered as key activities of advanced economies and the service sector in developing economies depends mainly on financial, accommodation, retail, health, and education, and information services. Apart from that, tangible products of manufacturers are also offered as a bundle with complementary supporting services. As a consequence, the strict line between pure services and goods has been replaced by a service-product continuum in modern economies.

Services are provided in different domains, and all economic activities that do not fall under agriculture or industry can be assigned to the tertiary sector. Advances in telecommunication and information technologies have created new service models, and today services are provided as complex economic activities depending on the service domain. According to World Economic Outlook Database of IMF[1], the nominal GDP[2] composition of three sectors in 2012 can be seen in Table 1-1 : The nominal GDP composition of three sectors.

Agriculture	***Industry***	***Services***	**Total**
5,9 %	30,5 %	63,6 %	100 %
4,230,731 $	21,870,727 $	45,605,844 $	71,707,302 $

Table 1-1 : The nominal GDP composition of three sectors (in millions of dollars)

By the 1990s, the Internet had expanded exponentially and become an important medium for commerce that overcame physical limitations. Today, e-commerce sites bring retail goods and various services together with their potential customers. Service providers offer personalization of their e-shops with the help of recommendation systems. Additionally, recommender systems also overcome the problem of information overload for e-shop visitors. Before the rise of e-commerce, recommendation systems were applied mainly as information retrieval and information filtering tools for textual documents. The main purpose of information filtering tools was selection of the most relevant documents according to user interests constrained by preferences or given key words. (Yao, 1995). Due to advances in information technology and increased service customer requirements, today's recommendation systems have more capabilities. They can mimic the major functions of sales assistants, as well as tourism agencies that organize holiday packages. Today, recommendation systems are still an important field of research, and new trends in the Internet as well as advances in information technology provide many new aspects and possible improvements for recommendation systems (Ricci, 2010).

[1] International Monetary Fund
[2] Gross domestic product

1.2 Problem Definition

Services have certain unique characteristics—including intangibility, inseparability, perishability, and heterogeneity—that distinguish them from goods in terms of marketing and quality evaluation. Services as recommendation items also show various differences to physical retail goods. These differences require explicit consideration of the unique service characteristics and dimensions during recommendation model design.

One of the most important differences between goods and services is heterogeneity, which corresponds to variability of service entities. Two services are different in nature, even though they share the same domain and attributes. Perception of service value varies by all individual customers since all service customers have unique needs and can experience different impressions during interactions with a service provider. In contrast to buyers of goods, service customers are involved in service delivery and their performance could affect the service value. Due to that, quality of service depends on many parameters, and individual customer preferences should be considered separately (Bhasin, 2010).

Apart from that, recommendation systems model goods with their tangible attributes, which can be used as input parameters in the utility function. In contrast, services are intangible and the physical evidence of the supplementary physical components can be irrelevant depending on the service domain. Inseparability of services makes quality evaluation a long-term process with multiple service touch points that can be spread to multiple service dimensions. As a consequence, overall service quality needs to be evaluated in multi-criteria ratings to cover distinct dimensions separately. A good can also be evaluated by considering multiple physical features and technical specifications as rating dimensions. In contrast to services, ratings do not need to cover service employee performance or customer involvement.

There are many notable recommendation systems in service domains, including accommodation, travel, restaurant, and food delivery. These service domains have many common dimensions. However, each service domain also has unique properties and requirements for service customers. These service customer requirements assign additional roles for service recommendation systems. Available recommendation techniques that consider physical goods as recommendation items need further considerations and extensions in order to cover individual service dimensions and service customer demands. Therefore, an accurate and trustful service recommendation system has additional technical and functional requirements.

1.3 Purposes of study

The hypothesis of this master thesis is that good oriented and service oriented recommendation approaches have various differences in modeling and operational levels. A service recommendation model requires additional considerations about rating dimensions and recommendation generation. Available recommendation techniques for goods need further extension or adaptation to unique service characteristics.

The goal of this thesis is the reasoning of common service dimensions and properties and the definition of a generic service recommendation model to cover these service dimensions and fulfill service customer requirements.

1.4 Research Questions

What are the unique service features that require explicit consideration for service recommendation?

1.5 Structure of the Thesis

Chapter 2 is about the state of the art and concentrates on services with their definition, classification, characteristics, and marketing concepts as differing from products. Since services are customer centric in contrast to goods, the role and involvement of customers during service delivery is investigated. Additionally, this section refers to various important service quality models and the impact of customer expectations on service quality perception.

The second part of this chapter introduces major recommendation techniques, with their various advantages and drawbacks. For the empirical part, relevant data mining techniques including similarity measures and data classification approaches are also briefly introduced.

The last part of chapter 2 is the state of the art in service recommendation. This section covers the role of tangibles and customer involvement in service recommendation. Additionally, the role of service recommendation systems for service customers is also emphasized.

Chapter 3 introduces requirements analysis of a generic service recommendation model by considering possible recommendation techniques, rating dimensions and user and item entities. Based on the findings of requirements analysis, a recommendation model is described in detail, including its user preference model, generic user rating dimensions, and prediction algorithm.

Chapter 4 introduces an application of the proposed model for the restaurant recommendation domain and investigates necessary extensions or adaptations of the generic model to cover domain-specific user requirements. Furthermore, data entity models including users and their preferences. Restaurants and restaurant ratings with corresponding weights are also discussed.

Chapter 5 covers an offline experiment to generate synthetic test data to evaluate the proposed model. The offline experiment simulates a concrete recommendation engine with given restaurant and user data and prior known user ratings. This chapter describes data generation rules to generate a desired number of restaurants that are grouped into certain categories, as well as different users belonging to given stereotypes with varying preferences and priorities.

Chapter 6 evaluates the proposed model considering recommendation accuracy, and discusses its drawbacks and strengths in various evaluation dimensions.

Chapter 7 proposes possible extensions and enhancement of the proposed model for future work, considering advances and trends in information technology as well as recommendation techniques.

Chapter 8 concludes this thesis.

2 State of the Art

2.1 Services

This section reasons the major differences between services and goods, which spread to multiple dimensions and need consideration from the perspective of service recommendation model design.

2.1.1 Definition

The first research on services considered them as "immaterial goods," and the main focus of this research was clarifying what services were not (rather than considering services unique and concentrating on their unique characteristics). Chesbrough and Spohrer (2006) state that any economic activity that did not fall into agriculture or manufacturing was considered to be a service during the era of industrialization and agriculture.

By the 1950s, services had expanded and started to become the dominating sector in advanced economies. Economists have agreed that services have their own characteristics rather than being just intangible goods. A modern perspective for service definition has been introduced with the *benefits without ownership* concept. This assumes that customers pay for perceiving an economic value, mostly for a limited period of time, and without acquiring ownership of this entity. Lovelock and Wirtz (2004) describe services as *"Any act, performance or experience that one party can offer another; one that is essentially intangible, and does not result in the ownership of anything. Its production may or may not be tied to a physical product."*

Hill (1977) suggests the following definition: *"A service is a change in the condition of a person or a good, belonging to some economic unit, which is brought about as the result of the activity of some other economic unit with the prior agreement of the former person or economic unit."* In contrast, goods are described as *"physical objects, which are appropriable and therefore transferable between economic units."*

Grönross (1998) proposes differences between goods and services considering the consumer's role and his interactions with the service provider. *"Physical goods are reproduced in a factory, whereas services are produced in a process in which consumers interact with the production resources of the service firm. Some part of the service may be prepared before the customers enter the process, but for service quality perception the crucial part of the service process takes place in interaction with customers and in their presence."*

A service needs to be the subject of an economic transaction. A transaction is an economic activity in that an actor processes a predefined task for another actor.

Flat cleaning services, car repair, driving lessons, surgery in a hospital, or massage in a spa center could be shown as typical examples of services. The mentioned services belong to different service domains, but in each case one unit performs an activity, which changes the physical or mental condition of the perceiving actor or a commodity. The performed activity has an output that provides certain benefits to the perceiving actor. At the end of the transaction, the service-demanding actor does not acquire any ownership, and only perceives promised benefits of the service.

2.1.2 Characteristics of Services (IHIP)

Services have various unique characteristics that make them different from goods. Economists have agreed on four, which are today abbreviated as IHIB[3]. These characteristics are especially important from the perspective of service marketing, which requires additional considerations to classical marketing techniques (which is known as *extended marketing mix*).

Intangibility

The intangibility and inseparability of services was first introduced in Traité D'Econonie Politique (1803) by French economist and businessman Jean Baptiste Say (1767–1832). He uses the example of a physician who visits his patient to diagnosis his health problem. After diagnosis, the doctor prescribes a medicine that heals the patient, and leaves without depositing any physical product. *"The physician's advice has been exchanged for his fee…. The act of giving was its production, of hearing [by the invalid] its consumption, and the production and consumption were simultaneous. This is what I call an immaterial product"* (Hill, 1999).

One of the key characteristics of a service is its intangibility. In contrast to goods, services cannot be seen, touched, smelt, or tasted before their consumption. Since consumers pay for the benefit that they perceive at the end of a service transaction, value estimation of services is also different than that of goods. Most of the time, service customers consider supporting factors like reliability of the service provider, company image, and other customers' reviews to make their decision before a service request. The first impression about services is important since it shapes customer expectations, which could be seen as a kind of *satisfaction threshold* in quality evaluation.

Services are provided as a set of small activities, which are coupled with certain physical goods. Required tangible components show differences depending on the service domain, and each service has a different level of tangible dependency. While consulting is a relatively intangible service without many physical components, during a meal in a restaurant a service customer experiences many tangible components, including the consumed meal itself, table and seats, and accessories inside the restaurant. These supporting components could help customers to estimate service quality. Before the service delivery, service customers rely on a service provider's reputation and judge tangible components of the service.

Inseparability

Jean Baptiste describes services as *immaterial economic units* whose production and consumption cannot be separated. In manufacturing, production is an independent process separate from the product delivery. In fact, design, production, and delivery of a good could be considered as totally different steps in the product life cycle. Goods are produced in factories and transported to different wholesalers or any other middleware resellers before they reach their end customers. Various business partners could carry out the distribution of manufactured goods. In contrast to goods, the production, delivery, and consumption of services are coupled and simultaneous (Hill, 1999).

Apart from product life cycle, assurance of service quality needs to be controlled during service production and delivery. In contrast, quality assurance of goods is controlled in factories. This consideration assigns to service personal an active role throughout the service process. Industrial reform suggests standardization as a measure and assurance of service quality. Many activities in services are dependent on the varying performance of service employees, so mass production or standardization of services is not possible as with an industry. Rust and Miu (2006) state that the product development process is concerned with the assurance of quality, while service companies need to ensure the quality of a service beginning with the first service request through to the last customer service provider interaction. This is related to the fact that delivery is an internal, inseparable part of service.

The inseparable nature of services also brings new marketing concepts for service employees and customers, as described in section 2.1.4.3.

Perishability

Scottish economist Adam Smith (1776) considered services different than goods since services are *perishable*, which denotes that they cannot be stockpiled for later use. In his work *Wealth of Nations* he distinguishes productive and unproductive labor where the value of productive labor could be stored as inventory for later use or exchanged, while the value of unproductive labor perishes as it is consumed. With unproductive labor he emphasizes occupations, which could be assigned to labor services.

Service providers cannot store service as quantities in their inventories, but they need to store service capacity. Supply and demand function of services cannot be controlled like goods. Service providers need to make sure that they can handle increased demand without lowering the service quality. Services in the information technology field are exceptional with the imperishable nature of computer software. While services cannot be inventoried, knowledge can be stored and reused later on in further processes. In this context, knowledge could be considered as a tangible entity of the service since service customers do not pay for the written software code, but rather for the whole information system and maintenance services (Gummesson, 2007).

A good example of perishability is the validity of a flight ticket. A ticket is only valid and has value for a given passenger on the exact day and time of the flight. It perishes after the travel.

Heterogeneity (Variability)

Industrialization aimed at the standardization of products and quality assurance. Automation enabled mass production with faster rolling assembly lines and increased production costs. In opposition to standardization, most services require customizability and service consumers demand personalization. Heterogeneity corresponds to variability of services, which was first proposed by British economist Joan Robinson (1903–1983). Services are highly variable and the service output depends on many controllable and uncontrollable factors associated with the service provider or customer. Service quality cannot only be achieved with skilled employees and proper service settings. Customer satisfaction and expectation fulfillment could depend also the customer's own participation and performance. For instance, if a student does not pay attention during lectures, his grades do not necessarily indicate poor teacher performance as a service employee. Additionally, for the same service domain, geographical factors, governmental regulations, and social requirements could also trigger variability of services. International service providers could show regional differences in their services considering these factors.

Variability should not only be considered as a disadvantage since, due to their variable nature, services are easier and more flexible to customize. In many service domains, customers demand personalization of service delivery. Due to unique consumer preferences, services need to be consumer-centric. A standardized service output is not achievable and also cannot fulfill varying consumer expectations. Wyckham et al. (1975) state that service companies use variability as a benefit to differentiate from competitors since flexibility of services is an important customer demand.

2.1.3 Classification of Services

Hill (1977) classifies services into various groups depending on the affected entity at the end of service delivery.

Services Affecting Goods

Services affecting goods change the condition of a physical good (like cleaning a house, repairing a car, transporting a package, painting a wall, etc.). These conditional changes might have a permanent or temporary effect on the commodity. Cleaning a house is only a temporary change of condition since a house could get dirty in time, but repainting a sport car would have a permanent effect on its value. The physical change of the sports car would be irreversible, but a transported package could be brought back to its origin so its change would be reversible.

Services affecting Persons

Services affecting persons lead to mental or physical changes for the service customer. Major examples could be passenger transportation, medical treatment, tattooing, and manicures, as all end up with different physical changes. In contrast, education and entertainment (movies, concerts) target mental changes.

Services affecting persons could be sub-grouped according to the number of actors that benefit from the service (as *individual* and *collective services*). Individual services affect a single person at a time and *collective* services could affect multiple persons. Typical examples of *collective* services are entertainment services or public transport, where huge groups of persons perceive the same service. Meanwhile, a hairdresser can take care of one single customer at a time, so hairdressing is mentioned under individual services. Education services could be *collective* in schools, but private lessons with instructors could be also seen as individual services. This consideration is also valid for health services. While a surgery is meant to be an *individual health service*, group therapies could be listed under *collective health services* (Ricci, 2010).

Apart from these two main categories, Hill (1977) also refers to labor and financial services as distinct categories. In labor services, an employee provides a service for the benefit of his employer against a salary. As the fifth and last group, Hill (1977) refers to public services, which include public administration and defense provided by a state or local government.

ISIC[4] classifies the top economic activities according to the following table:

[4] International Standard Industrial Classification

A - Agriculture, forestry and fishing
B - Mining and quarrying
C - Manufacturing
D - Electricity, gas, steam, and air conditioning supply
E - Water supply; sewage, waste management and remediation activities
F - Construction
G - Wholesale and retail trade; repair of motor vehicles and motorcycles
H - Transportation and storage
I - Accommodation and food service activities
J - Information and communication
K - Financial and insurance activities
L - Real estate activities
M - Professional, scientific, and technical activities
N - Administrative and support service activities
O - Public administration and defense; compulsory social security
P - Education
Q - Human health and social work activities
R - Arts, entertainment, and recreation
S - Other service activities
T - Activities of households as employers; undifferentiated goods- and services-producing activities of households for own use
U - Activities of extraterritorial organizations and bodies

Table 2-1 : Classification of top economic activities (ISIC Revision 4)

The first two activity categories (A, B) could be listed under primary and the third category (C) would belong to the secondary sector. The remaining 18 sectors can be considered as service-oriented economic activities.

Furthermore, ISIC categorizes the mentioned top economic activity sections into divisions, and divisions into groups, based on the common properties of activities. The following table demonstrates grouping of *information service activities*, which are listed under section J (namely *information and communication*).

Section: J	**Information and communication**
Division 58	Publishing activities
Division 59	Motion picture, video, and television program production, sound recording, and music publishing activities
Division 60	Programming and broadcasting activities
Division 61	Telecommunications
Division 62	Computer programming, consultancy, and related activities
Division 63	Information service activities
Group 631	Data processing, hosting, and related activities; web portals
Group 639	Other information service activities

Table 2-2 : Detailed classification of information and communication services (ISIC Version4)

2.1.4 Marketing of Services

This section focuses on the marketing aspects of services and the role of clients as active actors, which could also play an important role in service quality assessment.

Product-centric traditional marketing consists of four *Ps*, namely ***P**roduct, **P**rice, **P**romotion, and **P**lace.* The marketing logic of 4p for a good oriented company assumes that it manufactures a variety of products, applies a pricing policy, promotes them to attract the attention of customers, and finally distributes them so that end-customers can buy them. The four Ps denote the capabilities of a company, or the dimensions, which are under the control of the firm itself. These four factors could be used strategically to influence new potential customers or keep current customers loyal. The 4p marketing mix is product-centric and the concrete product is always in the highlight during considerations about price, promotion, and place.

The company management decides on the variety of their products, as well as their quality, design, size, packaging, name, and properties, which are called *product features*. When the product is developed, proper price, discount policies, and various price related issues are discussed, including activities like advertising, sales promotions, or any public relations events to elevate product sales. The last P (*place*) considers distribution—ways for the product to reach its end customers. The company decides on various possible sale channels, mediums, and ways of transport to different locations. Companies also need to make decisions about inventory and assortment policies as a part of their marketing strategies.

The primary goal in marketing a certain product is finding a suitable target market, which demands or is ready to buy a product with certain features. These product features are determined by the producer and offered to end customers. A product market or certain market segment contains potential customers of the company that share a specific demand or need to buy the provided product. Potential buyers of a product are grouped under a common market segment and remain anonymous with common demographic or geographic attributes like gender, age, income level, or location. The marketing staff of a company needs to analyze both the product and its target market, make surveys, and organize public relations events to keep in touch with their customers and understand their needs and requirements. Based on their findings, they recommend improvements or extensions to the product development team. These procedures are repeated in a loop during the whole product life cycle, and the mentioned marketing activities beginning from the production phase through to sale is a closed process without direct customer interaction.

Due to the unique characteristics of the services mentioned in section 2.1.2, the product concept of marketing mix is slightly different for service marketing. Primarily, the product itself is intangible and variable in analogy to the variability of services. The product of the same service provider could vary from one service transaction to another. Potential service customers do not only rely on service specifications provided by the service company, but also on other clues including company reputation, customer reviews, and observable tangible components of the service provider.

Pricing strategies of services might also differ from goods due to various factors. For instance, service providers do not count on price of quantities like goods. Service values might be measured with rates, fees, admissions, charge, etc. In the case of driving lessons and cleaning services, the amount to be charged to service customers could be calculated by time. Car repair services or surgery might depend on the complexity of the activity, and required physical goods and support services could determine the service fee.

Due to inseparability, services are delivered as they are produced, so there is no place for middleware or resellers. Services need be to delivered directly to service customers. As a consequence, exporting and transportation of services is not possible.

Another important challenge of service marketing is the promotion of services. For many service domains, differentiation from competitors is harder than with goods. The main reason for this challenge is that services are provided as a set of activities that make a process with a variable duration. It is not always possible to change service specifications or these processes as a target of service promotions. In general terms, service providers try to promote their services by impressing their customers and enhancing the user experience during the customer's interaction with the provider.

2.1.4.1 Touch Points & Service Encounters

Throughout the whole service process, a service customer could interact with service employees and other service customers. Additionally, service customers may need to interact with the technology, equipment, or physical surroundings of the service provider. In service science, all possible contact points of service customers with the provider are considered as touch points. More complex businesses like hospitals, airports, or banks have more touch points than smaller service domains. Depending on the service domain, certain service touch points are more significant than others, and these touch points are referred to as *moments of truth*. For instance, a bank's touch points include its physical building, bank-cards, self-service machines, customer assistants, and call-centers. Self-service machines can also be considered as one of the moments of truths since a bank customer expects to draw money with his bankcard any time he approaches the bank. A defective machine would result in a dissatisfied and disappointed bank customer (Clatworthy, 2011).

In general terms, each interaction of the service customer with one of the service touch points results in a service encounter that shapes his opinion about the service provider. Each service encounter could result in satisfaction or dissatisfaction. In service science there are different definitions for the term "service encounter." Shostack (1985) considers service encounters as "*a period of time during which a customer directly interacts with a service.*" Shostack (1985) does not limit service encounters to inter-human interactions. According to his definition, a service encounter could also occur between a customer and any physical service equipment. Czepiel et al. (1985) refine the definition of a service encounter as a social interaction between human actors of the service and exclude non-human touch points.

Service encounters need to be a subject of service marketing since they are critical for achieving customer satisfaction, building trust, and proving service quality to win a loyal customer. Throughout this thesis a service encounter is considered as any interaction between service customer and service provider that adds a service value for the service customers.

2.1.4.2 Extended Marketing Mix with 7Ps and Relationship Marketing

A product oriented marketing mix of 4p does not consider unique preferences of service customers and their active participation, or the role of service employees during service delivery. As mentioned in previous sections, services could be considered as processes with sets of activities. These activities include various service touch points and user interactions that are not only important for service marketing, but also for service quality evaluation. Service companies could differentiate themselves from their competitors in terms of these touch points. Additionally, a traditional marketing mix of 4p does not cover the aspects of service encounters or the importance of touch points for service marketing. Brown et al. (1991) offer three additional Ps to the existing 4p marketing mix, namely people, processes, and physical evidence. Today this new marketing mix is known as *Extended Marketing Mix*.

People refers to any individual, ranging from company employees and support staff to other customers of the company that interact with each other. Interaction of service employees is relevant for the success of a company, but customers also interact with each other before and after making purchase decisions. Customers could easily influence each other in negative or positive ways in terms of *word-of-mouth* communication. Gummesson (2007) compares loyal customers of a service provider with the external marketing staff of the company. Marketing personnel of a service company cannot always be where they need to be to fix the company's reputation. However, satisfied loyal customers promote services through *word-of-mouth* communication. Other customer impressions and experiences can be considered as more valuable and objective than information provided by service employees or service specifications. Furthermore, Gummesson (2007) proposes the term *part time marketing personnel* for service employees whose actual task is not marketing. These employees represent the company, and their attitude during the whole service process is very important in the eyes of customers. Although they are not qualified for marketing, they have to market provided services and should be capable of answering inquiries, encouraging potential customers to become customers, and elevate customer satisfaction.

Additionally, service customers need to participate in service production, and their involvement could influence service quality just like service employee performance. The service-marketing triangle of Grönross (1998) described in section 2.1.4.3 demonstrates the active roles of service customers and company employees in detail.

Physical evidence refers to tangible components that are highlighted and promoted in addition to core service output. As mentioned above, the tangible product is missing in service marketing, but there is much physical evidence about service providers that customers interact with. Physical evidence includes service facility, design, equipment, employee's dress, or any other tangible component that appears during any service encounter. This could be compared with the packaging and labeling strategies of goods marketing that help customers to identify brands and leave a first impression.

Bitner & Booms (1981) developed the concept *servicescape* to emphazise the impact of the phyiscal surroundings of the service. According to the definition of Bitner and Booms (1981), *the servicescape is the environment in which the service is assembed and in which the seller and the customer interact*. The servicescape plays an important role for many service domains since enviromental dimensions could enhance the customer experience by influencing his physiology or proving a convenient enviroment for location-based services.

In general terms, physical evidence is the visible component of the service provider, and it shapes customer expectations and opinions about the service company. From the perspective of service marketing it can be considered as a controllable dimension to represent company image and reputation. Many services are provided with exact same service specifications. Phyiscal evidence is an important dimension that the service provider can focus on to differentiate itself from its competitors.

Processes are the flow of activities with various steps and different levels of customer involvement. One of the most unique characteristics of services is their process nature. In contrast to customers of physical goods, service customers do not perceive the promised service in a certain point in time. The form, duration, place, and similar factors in these processes could be used as a differentiation strategy in service marketing. As mentioned earlier, service customers might demand a certain level of flexibility in service delivery. Service companies can demonstrate their service processes on service blueprints to ensure the same standards in delivery and find out possible extension points. A service blueprint aggregates the sequential set of activities from the perspective of a service customer. Required service touch points, service encounters, support processes and physical evidence can be modeled on different dimensions (Bitner et al., 2007).

The service blueprint demonstrated in section 4.1 shows possible restaurant customer actions, onstage/backstage contact employee actions, and support processes. Service blueprints can also be used to understand the quality evaluation aspects of service customers. Service blueprinting is considering to be an important technique for service marketing.

2.1.4.3 Marketing Triangle

Grönross (1998) proposes a services marketing triangle to demonstrate differences between good- and service-oriented marketing perspectives. The marketing triangle for the product-oriented approach demonstrates the interactions between the customer segment, company, and product as marketing functions. The market edge of the triangle aggregates the market segment of the product that corresponds to customers. The company continuously improves the provided product. In this context, the product is the outcome of a closed production process where individuals in the market segment do not participate. Marketing staff (or the firm) leads research on their target market and analyzes the technological capabilities of the firm to implement additional features or improve the existing product. Marketing activities of the company are considered as giving promise to the market segment, which is fulfilled by the offered product.

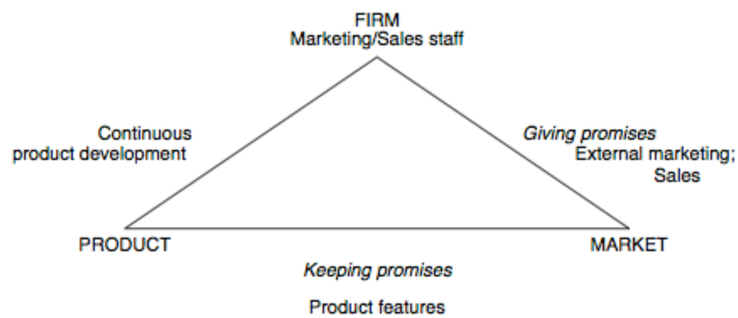

Figure 2-1 : Product-oriented marketing triangle (Grönross, 1998)

In the service-oriented marketing triangle, the market segment of the mentioned model is replaced with service customers, who require individual treatment for their unique demands. In contrast to the product oriented marketing triangle, individual service customers are considered to perceive different service outputs.

Apart from the change of the market segment to customers, product is replaced with various components of the service process, including personnel, technology, knowledge, and customer time. The service company tries to fulfill service specifications in terms of these factors rather than providing product features. Service employees are the main actors in this process that try to create a service value in the light of available technology, knowledge, and service settings. They need to act as part-time marketing personnel during service encounters. The part-time activity of service employees is considered to be interactive marketing and is typically unplanned and unorganized, occurring upon the demand of the service customer. Since not all service personnel that encounter service customers are qualified in marketing, employees could experience unexpected situations that might result in a dissatisfied customer. Analogous to continuous product development, service-oriented companies need to educate their service personnel by continuously improving service settings and enabling technologies. These actions are considered *internal marketing activities* inside the service company. Service companies need make sure that they own sufficient personnel with knowledge and technology to deliver the service with the expected level of quality.

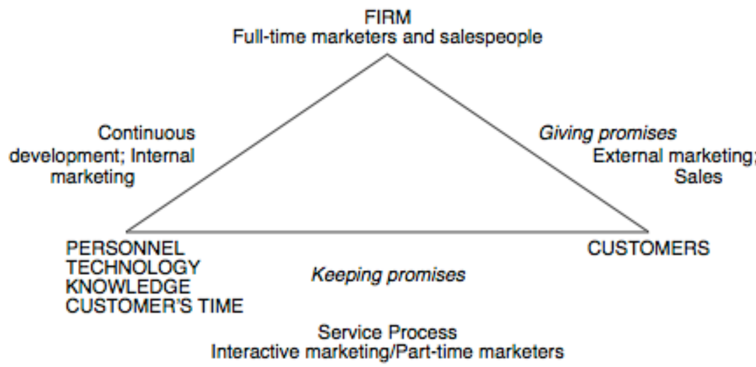

Figure 2-2 : Service-oriented marketing triangle

2.1.5 Service Quality Evaluation (Quality Models)

Service quality is expressed by most service customers simply as *customer satisfaction.* Customer satisfaction is a variable measure and is not deterministic for all service customers. On the other hand, customer satisfaction does not necessarily indicate service quality, considering different levels of customer expectations. This section introduces major service quality models and clarifies relationships between customer satisfaction, service quality, and the role of customer expectations.

Service aggregate different sets of activities where the customer participates in various service encounters with different outputs and purposes. A customer's satisfaction denotes the level of pleasure and gladness that he obtains during these service encounters and his attitude after service delivery. Service satisfaction can only be achieved if all service encounters have been experienced with positive impressions. If a service customer feels a drawback or insufficiency in any service encounter, this could result in dissatisfaction or a decrease in the customer's expectations of further service encounters.

According to customer-centric service philosophy, the primary aim of service providers is achieving customer satisfaction. In negative cases, companies should investigate the reason why the customer was not satisfied or what exactly in the service delivery could have made him dissatisfied. Customer feedback is essential for improving quality of service and value offered to other customers. It should be considered that despite high investment in advertising, most new service customers can be gained through existing customer referrals and recommendations. Dissatisfied customers who move to a competitor can tell companies what to improve better than the management staff of the company can (Reichheld & Sasser, 1990).

Various academics have developed models of service quality that describe relations between customer satisfaction, expectation, and service quality. This section introduces some of these models, including the *Expectations Confirmation*, *Nordic*, *Three-Component* and *Gaps Model of Service Quality*. These frameworks could be used to improve service quality and help companies to find weaknesses in their businesses.

2.1.5.1 Expectation Confirmation Theory

The disconfirmation of expectations (Expectation Confirmation Theory) proposed by Oliver (1980) is considered to be one of the most dominant models. This model argues that post purchase satisfaction is a function of expectation and perceived performance. This generic model could also be applied to service domains where perceived performance denotes overall service experience. Expectations correspond to the expected service output based on marketing activities of the service provider, word of mouth communication, or prior service experiences.

The model assumes that unmet customer expectations have negative affects on the service perception and customer attitude towards the service provider. Customer expectations are considered to be a threshold that indicates if the perceived service could be considered to be satisfying. As demonstrated in Figure 2-3 : Expectations confirmation model, a service customer could end up with three different outcomes when he compares his expectations from the service with the perceived performance. Therefore, if a customer has low expectations from the service, he can be satisfied even with low service quality. In contrast, if the level of expectations is too high, even a proper service delivery may not satisfy service costumers.

A good example of this situation could be bistro kiosks in gastronomy, where people go to have a small snack for a reasonable price in a very short time. Customers do not consume their food in a confortable environment with tables or chairs, but they go to these places aware of the available service setting, so the absence of a comfortable atmosphere does not disturb them.

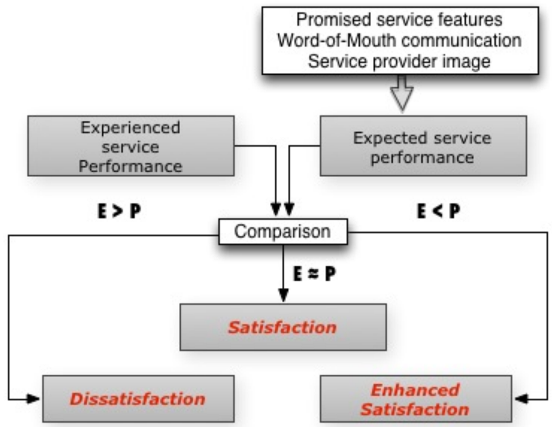

Figure 2-3 : Expectations confirmation model[5]

The expected service performance is most noticeably shaped by customer-to-customer interaction. Customer satisfaction is naturally a significant requirement for customer loyalty and positive *word-of-mouth*. Today's service companies aim to have long lasting relationships with their customers in order to increase their profit margins, rather than focusing on one-time transactions. Considering customer satisfaction as an economic goal, service companies could gain more market share and customer loyalty and improve their brand reputation, which would benefit the company in the long term in the form of increased margins and revenue (Jonathan & Nash, 2003).

2.1.5.2 The Nordic Model

The Nordic Model of Grönross (1998) represents service quality in two distinct dimensions, namely technical and functional service quality, which originates from disconfirmation of expectations model. Technical quality refers to service outcome that the customer perceives. Functional quality refers to how the service delivery is operated and managed during the service process. The Nordic Model considers the image or reputation of a company to be a dynamic parameter that could influence quality perception in a neutral, positive, or negative way. In the long-term, a worsening or convalescent company image would also lower the expectations of service customers.

As an extension of the Expectation Confirmation Theory, service customers consider technical and functional service quality during their service experience. In other words, customers do not only compare the service output with their expected service performance. Functional service quality aggregates all service touch points, used resources, technology, and service personal. Customers expect an acceptable level of technical quality, while provided functional quality could elevate or decrease perceived

[5] Adapted to services from (Oliver, 1980)

technical quality. These distinct quality dimensions should be considered separately since insufficient technical quality is hard to compensate for with better functional quality. Service customers tend to evaluate functional quality as a secondary dimension only if the level of technical quality is satisfying (Grönross, 1998).

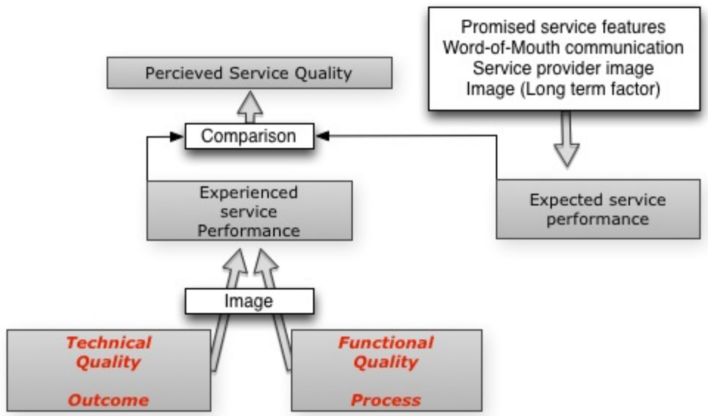

Figure 2-4 : The Nordic Model[6]

2.1.5.3 Three-Component Model

One of the most important dimensions of the *7P service marketing mix* is physical evidence as mentioned in section 2.1.4.2. The physical environment of the service also plays an important role in service quality perception by the customer. Rust and Oliver's (1994) three-component model extends the Nordic Model by taking the *physical environment* of the service into consideration. This model divides service quality into three distinct dimensions, namely *service product*, *service delivery,* and *service environment*. The service product is the outcome of the service that the customer benefits. In terms of many service domains, promised benefits are offered as small service activities that service customers experience throughout the service process. In this context, service product could be considered to be the core service benefit. The service delivery aggregates all activities and service encounters required for service value creation. The service environment describes external and internal factors that service customer encounter, including service settings, organizational structure of the service, and the physical environment. Rust and Oliver (1994) stress that only successful coupling of these three service dimensions could lead to desired service quality.

[6] Adapted from (Grönross, 1998)

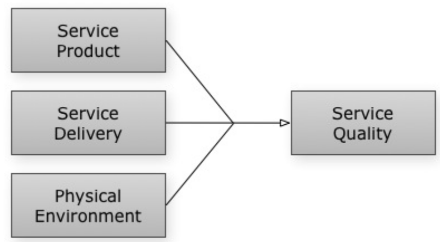

Figure 2-5 : Three-component model

2.1.5.4 The Gap Model (SERVQUAL)

SERVQUAL is another imported quality framework and an analytical tool to measure the gaps between customer expectations and perceptions. This tool can be used as a sort of benchmark to measure possible gaps between customer expectations and considerations of service management about customer expectations (Parasuraman, Zeithalm, & Berry, 1988).

The model proposes five potential sources of gap in the service organization and five service dimensions according to following groups.

Gap 1: The first gap describes differences between what the service customers really expect from the service and what the management thinks their customers expect. It is assumed that service providers do not really know in detail and understand what customers expect. This gap can occur if there is insufficient internal communication between contact people and management staff. If the market orientation of the company is not right or the market's demands do not match the service specifications, expectations of customers cannot be fulfilled.

Gap 2: The second gap targets quality expectations of the management and formal service quality specifications. This can occur because of wrong service design, insufficient service settings, or missing service quality standards.

Gap 3: The third gap describes poor service delivery performance so that the service quality specifications cannot be fulfilled. This can appear due to lack of qualified service employees or insufficient service technology. It might be related to poor team communication or lack of control in larger service domains.

Gap 4: The differences between service delivery and formal specification or in other words promises given over external communication by the company. This could occur if service specifications do not denote realistic goals for the service provider. In large service businesses, separate departments are not aware of each other's capabilities and productivity, which could be related to inadequate internal communication.

Gap 5: Gap five describes the gaps between expected customer quality and perceived service quality analogous to the expectations disconfirmation model.

Apart from these mentioned organizational gaps, SERVQUAL distinguishes five different service dimensions that service customers experience during various interactions with the service provider.

Reliability is the ability of the provider to perform the promised service accurately and dependably.

Responsiveness is the willingness to help customers and provide desired service instantly.

Assurance aggregates four sub-categories to represent the knowledge, attitude, and skills of employees.

> ***Competence*** represents the required skills and knowledge to perform the service.
> ***Courtesy*** is the attitude of the employee towards service customers (politeness, respect, friendliness, hospitability, etc.)
> ***Credibility*** is the honesty of the employee.
> ***Security*** is if a service employee completes his tasks without any risk or danger during service delivery.

Empathy aggregates three sub-categories related to the treatment of individual customers by service employees.

> ***Accessibility*** is being easy and approachable towards customers.
> ***Communication*** includes listening to customers individually, considering their thoughts and concerns, and informing them in a language or "form" that they can understand.
> ***Understanding the customer*** and their needs.

Tangibles include equipment, personnel, appearance of facilities, and any communication material.

SERQUAL is generally a good analytical tool for service providers to adjust their service settings and discover their weaknesses. Service providers need to discover possible gaps in any of their service processes and improve them with possible measures. Through this tool they can judge their processes from the customers' perspective. They can also track customer expectations on different levels of service delivery to adjust their service settings.

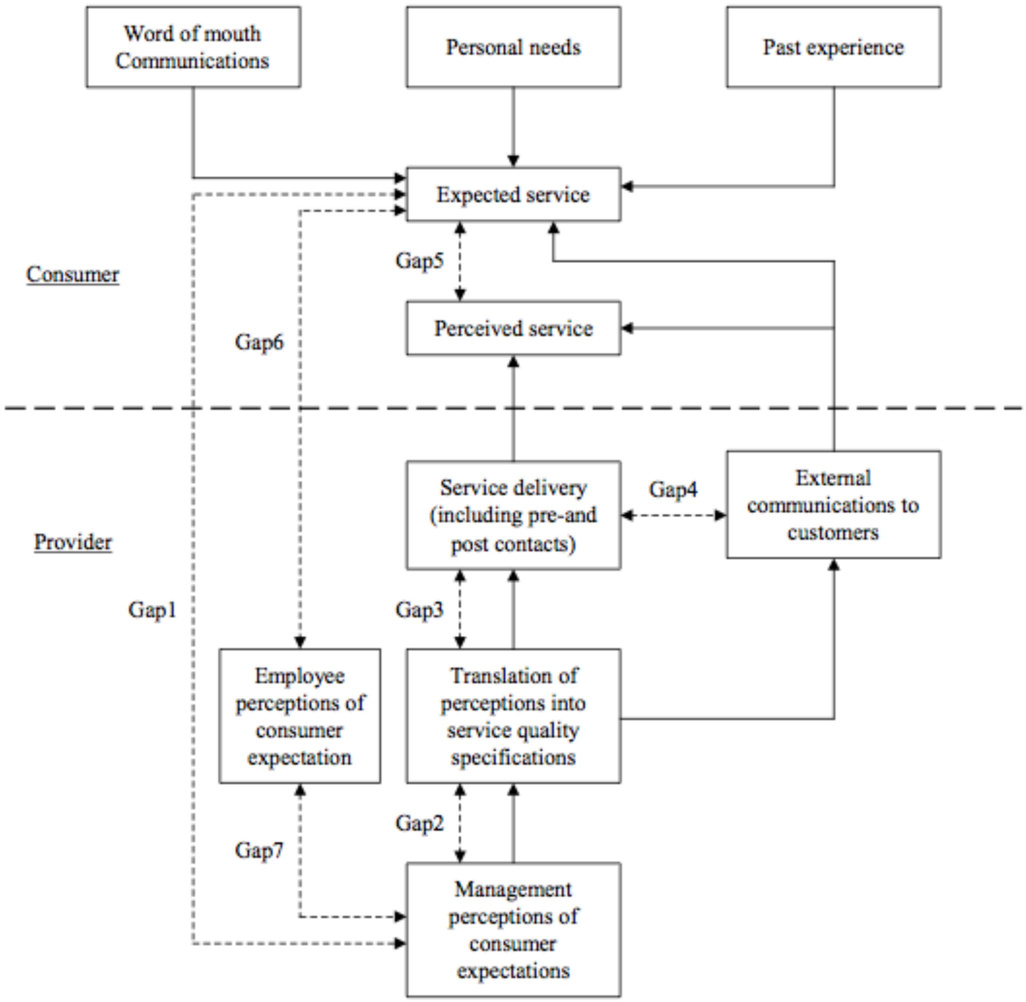

Figure 2-6 : Model of service quality gaps (Parasuraman et al., 1988)

2.1.5.5 Multilevel Model

Multilevel Model is an extension of the three-component model proposed by Dabholkar et al. (1996) in order to test quality of service in retail stores. It suggests that service quality can be modeled in three separate dimension levels, namely, overall perception of service quality, primary dimensions, and sub-dimensions. Although it aims at service quality in retail stores, it could be generalized for other service domains by adding domain-specific dimensions and assigning proper sub-dimensions.

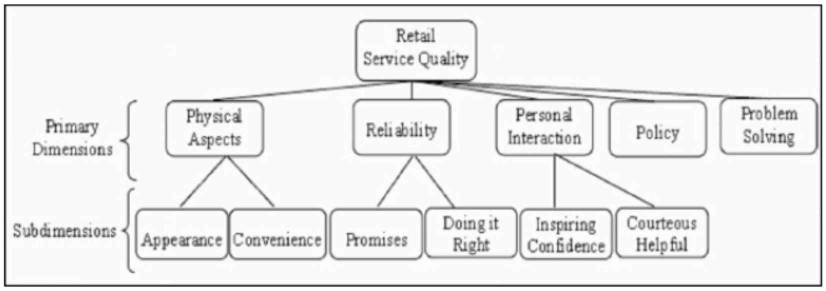

Figure 2-7 : Multilevel model of (Dabholkar et al. , 1996)

2.1.5.6 Integrated Quality Model

Brady et al. (2001) propose another hierarchical model, which was created by considering expectations of service customers from various industries, including fast food, photography development, amusement parks, and dry cleaning. This model could be considered as an extension and general form of the *multilevel* model, combined with the *Nordic* and *SERVQUAL* models. The model assumes that the service quality perception of customers occurs in three distinct dimensions: *interaction quality*, *physical environment quality,* and *outcome quality*. *Interaction quality* could be compared with *functional*, and *outcome quality* with the *technical quality* of the *Nordic Model*. The model assumes that each of three primary dimensions consist of three sub-dimensions as demonstrated in Figure 2-8 : Integrated hierarchical model. Each sub-dimension is evaluated over three criteria, including reliability, responsiveness, and empathy as discussed in the SERVQUAL model. These dimensions are represented by leaves, which correspond to evaluation factors of sub-dimensions by service customers (Shu, 2010).

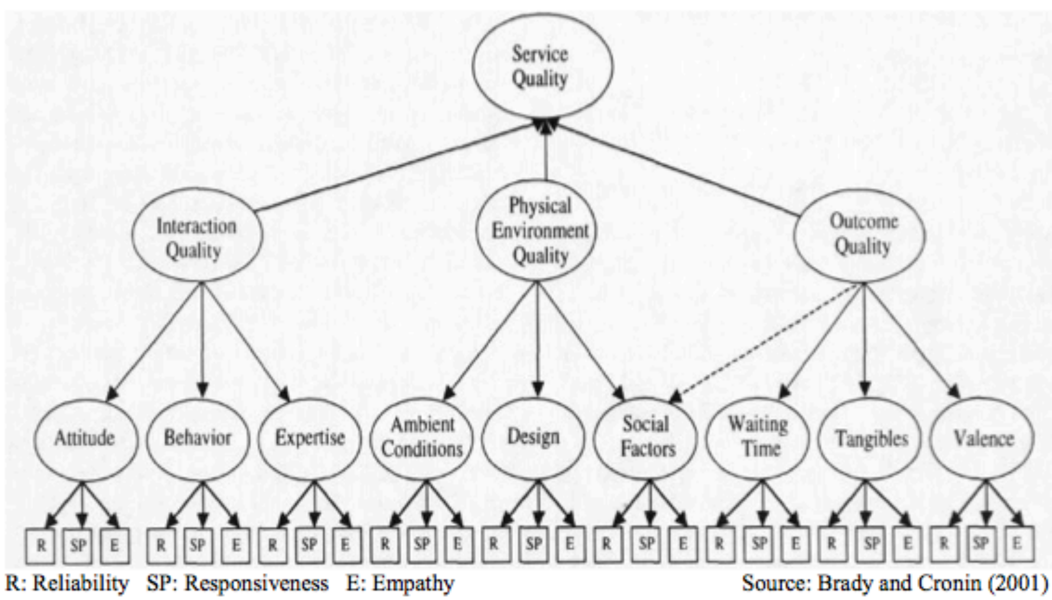

Figure 2-8 : Integrated hierarchical model (Brady & Cronin J. Jr, 2001)

2.1.6 Customer Expectations of Service

Customer expectations are pre-purchase beliefs about a service that help service customers to judge the service performance. As mentioned in service quality models, customer expectations play an important role for service quality evaluation. The disconfirmation of expectations model assumes that the service quality evaluation is implicated in comparison with customer expectations and perceived service. The gap model of service quality clarifies potential gaps between the service organization and the customer on different levels by confronting the expectations of both actors.

Zeithalm et al. (1993) proposes a conceptual model for articulating service customer expectations. The model classifies customers' expectations into three groups, namely *desired service*, *adequate service,* and *predicted service*. Furthermore, different types of customer expectations and certain factors that influence these expectations are also discussed. Predicted service describes what customers expect to see. Customers naturally want to perceive a perfect service though there exists a threshold between the desired service and tolerable service quality level. This floor level is considered to be adequate service that the customer would still accept. If the level of service quality drops below this adequate level, customers start to complain about the service and probably leave the service provider dissatisfied.

Possible deviations between desired and adequate service level is described as a *zone of tolerance*. Zone of tolerance is a window wherein customers are ready to accept quality variations. Depending on the customer or service type, this zone could be narrow, meaning the customer wants consistent service and is not flexible in the name of service quality.

As seen in

Figure 2-9 : Customer expectations modelthere are various factors that affect desired service quality. Personal needs describe social, psychological, or physical conditions, which are required to make customers comfortable. Customers of a fast food restaurant expect quick service, a convenient place to consume their meal, and a clean environment. In contrast, customers of a luxury restaurant demand a nice ambiance, elegant employees, fine food, candlelight as an option, and maybe soft background music. Apart from physical needs and preferences, customer's emotions during service delivery could dramatically change their satisfaction level. A passenger who would prefer a quiet compartment to read his book would probably complain if he had to travel with a group of loud teenagers during the whole journey. In contrast, a passenger who would like to socialize with other passengers in the same compartment could get bored if he couldn't find company during the journey. This scenario indicates how the other service customers (People of 7P) could influence customer satisfaction.

Enduring service intensifiers are described as "*individual, stable factors that lead the customer to a heightened sensitivity to service"* (Parasuraman et al., 1988). Derived service expectations are those that are driven by a third party. Outsourcing in service business would refer to this type of expectation, since service employees also demand other parties in the service delivery process. In addition, customers who have experience in the services sector mostly have a basic understanding of the service concept, and this influences their attitude towards the service provider. Since they already know the service settings with *behind the scenes*, this awareness has an influence on their level of desired service.

Transitory service intensifiers make service customers sensitive to service in the short term. In the case of a car accident, an insurance company customer requires response and assistance from an insurance consultant as soon a possible. Through the emergency, the customer becomes aware that the service is really needed. For instance, in the case of an electric blackout in a bank, customers could have to wait longer than usual, and this would also increase the level of adequate service and narrow the zone of tolerance.

The more *service alternatives* a customer has, the higher is the level of adequate service. In contrast, if there are not enough alternatives where the same service could be perceived, the customer has a wider zone of tolerance. In this case, service providers could monopolize the market and determine the standards, causing customer demands and expectations to be ignored (e.g. public services).

Situational factors that correspond to temporary changes could lower the level of adequate service. For instance, in the minutes following a natural disaster, if the GSM operator is blocked by a huge connection overload, customers tend to complain less about the problem since they are aware of the extraordinary overload. If a restaurant customer has to wait longer than usual for the waiter to order his meal, this could be tolerated during the *rush hour* of the restaurant. In contrast, if the waiter does not pay attention when there are not so many guests, a customer would interpret this as a lack of service.

As mentioned above, a service customer can take an active role in many service domains. During this participation, he might need to perform certain tasks. If the customer cannot perform his own tasks, the zone of tolerance could extend. A good example of this type of factor is telephone speech with a customer call center for technical support. If the customer cannot explain the technical problem in detail, the right solution or a helpful consultation cannot be expected.

There are also several factors that could influence desired and predicted service quality. Positive *word-of-mouth* information or positive past experiences could elevate them both. The past experience does not have to be with the exact same service provider. It could have been with any provider within the same industry, and through that customer has the chance to compare different service providers. The more service experience a customer has, the narrower the zone of tolerance is.

Explicit service promises include any informative material about the service or what service employees promise about their service to their customers. If the given promises are high, customer will naturally have higher expectations from the service and the zone of tolerance will be narrower.

Implicit service promises are clues about the service, like price or touch points. Customers use price and tangibles as a quality indication to pre-evaluate service quality. Customers who are ready to pay higher prices for a certain service think that a higher price implies higher service quality. At the end of the service delivery, customers want to see that the paid price is worth the perceived service. If the customer feels that the price was high in relation to the delivered service level, he could feel dissatisfied.

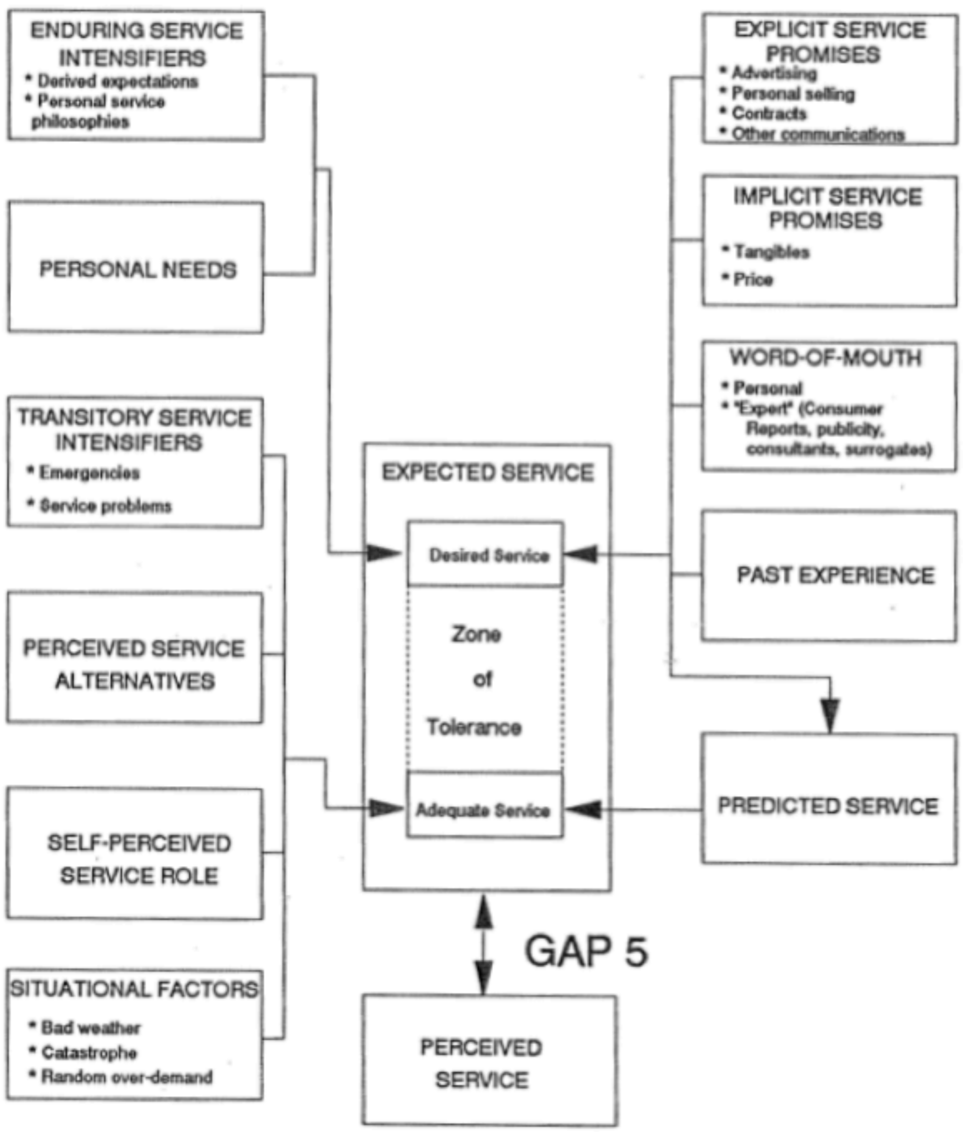

Figure 2-9 : Customer expectations model (Parasuraman et al. , 1988)

2.2 Recommender Systems

2.2.1 Background and Introduction

Recommender systems could be considered as software tools to simplify decision-making processes by filtering out a huge set of items and presenting to end users a smaller set of relevant items. A recommendation item is an abstract term to denote any entity, including books, CDs, songs as retail products, or services. Recommendation engines originate from the information retrieval field of computer science. The World Wide Web has already become an enormous network encompassing 1,471 billion indexed pages, and it keeps growing day by day[7]. In this complex network with a huge amount of information sources, Internet users need assistance in finding relevant information. Due to this requirement, the oldest application area of the recommender systems is the search engine. RSs overcome the information overload problem by filtering out search results depending on the explicitly provided search query. Presented search results correspond to most relevant *top N* query results, which could be shown to the user as ranked links of items.

Apart from search engines, since beginning of 1990s the number of e-commerce sites has increased rapidly, and many of these sites have already become big companies with millions of dollars of yearly revenue. From the perspective of e-commerce, recommender systems have become important marketing tools, and for their customers, a convenient sales assistant. For users, RSs ease the information overload and make the browsing experience easier. In the long term, they could help service providers to gain loyal customers through user-friendly interfaces and personalized recommendation techniques. In the scope of this section, introduced techniques mainly target recommendation engines in e-commerce and do not cover details of recommender systems as information filtering tools for document based search engines.

Recommender systems aim to find relevant items for the active user depending on his preferences, prior ratings, reviews, preferences, interests, and geographical or demographical information. Fundamentally, RSs require user and item entities as data models. Recommendation algorithms rely on a *utility matrix* or *user-rating matrix* that store the item ratings of individual users. Utility matrix of a RS with *m* users and *n* items theoretically contains *n x m* values for the rating data. In real world scenarios, this matrix contains many missing values since not all users are interested in the whole set of available elements. A mathematical interpretation of RS is filling out those missing values by finding certain patterns between users and items and representing predicted ratings as ranked lists to the active user.

This chapter summarizes the state-of-the-art in recommendation engines and compares different recommendation techniques in terms of their drawbacks and advantages. Additionally, fundamental mathematical and statistical concepts of RSs are also introduced shortly for the empirical part of the thesis.

[7] www.worldwidewebsize.com last visited (30.03.2013)

2.2.2 Core Functions of Recommender Systems

Recommendation systems have various roles and functions on behalf of active users and service providers[8] depending on the application domain.

Increased Conversion Rates[9]

Most e-commerce site visitors browse Internet sites without purchasing any items. If the attention of the user could be attracted through an interesting recommendation, site visitors could be converted into buying customers. Additionally, a user might tend to leave the e-commerce site if he cannot find the demanded item. A recommender system could suggest an alternative instead of this unavailable item that the user could potentially buy.

Increased Cross Sell

From the perspective of an e-shop, the main function of a recommender system is to increase revenue by selling additional items to users besides the explicitly chosen ones. Additional items are those that a user would not have purchased without the suggestion of the system. Recommender systems in e-commerce tend to show a sequence of recommendations or a bundle of item sets that fit each other. For instance, if a user orders contact lenses, it is reasonable to show lens cases, contact lens solution, and any related lens accessories as further recommendations.

Customer Loyalty

The main principle of the service philosophy is building long-lasting relationships with service customers. This is also an important business strategy for e-commerce sites where different costs at different e-shops are only one click away. E-shops try to keep in touch with their existing customers by sending them newsletters, promotions, or any other reminding information about the site. Some of them provide loyalty programs where customers collect points for their purchases.

Recommendation engines can help e-shops build long-lasting relationships with their customers in the long term. Regardless of the recommendation technique, the quality of their recommendation increase over time after collecting enough evidence about the user (mainly as user ratings). After a long interaction history with a recommender system, users do not tend to leave and go to another site, since they would need to make the same effort to be able to get novel recommendations from the new system (Schafer et al., 1999).

[8] E-commerce sites

[9] In Internet marketing, conversion rate is the proportion of user visits to desired actions of the service provider, including purchases or advertisement clicks.

Increasing Overall User Satisfaction

A user-friendly designed recommender system can also increase overall customer satisfaction if it can make site navigation easier and faster for its visitors.

Personalization

Personalization has become more than *"nice to have"* in information retrieval systems (including search engines) in recent years. For instance, *Google search engine* considers recent user queries and browsing history and starts showing suggestions as the user types keywords into the search field. The founder and CEO of Amazon.com Jeff Bezos, who was designated *Person of the Year in 1999* by Time Magazine, reflects on the importance of personalization with his quote *"If I have 2 million customers, I should have 2 million stores on the Web"* (Schafer et al., 1999).

Recommender systems can make visitors feel special and treat individuals as valuable customers. In personalized recommender systems, as users interact with the system more frequently, the system can learn more about the user and his profile can be extended and updated accurately. As users realize that they are getting reasonable and consistent recommendations, they enjoy using recommendation engines and become loyal customers. This can also increase trust in the recommendation system.

Marketing and Sales Tools

Recommender systems are capable of storing valuable information about system users, including demographic geographic information, transaction history, item ratings, and possibly textual feedbacks. This collected information can be used to understand what customers demand, prefer, and dislike.

Service providers try to establish long-term relationships with their customers and do their best not to lose any customers while attempting to attract new customers. All collected information about the buying behavior of users can be reused for management and marketing purposes.

2.2.3 Properties of Recommender Systems

In literature, recommender systems are mainly classified based on their recommendation techniques. Wörndln (2009) introduces an alternative classification for recommender systems based on various aspects.

Individual/Collaborative

Collaborative recommender systems consider information of other system users to generate recommendations. These systems operate with a user-item matrix to find similar users or alternatively a similar set of items in an *item-to-item* approach. Individual recommender systems do not require any information about other system users. They operate with the active users' preferences model to find relevant items. For instance, content-based recommendation techniques could be classified as individual.

Memory Based/Model Based

Memory based recommender systems take as input parameters the entire user-item matrices and compute similarities between items or users. Model based recommender systems operate with a mathematical or statistical model to predict recommendations.

Reactive/Proactive

In the case of reactive recommender systems, the user needs to enter a certain query explicitly in order to get recommendations. In contrast, proactive systems do not require any user interaction to operate.

User-Based/Item-Based

User-based recommender systems take as input user ratings, purchase history, or explicitly defined preferences. Item-based systems focus on item descriptions and consider similarities between items by comparing their attribute values.

Client Side/Server Side

Most of the recommender systems rely on a centralized approach where all information about the system users, items, and ratings are stored in a centralized database on the server side. In this approach, the client only represents the recommendation results to the user without any computation. Client sided recommendation systems are not very common, but virtual tourist guide applications for smart phones fall into this group that can operate without any data connection. These applications operate locally on the smart phones without interacting with a server to avoid roaming costs while abroad.

2.2.4 Information Sources for Recommendation

Regardless of the application area and system architecture, recommender systems have three major information sources, including recommendation items, users, and logs of user interactions with the systems. The complexity and structure of recommendation data depend on the recommendation technique and application area of the system.

Recommendation items and their describing attributes are especially very important for content-based recommender systems since item attributes need to be preprocessed and analyzed for comparison with user preferences. In real world scenarios, item descriptions correspond to one or more database tables depending on the domain and structure of the item attributes. For instance, in the case of a recommender system for restaurants, notable attributes to be saved in the database could be the restaurant category, cuisine type, price range, and restaurant location. An online shop could sell various product categories including books, movies, or digital mp3 players. The set of attributes that describes books and CDs is naturally smaller compared to the attributes required to describe the technical specifications of an mp3 player. A recommender system should be capable of coping with various types and numbers of item attributes without affecting the consistency of the final recommendation. As a golden rule of data modeling and mining, descriptive attributes of items should be always structured and the range of attributes should be limited.

Personalized recommender systems require various types of information from users regarding their preferences and demographic information. In pure collaborative filtering systems, users are considered with their item ratings, while demographic systems need specific information like gender, location, or age. Utility-based systems could optionally use specific demographic user information to operate depending on the application area of the system.

Ricci et al. (2011) considers user interactions with the system to be *"transactions,"* which are not necessarily completed purchase transactions. Implicit or explicit transaction information could be collected during any user interaction with the system. User ratings and feedback on items are explicit, while viewed pages, search queries, and bookmarks are considered implicit information. Recommender systems need to keep track of such interactions in order to update user profiles through machine learning algorithms or for generating association rules, which are required in rule-based recommendation systems. Rule-based recommendation systems generate predictions based on certain rules, which are acquired through the user interaction history (Pazzani & Billsus, 2007).

Collected user interactions could be also used for other purposes. For instance, a recommender system could assume that already viewed or bought items should not be recommended to the same user again, even corresponding rating is missing in utility matrix.

Textual comments are other examples of explicit user feedback. Due to the unstructured form of text and complexity of natural languages, textual comments cannot be used directly in recommendation generation. They need to be preprocessed and mapped to a numeric representation for that purpose.

2.2.5 Data Processing for Recommender Systems

Each recommendation system could operate with different data models based on application area or recommendation technique. Collected user or item data passes through a set of processing steps for normalization, scaling, and classification purposes. The first step in the recommendation process is preprocessing of the training data to make it computable for the analysis phase. Based on item or user attributes and ratings, recommender systems try to find certain patterns and generate predictions over these patterns. In order to locate these patterns, item or user attributes need to be compared throughout the whole recommendation process. These attributes could have originally different value ranges and structures, so they cannot be used in machine learning or analysis algorithm as raw input parameters without preprocessing. It might be required to filter, normalize, or transform this kind of data to a different dimension for computation purposes.

After preparing the training data, the second step in the recommendation process is the analysis or inspection of the training data. During data analysis, items and users could be classified into prior known classes or on wire created clusters. The purpose of data analysis is to create a proper user model that indicates the likes and dislikes of the active user for use in the prediction phase. Recommendation engines need to calculate similarities between individual items or users by comparing distances between their attributes.

The last step in the recommendation process is the presentation of relevant items to active users based on the findings of the analysis step. Recommendation engines compute their predictions based on a recommendation algorithm that could output a ranked list of recommendation items. Choosing the proper algorithm and presentation style depends heavily on the recommendation domain.

This section introduces the fundamental concepts and approaches required in the data processing and analysis phases of the recommendation process.

2.2.5.1 Preprocessing

As mentioned earlier, item or user attributes can be of different types and have different value ranges. Recommendation engines need to compute similarities between items, and recommendation results should not be affected by the type and range differences between item attribute types. In real world scenarios, a recommendation item could have attributes of Boolean, integer, categorical, textual, and fraction numbers.

Apart from the type of attribute, varying ranges of different attributes could have a huge impact on the similarity result. Numerical representations of ordinal and continuous attributes are ranked values but they need to be normalized or scaled based on the recommendation algorithm. For instance, the difference in ages of two users cannot be compared without normalization of monthly wage differences due to their different scales. Categorical or nominal values might need to be mapped to a numeric value so that they can be taken into calculation with other numeric attributes. Numerical representation of items or user attributes should be in the same scale so that higher attributes do not dominate the attributes with lower cardinality.

In principal, an item *a* with low overall rating and an item *b* with a higher overall rating should have a distance value as large as possible to indicate dissimilarity. One way to elevate distance measures between higher and lower ratings is by rounding rating values up/down depending on their position in a threshold value. This approach decreases computation overload of the recommendation system but is not preferable in systems where accurate ratings are essential. In general terms, the normalization of an item data set would make the computation more efficient but it should not cause any information loss for the recommendation prediction.

User ratings in the utility matrix could also be normalized, analogous to attribute values. There are various approaches for normalizing a utility matrix including subtracting the global mean from individual values, subtracting the column or row mean from the individual values, or with Z-score. Through the normalization of the utility matrix, identifying users with opposite opinions or totally different preferences is possible. Considering item ratings as n dimensional vectors in space, rating vectors of users with negative ratings for certain items would show the opposite direction compared to vectors of users with positive ratings for the same items (Lin, 2010).

2.2.5.2 Distance Measures

Distance measures are required to denote how similar two entities are. Depending on the recommendation technique, one could refer to one of the following distance measures:

Euclidian Distance

One of the simplest distance measures is the *Euclidian distance,* which describes the ordinary distance between two points in n dimensional space.

$$d(x, y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

Formula 2-1 : Euclidian similarity between item x and y

$\mathrm{d}(\mathrm{x}, \mathrm{y})$ gives the distance between item x and y with n different attributes. x_k or y_k corresponds to value of k^{th} item attribute.

Cosine Similarity

A very common approach is one in which items are represented as feature vectors in n dimensional vector space. The similarity between two recommendation items is computed based on the cosine angle that they form in the vector space.

$$\cos(x, y) = \frac{(x \cdot y)}{|[x][y]|}$$

Formula 2-2 : Cosine similarity

In information retrieval, feature vectors (keyword vectors) contain Boolean values depending on the existence of a certain keyword in the document. In this context, the dot product of vectors correspond to number of keywords in common, and the length of the vectors correspond to square roots of the number of words in each document. In real word scenarios, the feature vectors do not only contain Boolean values, and some of the item features could have higher cardinalities than 2. If the feature vector is not scaled in a proper way, higher values in the vector would dominate the calculation and the features with a lower range would be irrelevant. In other words, the scaling factor for numeric values affects the decision about how similar two items are.

Another issue with cosine similarity is interpretation of the missing values in the utility matrix. If missing values were considered as 0, this assumption would treat them as dislikes.

Jaccard Similarity

The Jaccard distance measures similarity between two items by the size of attribute value intersection divided by the union of item attributes (total number of attributes). Similarity between two users based on binary ratings could be expressed in a simple way with Jaccard distance. In this case, common rated items divided by union of users' ratings would denote the similarity between users ("Jaccard Index," 2013).

$$JD_{x,y} = \frac{x \cap y}{x \cup y}$$

Formula 2-3 : Jaccard distance between sets x and y

Hamming Distance

Hamming distance is another alternative distance measure for domains with binary values. It originates from the information theory where it denotes the number of positions that two strings with same length differ. In terms of similarity measures for recommendation systems, it could describe the number of user ratings, items, or attributes with different values.

Linear Correlation

Linear correlation is one of the most dominant distance measures due to its simplicity. The most commonly used correlation type is the *Pearson* correlation. It can be calculated given the covariance of data series x and y and their standard deviations. In the case of perfect linear relation (correlation), Pearson correlation returns +1. In the case of perfect negative linear relation it returns -1.

$$Pearson\ (x, y) = \frac{\sum(x, y)}{\sigma_x \text{ X } \sigma_y}$$

Formula 2-4 : Pearson correlation

2.2.5.3 Sampling

Whether in memory or model based recommendation approaches, processing of the whole data set is not scalable with limited time and resources. In data mining, sampling is the selection of a relevant subset from a larger data set. Sampling could be necessary during preprocessing or machine learning phases of recommendation to avoid the processing of training set entities.

The simplest sampling technique is the random sampling, wherein each item has equal probability to be selected. If items could form subgroups inside the whole data set, they could be divided into n homogeneous subgroups before random sampling. This sampling approach is called *stratified sampling*. Random sampling could be done with or without replacement, where the selected item is kept further in the data set or removed after it is selected. In order to prevent the overspecialization of the chosen data set, a random sampling could be repeated many times and the average of the learned models could be considered as final data set. This approach is called cross-validation (Pazzani & Billsus, 2007).

2.2.5.4 Reducing Dimensionality

Recommendation items could be represented as n dimensional vectors where each dimension represents an attribute. An increased number of dimensions makes the data processing and analysis difficult. The problem of high dimensionality is also known as the *curse of dimensionality*, and was introduced by Richard Bellman[10] (1961). In addition to high dimensionality problems, recommender engines should also overcome the problem of sparse information. In most of the cases, many ratings in the utility matrix are missing and most of the rows are filled with zeros (Wang, 2000).

The dimensionality reduction could be applied to overcome high dimensionality problems and sparse information. The main idea behind dimensionality reduction is to consider the utility matrix as the product of two matrices.

Principal Component Analysis

PCA is a matrix factorization tool that relies on linear combination among variables. It aims to find out certain patterns in high dimensional data sets and express them with their similarities and differences. The original data set is projected onto a new coordinate system with fewer dimensions, but despite dimension reduction the new data set still contains most of the information from the original data set. For instance, the factorization method could reduce the number of attributes to the most important 30 factors out of 100 and could also remove noise from the data set. Lindsay (2002) demonstrates how the PCA technique works in detail with a concrete example data set.

Singular Value Decomposition

The main idea behind SVD is converting a highly variable data set to a lower dimensional space that could describe the original data clearly by ordering it from the most variation to the least. It is based on the theorem of linear algebra that a rectangular matrix *M (n x m)* can be decomposed into the product of three matrices: an orthogonal matrix *U*, a diagonal matrix *S*, and the transpose of the orthogonal matrix *V*. In terms of recommender systems, the original matrix *M* represents *n* items with *m* features. The orthogonal matrix *U* has *n* items instead of *m* features with a reduced number of *r* concepts. The diagonal matrix represents the strength of each concept.

$$A_{mn} = U_{mm} S_{mn} V_{nn}^{T}$$

Formula 2-5 : Singular Value Decomposition

Baker (2005) discusses and demonstrates singular value decomposition on a demonstrative example data set.

2.2.5.5 Data Classification

Classification is a classical data mining technique based on machine learning. The goal of the classification is separation of a training set into predefined classes and groups based on their similarities. Classification methods expect as input a set of unclassified items and a set of attributes, which are considered to be class labels. Individual items of a data set are assigned to certain classes based on their similarities. During the analysis phase of the

[10] Richard Ernest Bellman (1920 - 1984), American applied mathematician

recommendation process, various classification methods could be required depending on the type and architecture of the system.

Another application area of the data classification is identification and labeling of newly added recommendation items without any user ratings or feedback. A recommendation engine could assign new inserted items into an existing class by inspecting their attribute values.

k-Nearest Neighbors

The *k-Nearest Neighbor* is an example of a memory based machine learning algorithm, which is very common in collaborative filtering. The algorithm tries to find *k closest neighbors* of an unclassified item, and if it succeeds, the new item gets the class label of its closest neighbors. It could be used to find similar users or items in terms of collaborative recommendation systems. For instance, a user X is compared with other users to find k nearest neighbors based on their past ratings. These neighbors' attitudes towards an item, which is unrated by the user X, could be considered to be a possible rating prediction. In this context, classification of a potential recommendation item denotes the rating prediction for the active user.

In addition to user similarity, it could also be used to explore newly added items. Items to be inspected are represented as feature vectors in n dimensional space. Newly inserted recommendation items could be classified as their nearest neighbors. The similarity function depends on the type of training data. For structured data sets, Euclidean distance can be used, and for unstructured data, cosine similarity function could be used (Pazzani & Billsus, 2007).

The *k-Nearest neighbor* is well known and very common due to its simplicity compared to other machine learning algorithms. In addition to its simplicity, *kNN* does not require a model to compute required classifications. This means it could easily cope with changes in rating matrix rapidly.

Decision Trees

A decision tree is a data structure where branches represent a question (choice) under predefined alternatives to an item attribute. The leaves of the tree represent the classification or decision over a certain path. Decision trees could be also used as a classification algorithm for the machine learning process of recommender systems. The main advantage of decision trees is their simplicity in classifying unknown items. However, construction of decision trees could be too costly in certain domains. Decision trees could be used to generate induction rules for user preferences. Attributes of items to be classified become the interior nodes (branches) of the tree. These attributes represent features of the given item like category, price, color, etc. The leaves refer to user feedback as positive or negative ratings (Mitchell & McGraw , 1997).

Rule-Based Classifier

Rule-based classifiers group data sets using rules in form of ***"if…then…."***. The rule consists of two parts: the rule condition and the class label that classifies the fulfilled condition. The rule condition is a conjunction of several item attributes and items satisfying the rule condition in training set could be assigned to the corresponding class label, which is considered as rule consequent.

$R1 : (Self\ service = YES) \cap (Cuisine =' American') \ \rightarrow Fast\ Food$
$R2 : (Valet\ Parking = YES) \ \cap (Prices =' High') \rightarrow Luxury\ Restauran$
$R3 : Category \neq' Vegeterian' \rightarrow Dislike$

Table 2-3 : Possible rules to classify restaurant instances

As seen in the example above, R1 and R2 are used to classify instances of restaurants while R3 could be used to eliminate all *non-vegetarian* restaurants for the active user.

The relevance of a certain rule is measured by two different means: the *rule coverage* and *rule accuracy*. *Coverage* is the fraction of items among the whole set that satisfies the rule condition. Accuracy denotes the number of items covered by the rule that belong to a given class divided by number of items covered by the rule (Tan, Steinbach, & Kumar, 2005).

$$Coverage\ (r) = \frac{n_{Rule\ condition\ fulfilled}}{n_{all}}$$

Formula 2-6: Coverage

$$Accucary\ (r) = \frac{n_{Rule\ condition\ fullfilled} \ \cap \ n_{rule\ \ implication}}{n_{rule\ concidion\ fullfilled}}$$

Formula 2-7: Accuracy

The main advantage of rule-based classifiers is that they do not require any transformation on the training data set and extracted rules are easy to interpret. Rules could be generated over other classifier methods like decision trees or neural networks, or directly from the data set itself by using a proper method like *RIPPER* (Pazzani & Billsus, 2007).

Rule based classifiers are used in the empirical part of the thesis to classify restaurants and user stereotypes for generation of the data set for the offline experiment.

(Naïve) Bayesian Classifier

The Bayesian classifier is a probabilistic modal based on *Bayes Theorem* and assumes the conditional probability of item attributes. The Naive Bayesian classifier assumes the probabilistic independence of the given item attributes. Analogous to Bayes theorem, the classifier looks for the maximum conditional probability of class labels given the values of item attributes. In other words, it computes the conditional probabilities of all class labels and assigns the unclassified item to the class with maximum class probability.

$$P(C_i \mid X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

Formula 2-8: Bayes theorem

Considering the Bayesian formula above, all classes have the same *P (X)* value, so the classifier tries to maximize $P\ (X|C_i)P\ (C_i)$. For simplifying the computation of posterior probabilities for high dimensional items, it assumes the conditional independence of individual attributes given the class labels.

$$P(\mathbf{X}|C_i) \approx \prod_{k=1}^{n} P(x_k|C_i).$$

Formula 2-9 : Posterior probabilities of attribute

x_k : $value\ of\ attribute\ with\ the\ index\ k$
$P(x_k \mid C_i)$: $frequency\ of\ items\ with\ the\ value\ x_k\ inside\ the\ class\ C_i$
n : $number\ of\ given\ item\ attributes$

If one of the item attributes has a continuous value range, it could be assumed that values have Gaussian distribution. In this case, the classifier needs to compute standard deviation and mean of attribute values for x_k (Leung, 2007).

Despite their simplicity, Naive Bayesian classifiers are considered to be an effective technique comparable with decision trees and neural network classifiers. They demonstrate robustness, high levels of accuracy, and reasonable speed when applied to large data sets with many missing values. The biggest drawback of naïve Bayesian classifier is their assumption about the independency of individual attributes. In most cases, describing item attributes might be conditional dependent. Bayesian classifiers are mainly used for building user models in content-based recommender systems. In collaborative filtering systems, this approach could be used to cope with a cold-start problem to improve recommendation performance if there is not enough rating history (Pazzani & Billsus, 2007).

Artificial Neural Networks

Artificial neural networks originate from the nervous system of living organisms, which is a huge network of many elements working parallel to and in connection with each other. Transmission of information by electrical and chemical signaling is done by nerve cells (neurons). A neuron has many inputs, while it owns only one output with a binary range. A neuron evaluates its output by continuously comparing the sum of its inputs against a threshold. Neurons are connected to each other over synapses, which allow for information transmission and assign a weight for the connection that corresponds to the strength of the connection (Gershenson, 2003).

Similar to biology, artificial neural networks are based on the same modeling assumptions, with three major components. The *synapsis* of the neuron is modeled through numeric weights, which could take positive or negative values depending on the connection. All the input values of a neuron are summed up in a *summing junction*. The final component of the artificial neuron is the *activation function*, which determines the amplitude of the output. The input parameter of the activation function is the output of the internal activity of the neuron.

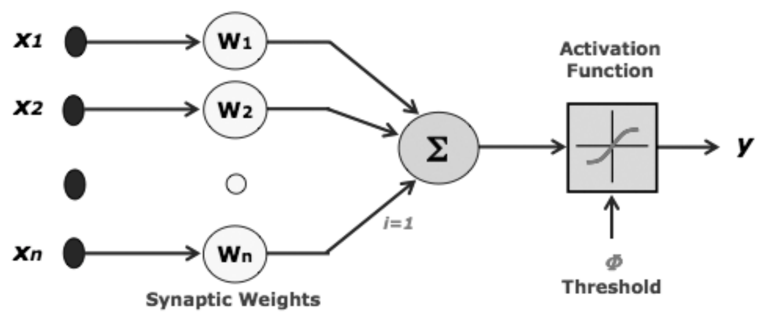

Figure 2-10 : Artificial linear neuron (Chackraborty, 2010)

The simplest form of the ANN is a linear classifier, which compares the weighted sum of inputs with a threshold value and returns 1 if it is greater than this value, and 0 otherwise. In addition to threshold function, sigmoid or hyperbolic tangent functions can be also used. A neuron receives an output signal from another neuron, assigns a weight to it, and internally computes its own output value. Finally, it forwards its output to other neurons according to the state of its activation function. These could be considered as simple processing units. In ANN there exist three different kinds of units that work parallel. Input units perceive their signal from outside of the network. Output units send in parallel their signals to the outside of the network. The hidden units are the intermediately units whose output and input signals remain inside the neural network. There are various network topologies that put the processing units together in an ANN. In *feed-forward* neural networks signals are carried from input units to output units as feed forward. This means there is no signal forwarding between a higher layer and a lower layer. In contrast, in recurrent neural networks there exists feedback connections between layers (Galkin).

Neural networks can perform tasks that linear programs cannot achieve. If one of the components of the system fails, the other components continue to work in parallel. The disadvantage of ANN is that selecting a proper network topology and learning an algorithm might be complex in huge networks.

Neural networks could be applied as classifiers for data sets with linear, non-linear, or quadratic boundaries. The perception-learning rule assumes the following procedure: In order to train a data set, n-dimensional data vectors are put as input units to network. Depending on the value of weights, they produce a set of values for the output units. After that, the desired output is compared with the output of the network. If desired level of matching does not exist, some connections are adjusted. Otherwise, the network is kept as it is. Gales (2011) explains further details of ANN on a concrete example of single layer perception as linear classifiers.

The challenge in the construction of the neural network is the determination of the network topology and assigning the corresponding weights for the units, which influence the value of the activation function.

2.2.5.6 Cluster Analysis

The cluster analysis is considered to be one of the unsupervised classifiers in data mining literature. As described above, the classification inspects a training set and matches each item to one of the prior known classes. This approach is known as supervised classification since the describing classes are known before the classification process. In contrast, the set of classes is unknown in unsupervised classification. All classes are created on wire during the classification process. The clustering algorithm assigns items that tend to be similar to the same group while separating dissimilar ones. The goal is to achieve minimum distances between items of the same class and maximize the distance between items of different classes. Clustering algorithms are differentiated into two groups, namely partitioned and hierarchical. In *partitioned clustering* algorithms, each item can reside only in one class, while in hierarchical algorithms the item could be listed under multiple classes within a hierarchical form.

k-Means

The k-Means clustering is a partitioning clustering method that is commonly used in data mining. A given training set with *n* items is separated into *k* disjoint groups so that the individual items in the subset are as close as possible to each other according to a predefined distance measure. A centroid item represents each cluster. The sum of distances from all items to the centroid item needs to be minimized in all subsets. The k-mean algorithm moves items between groups as long as the distance of the centroids to other cluster members can be minimized further. Initially the centroids are selected randomly, and after the first iteration all items are assigned to different clusters depending on the centroids. After clusters are formed, new centroid items are selected according to items in the cluster. The iteration is terminated when clusters reach a stable state so that there are no more items to change their clusters (Tryfos, 1997).

In the context of recommender systems, hierarchical k-Mean algorithm is the major clustering algorithm due to its simplicity and efficiency. It can be used to improve the performance of content-based recommender systems. The k-Mean is a very simple and basic algorithm, but it has also some pitfalls. First of all, it requires prior knowledge about the data to choose the k separate clusters. Secondly, the final shape of clusters is heavily dependent on the initial choice of the centroids. On the other hand, outliner data items could cause problems for determination of new clusters.

2.2.5.7 Association Rule Mining

Association rule mining is a well-known data mining technique that is used for shopping behavior analysis in marketing. The basket analysis associates items that customers buy frequently together. The association could be implicated by discovering occurrences of certain item patters in the purchase history. It represents how likely a customer is to buy an item *x* if another similar item *y* has already been bought. The goal of the rule mining is building rules in form, *"If user buys the item X, then he also buys with 85% probability the item Y."* Rules of this kind could be easily extracted from the set of sale transactions. An association rule is an implication with the following form, where X and Y are item sets.

$$X \rightarrow Y \; where \; X, Y \subset I \; and \; X \cap Y = \emptyset$$

Formula 2-10: Association rule

The relevance of the rule is determined once again with the terms *support* and *confidence.* The support is the frequency of the items $(X \cup Y)$ among all transactions. The confidence of a rule denotes how often transactions that contain X also contain Y.

$$support = \frac{|X \cup Y|}{|Transactions|}$$

Formula 2-11: Support of an association rule

$$\boldsymbol{confidence} = \frac{|\boldsymbol{X} \cup \boldsymbol{Y}|}{|\boldsymbol{X}|}$$

Formula 2-12 : Confidence of an association rule

A minimum support and confidence threshold could be used to filter out unimportant rules.The brute-force appraoch would list all possible association rules and filter out the ones below the support or confidence threshold. Since the computation of this approach would be too expensive, the two-step approach is usually preffered in enterprise systems.

First a set of frequent items is calcualted whose suport is over the threshold. Aftwerwards, high confidence rules are implicated from those frequent item sets (Lai & Cerpa, 2001), (Ricci et al., 2011).

2.2.6 Classification of Recommendation Techniques

The recommendation technique of a system denotes how recommendation items and their corresponding ratings are predicted for the active user. This could be considered as a probabilistic function that takes as parameter information about the items and active users and estimates a set of items that the active user might like. Mainly recommendation systems operate over collaborative filtering or content-based recommendation approaches. Other types of recommendation techniques originate from these architectures and might have additional properties, extensions, or simplifications depending on the application domain. Hybrid systems merge benefits of various techniques to generate recommendations.

Burke (2002) categorizes recommendation techniques into five groups, taking into consideration the background data, operational data, and recommendation algorithm. This section introduces major recommendation techniques, with their advantages and drawbacks.

2.2.6.1 Collaborative Recommendation

Collaborative filtering is also known as *social filtering*, and it is one of the most significant and common recommendation approaches in e-commerce. This approach fits well to very large, e-commerce sites with thousands of users and products. This technique considers the rating history of other system users who have similar preferences to the active user, and makes its predictions according to the ratings of other users. Similar preferences are implicated by comparing the rated items and the rating values of those items. CF assumes that if a user liked certain items that a group of other users also liked, unrated items of the active user could be predicted by considering the ratings of the similar users. Due to its architecture, collaborative filtering requires a huge database with as many users as possible, including their ratings of different items. The systems assumes that user preferences do not change over time because the system cannot react to short-term preference changes.

Collaborative filtering has two major sub-categories, namely user-based and item-based CF. User-based CF was introduced by the *GroupLens*[11] system in 1994, and predictions about items depend on the opinions of similar users. The second approach was introduced in 2001 and is known as item-item CF. It is based on similarities between items. It assumes that users tend to have the same opinions of similar items. The similarity between items is not calculated by comparing their content. Rather, it depends on the ratings of users. Finding similarity among recommendation items is easier than finding

[11] GroupLens is a research lab in the Department of Computer Science and Engineering at the University of Minnesota

similarities between users. For instance, a certain restaurant could have only one category while a specific user could like more than one restaurant category. Due to this consideration, the item-based approach is preferred in many CF applications. However, once the similar users of the active user is known, all the missing item ratings can be predicted based on the set of these users. In contrast, if the missing values are predicted over similar items, similar items need to be calculated over and over again for each item (Anand & Jeffrey, 2011).

User-User CF

This approach assumes that the missing rating for an item *i* of the active user *X* could be estimated over similar users who have rated this item before. The first step is finding a set of users like *X* who liked the same item. Secondly, the set of users is reduced to users who have also rated the desired item *I*, whose rating is missing for the active user. If the utility matrix provides very high similarity values between two users *X* and *Y* who both have not rated the investigated item *i*, it might be notable to estimate the missing ratings for the near neighbor *Y* first. The main idea of this approach is to benefit from correlated users recursively so that their missing ratings are not calculated over more distant neighbors.

After finding necessary users, the recommendation algorithm requires a distance measure to describe the similarly between users. Pearson correlation is a very popular similarity measure to describe similarities among users in collaborative recommender systems. The advantage of Pearson correlation is that the deviations in the user ratings are also considered. The similarity between active users *i* and *k* could be expressed with Pearson correlation with the following formula based on their ratings, given that *j* item ratings are in common.

$$u_{ik} = \frac{\sum_1^j (v_{ij} - v_i)(v_{kj} - v_k)}{\sqrt{\sum_1^j (v_{ij} - v_i)^2} \sum_1^j (v_{kj} - v_k)^2}$$

$v_{i,k}$ $average\ rating\ of\ user$
$v_{i,k\ j}$ $rating\ of\ user$ on item j

Formula 2-13: Similarity between users based on Pearson correlation

After finding similarity values of other users to the active user, missing ratings could be predicted with the following formula:

$$v^*_{ij} = v_i + K \sum_{v_{kj} \neq ?} u_{ik} (v_{kj} - v_k)$$

Formula 2-14 : Rating prediction for item j of user i

K is a normalization factor of the similarity so that the sum makes exactly 1. Alternatively, cosine similarity could also be used to compute the similarities between users (Ricci et al., 2011).

The rating prediction formula takes all neighbor ratings into calculation, even if they have negative correlation values. In statistics, negative correlation denotes the opposite correlation of data sets. For the sake of reliable predictions, not all the neighbors should be equally valuable. A possible solution to cope with this problem is assigning weights to neighbors depending on their similarity value to the active user. Neighbors with similarity value close to 1 should have higher weights compared with others. An easier approach could be defining a threshold and eliminating all neighbors under this threshold value. In the case of a utility matrix where many neighbors have large similarity distances, only top n similar neighbors could be taken into calculation to simplify the computation. Another issue with finding similarity between users is the total number of common rated items. Users with more items in common with slightly lower similarity values are more significant compared to users with less items in common but higher similarity values. Users with less common rated items could have higher similarities as a result of similar rating values, but number of common items could be used as a significance factor. A possible significance value needs to increase analogous to number of common rated items to elevate the similarity of users. Another drawback of rating values is that commonly positive rated items tend to be less informative than controversial items. Item ratings with higher variance should get higher weights since they indicate more evidence for the prediction (Zanker & Jannach, 2010).

Item-Item CF

Instead of matching similar users with similar preferences, the item-item CF approach matches active users' rated items to similar items. Similarities among items are likely to be more stable than similarity among users. This approach requires a smaller similarity matrix compared to user-user CF since it considers only items rated by the user. Compared to Pearson correlation, cosine similarity provides better results in case of item-to-item filtering (Zanker & Jannach, 2010).

For this approach, recommendation items are represented as n dimensional vectors, where dimensions correspond to users who have rated these items. This approach compares similarities of these vectors and takes the most similar k items into consideration. The prediction for the missing item rating is then computed by taking the weighted average of these similar k items (Sarwar et al., 2001).

Prediction of an item i for the user u could be computed by calculating the sum of all items similar to i. Each rating is weighted by its similarity value to i $(s_{i,N})$.

$$P_{u,i} = \frac{\sum_{all\ similar\ items}(S_{i,N} * R_{u,N})}{\sum_{all\ similar\ items}(|S_{i,N}|)}$$

Formula 2-15 : Rating prediction for item-item CF (Sarwar, Karypis, & Konstan, 2001)

2.2.6.1.1 Advantages of CF

Collaborative filtering is a very common recommendation technique and could be applied in many domains where content awareness is not required. This approach is easy to implement and does not require any knowledge engineering.

2.2.6.1.2 Drawbacks of Collaborative Filtering

Cold Start Problem:

As mentioned above, collaborative filtering could operate with similarity measures between system users or items. In order to get consistent and reliable results there needs to be enough users with many ratings in the system. If initially there are not enough users in the database, the system could fail to find a reasonable match or the recommendation output might be inconsistent. As a simple walkthrough for this problem, the system could ask users to rate a small set of items initially so that the system collects information about users before computing recommendations (MovieLens)[12].

In real world sceneries, the collaborative filtering approach is assisted by alternative methods including content-based or knowledge-based systems until the newly registered user has enough feedback stored in the system.

Sparsity Problem:

Utility matrices are essentials of collaborative filtering systems. Generally, most of the users are only interested in a very small set of items. Apart from that, users do not always tend to rate all items that they are interested in. A sparsity problem arises if there are many missing values in the utility matrix. In huge systems there are million of users and items, and despite a high number of ratings, it is not always possible to find users that have rated exactly the same items. To cope with this problem, missing ratings could be replaced by their estimations considering the implicit user interactions with the systems. Implicit ratings assume that page views, number of visits, and time spent on a certain page could be interpreted as an implicit positive user rating. However interpretation of these implicit feedbacks are not always sustainable. Another approach to handle a sparse dataset is the transitivity of neighbors. If the recommender system already knows a very close neighbor and one of them has a missing rating for a certain item, the rating of the close neighbor could be used as a predicted rating to fill the missing rating of the active

[12] www.movielens.org is a movie recommendation site based on collaborative filtering. (Last visit 10.11.2012)

user. This approach would not only solve the problem of data sparsity, but also makes computation overload lighter for users whose very close neighbors are already known.

The sparsity problem is a general issue for nearly all collaborative filtering systems, and in real world scenarios the utility matrix is transformed by matrix factorization methods as mentioned in 2.2.5.4 before computing recommendations.

First Rater Problem:

The CF assumes that only items with at least one rating could be taken into the recommendation step. Newly inserted and outliner items without any ratings cannot be recommended. With the help of clustering or classifier algorithms, coverage of the system could be extended with the items or users without any clue to ratings.

Popularity Bias Problem:

The CF generalizes user interests and relies on prototyping of user preferences. Users with extraordinary preferences and unique interests do not get any novel and consistent recommendations since the system tends to find only popular items.

Scalability Problem:

User-based collaborative filtering is considered to be *memory-based* due to its computation approach of similarity matrices. Generation of a similarity matrix is not achievable for systems with millions of users and hundreds of items. *k-NN* is a widely used classification algorithm to find similar items or users. As described above, *k-NN* is a memory-based approach, and as the number of active users and products increases, recommender systems could face scalability problems due to huge utility matrices. For creation of similarity matrices, the recommendation algorithm could only consider a sampled training set rather than iterating the whole data set (Section 2.2.5.3).

In real world scenarios, model based approaches are preferred compared to memory-based approaches. Model based approaches are based on mathematical or probabilistic models. During recommendation generation, the corresponding model is used to make the predictions and these models are updated periodically.

Black Box Recommendations:

Similarities between users or items are calculated over user ratings and recommendation results cannot be reasoned once the recommendation is generated.

No External Information Source Integration:

Due to absence of content awareness, it is not possible to use any external information source about the recommendation items.

Multi-Criteria Ratings:

Similarities between items or users depend only on user ratings, but in some domains it is necessary to evaluate the items with multi-criteria. This might result in totally different similarities due to varying ratings for different criteria. Considering this pitfall, CF fits well with domains where items are evaluated with single criteria ratings.

2.2.6.2 Content-Based Recommender Systems

Content-based recommender systems originate from traditional information retrieval methods. The goal of an information retrieval system is finding out relevant text documents with desired keywords and listing them according the their relevance ranking. In the case of information retrieval systems, items to be recommended are textual documents. Due to unstructured form of text documents, they require a preprocessing step where the structured relevant information is extracted from the documents. The first step is considered to be a content analysis step of recommendation step, and the required technique depends on the domain of items. The purpose is preparing a structured item data to be proceeded in the analysis phase. If the information source of items already provides structured data about the items, this step could only involve reducing the dimensionality of the feature vector or normalizing value ranges for simpler calculation.

Apart from textual documents, content-based recommendation systems try to recommend items similar to those a given user liked in the past. Content-based recommendation could be preferable for recommending web sites, news, scientific articles, and retail products, as well as services, where content analysis is significant for the recommendation generation. This approach focuses on the attribute values of items rather than considering opinions of the other system users. As mentioned above, collaborative recommender systems consider user ratings during recommendation generation while any significant information about the content of items is ignored. CBRS calculates similarities between items by considering their attribute values.

CBRS requires two sets of information for its operation: primarily, a set of items with describing attributes, and secondly, users' profile information that denotes user preferences. CBSR analyses items with their features that users rated before and builds the corresponding user model based on this information. This model corresponds to user preferences, or in other words interests. When the required user preference model is created once, the recommendation algorithm can compare potential recommendation items with preference values and assign a distance value to each item. Depending on the relevance values, items with less distance to user preference could be shown to the user in a ranked list. In this step, the user could optionally provide additional feedback about the relevance of recommendations so the existing user model could be updated and extended optionally. This procedure could be considered as a learning cycle for the user profile. Through this machine learning cycle, the recommender system could cope with changing user preferences and refine its recommendation accuracy.

2.2.6.2.1 Item Representation in Content-Based Recommender Systems

Each CBRS requires an *item profile* where item specific information is stored as feature sets. This information could be saved as an item *feature matrix,* where each item is represented as n-dimensional feature vector that corresponds to a row in the matrix. This vector describes the key features and properties of an item and depends on the type and domain of items. For instance, textual items including news, emails, or web pages are represented by various keywords while an electronic device could have different attribute types to describe its physical and technical properties. This representation technique is known as *Vector Space Model.*

	c_1	…	c_m
a_1	$v_{1,1}$	…	$v_{1,m}$
…	…	…	…
a_i	$v_{i,1}$	…	$v_{i,m}$

a_i *item with index* i
c_i *feature* i
v_{ij} *value of attribute* j *for item* i

Table 2-4: Item feature matrix

The same representation technique could also be applied to user queries where the interested terms make up the query vector. If a certain keyword from the document collection is to be found in a document, its value in vector representation gets a non-zero value, and otherwise zero. Representation of keywords as vectors brings a challenge since not all the keywords have the same significance for the document. For instance, stop words tend to appear multiple times in documents even though they do not reflect the topic of the text. Another important issue is the length of the document, where a keyword might appear multiple times in a long document even though the subject of the text is about something else. From this perspective, longer documents would have higher probabilities than shorter documents to contain the relevant keywords. To overcome these problems in textual documents in vector space representation, a numeric value is assigned to each keyword that represents its weighted relevance. This value is called weighting factor *tf*idf* (*term-frequency times inverse document frequency)* (Pazzani & Billsus, 2007).

$$w(t,d) = \frac{tf_{t,d} \log\left(\frac{N}{df_t}\right)}{\sqrt{\sum_i (tf_{t_i,d})^2 \log\left(\frac{N}{df_{t_i}}\right)^2}}$$

Formula 2-16: tf*idf weight or w(t,d) of term in d13

Since documents and queries could be formalized as vectors, their similarity could be calculated based on cosine similarity. The feature vector of keywords in IR is comparable to feature vector of recommendation items in other application areas of CBRSs. Additionally, query vectors could be compared with the user profiles to designate the interests of the active user. Analogous to keywords of textual documents, users do not consider all of the item attributes, and some users could ignore certain item features. In this context, weighting of item features is a variable parameter depending on the active user. As mentioned above, IR assumes that feature vectors of documents contain only binary values to denote existence of a keyword in a document. In other recommendation domains, item attributes could have higher cardinalities like numeric, nominal, and ordinal values.

2.2.6.2.2 User Profiles in Content-Based Recommender Systems

In addition to item feature profiles, a CBRS also requires information about user preferences and interests. The main purpose of user profiles is expressing overall interests of the active user about items that he liked and rated in the past. One of the biggest challenges of content-based recommender systems is building the required user profile, which is a long-term process. It requires active user participation and interaction. In order to build required user profiles, the system could consider explicit or implicit feedback or ask the user to provide it initially before starting recommendation generation for the new user. User profiles could be computed in various ways, and optionally, preference weights could be calculated considering the utility and item feature matrix. Based on *TF-IDF*, feature weighing could be calculated as following :

$$W(u, c_j) = FF(u, c_j) * IUF(c_j)$$

Equation 2-1 : Feature weight calculation with TF-IDF

FF(u,c_j*)* corresponds to number of times that the active user rated an item with the feature value c_j. The second factor *IUF(*c_j*)* could be calculated as *Log(|U|/UF(*c_j*))* where U is the

[13] *(*tf_t*,d)* is the frequency of term t in document d.
N is the number of documents in the collection.
df_t is number of documents that contain the term t.

total number of users in the system and $F(c_j)$ is the number of users that liked any item with the feature c_j. Calculation of feature weights over TF-IDF is only relevant when the value range of features is binary. Most recommendation domains require attributes with higher cardinality values. Considering that variability *tf*idf* weight is not a proper measure to reflect significance of an item feature, Martinez and Barranco (2010) propose an alternative method to calculate the weight of multi-valued item features by considering the amount of information they contain and correlation between user ratings and feature values of rated items. They assume features with higher entropy or features with higher correlation to user preference values would get higher significance weights. The mentioned approach is described in detail in section 4.5.

	c_1	…	c_m	R_u
a_1	$v_{1,1}$	…	$v_{1,m}$	$R(a_1)$
…	…	…	…	…
a_i	$v_{i,1}$	…	$v_{i,m}$	$R(a_m)$
P_u	p_1	…	p_m	
W_u	w_1	…	w_m	

a_i *item with index* i
c_i *feature i*
v_{ij} *value of attribute j for item i*
$R(a_i)$ *user rating on item* a_i
p_i *preference value on feature i*
w_i *preference rating value on feature i*

Table 2-5 User preferences model

2.2.6.2.3 Advantages Of Content-Based Recommender Systems

User Independency

Content-based recommender systems do not require rating data of other users to calculate similarities between items. Each user is treated independently and recommendation results are not bound by popularity bias.

Transparency of Recommendations

The biggest advantage of content-based systems in contrast to collaborative filtering is that they can analyze the description of items and user preferences and thus can reason recommendation results. This means recommendations are not presented as a "black box." The recommendation algorithm can associate recommendation results with a list of the item features, weight, constraints that made an item relevant for the recommendation result set.

Unique User Interests and New Items

In contrast to collaborative filtering, *unpopular* or new items could also be recommended as long as their feature vector shows similarity with the user model. This means CBRS does not suffer from the first rater problem. On the other hand, users have the opportunity to define their preferences explicitly independent of other users. Outliner users with unique tastes can also get recommendations for their demands. Outliner user preferences remain under coverage of the recommendation model.

2.2.6.2.4 Pitfalls of Content-Based Recommender Systems

Limited Content Analysis

CBRSs need to analyze the content of items to find relevant item attributes and compare them with user models to reason potential interesting items. If it fails to find enough evidence to discriminate potential items, it could fail to generate a recommendation.

Recommendation items have certain features, which could be encoded as keywords. This approach works perfectly for items with textual content, but images or videos are hard to represent with a few keywords. On the other hand, if different items are encoded with the same keywords, it is not possible to distinguish them. Due to the natural complex form of languages, assigning keywords to items as features is mostly insufficient since keywords might cause inconsistency in the case of synonyms or polysemy. Therefore, a proper semantic analysis and personalization is necessary for more advanced systems in complex domains. For instance, meaning of words could be compared over an external information source like *WordNet.*[14].

CBRSs could also take advantages of web 2.0, including social tagging, to overcome the limited content analysis problem. In literature, these recommender systems are considered to be *social tagging recommender systems*. Social tagging systems model user preferences as tag vectors. A very interesting and good example of such a system is the website *Stackoverflow*[15]*,* which features questions and answers to wide range of topics in computer science as an open forum. Registered users assign relevance tags to their questions, which are stored in their profiles. Other forum users can search for open questions with their favorite tags and answer these questions. Each time a user asks or answers a question, his tag vector is updated with the actual number of interactions that he has done with a certain tag.

New User

CBRSs need a complete user profile where user preferences and interests are held. The system can only become capable of making accurate recommendations if the active user has provided enough feedback.

14 Lexicon database for English.
[15] www.stackoverflow.com

Over-Specification (Serendipity Problem)

Recommendation results have two aspects, namely novelty and serendipity. A serendipitous recommendation system helps users to find items that they might not have discovered without the recommendation system. In contrast, a novel recommendation system suggests items that users could have also discovered without system assistance (Herlocker et al., 2004).

A CBRS suggests items whose attributes are similar to preference values of the user profile. This means the system can only recommend items that are similar to ones that the active user has already liked before. The method of content-based recommender systems cannot recommended an unexpected item. This limitation is known as the serendipity problem. CBRSs operate with a limited degree of diversity and always provide novel recommendations (Pazzani & Billsus, 2007).

In order to overcome this problem, content-based recommender systems need to be open to a certain level of randomness, or recommendation results could be extended with other recommendation techniques like collaborative filtering.

2.2.6.3 Knowledge-Based Recommender Systems

Knowledge-based recommender systems rely on their knowledge base about users and products to generate recommendations. Based on their knowledge base, they can reason what products meet the user's requirements (Burke, 2000). KBRS could be considered to be a type of content-based recommender system, which operates without any need for user profiles. User requirements are captured on wire during recommendation generation as conversational user interactions. Users could, for instance, evaluate presented recommendations and, based on given user feedbacks, recommendations of the next iteration could be refined. KBRSs are capable of reacting to changing user needs in the short term. Apart from that, They do not require any user rating or information and do not suffer from cold start problem. Knowledge-based systems have the advantage that individual recommendations could be explained to users. They tend to provide better recommendations at the beginning of deployment compared to CF and CBRSs. The biggest challenge in KBRS is that they require knowledge engineering for their implementation. Additionally, their suggestion ability is static and bounded with the present knowledge base. Ricci et al. (2011) distinguish two different kinds of knowledge-based recommender systems. Constraint-based recommendation systems are based on certain constraints as association rules to relate predefined items with user requirements. Case-based recommendation systems operate on similarity metrics like content-based recommendation systems.

2.2.6.4 Utility-Based Recommender Systems

Utility-based recommender systems are similar to KBRS and do not attempt to build long-term user preferences. User preferences are replaced with a utility function that tries to match user needs with available options based on item features like price, quality, location, or delivery date. The system computes utility of available items over weighted features and generates the recommendation based on these utility values. UBRSs are not very flexible and require more user interaction than KBRSs since active users need to construct a preference function explicitly (Burke, 2002).

2.2.6.5 Demographic Recommender Systems

Demographic user information including age, gender, education, location, occupation, and marital status could be used to create certain stereotypes among users. Demographic recommender systems assume that users that fall into the same demographic groups share the same preferences. Primarily the system needs to classify users based on their demographic data and match items with them considering their demographic information rather than the user's past interactions. They can be considered as a sub-group of collaborative recommender systems that operate with user attributes to find similarities between users. They operate in a simple manner compared to CF and CBRS, but gathering required demographic information from users is not always straightforward. Social networks could be used as external information sources for that purpose. Personalization of the recommendation results is limited by the dimension of the stereotypes. Outliners in the stereotype class could mislead the preferences of other users (Anderson, 2011).

2.2.6.6 Community (Social Network)-Based Systems

Community-based recommendation systems could be considered as another sub-category of collaborative filtering. Instead of associating similar users or items over user ratings, they are based on the relationships of the active user with other users in a social community, and consider preferences of user's friends for the recommendation process. People tend to trust people who are close to them more than anonymous users. Community-based recommendation systems assume that the active user has similar preferences with his friends in social networks. In many cases, friendship in a social network does not necessarily indicate that two users have similar tastes. This drawback could be overcome by considering demographic similarities of friends of like age or location. Jianming and Wesley (2010) propose a social network-based recommendation approach that refines users own preferences based on his social network friends' ratings. This approach can be considered to be a CF technique that does not require similarity computation for the active user. They also investigate the fact that social relationships of users influence the acceptance rate of items as a network effect.

2.2.6.7 Hybrid Recommender System

Hybrid recommender systems aim to take advantage of various techniques and combine them to alleviate drawbacks of individual techniques. A common approach is the combination of collaborative filtering with some other techniques to avoid the *new user* and *new item* problems. Burke (2002) introduces various hybridization approaches as shown in the following table.

Hybridization Method	Description
Weighted	Scores of several recommendation techniques are combined together to produce a single recommendation.
Switching	The system switches between different techniques depending on the current situation.
Mixed	Recommendations of different techniques are presented at the same time.
Feature combination	Features from different recommendation data sources are merged for a single algorithm.
Cascade	The output of a recommendation technique is refined by another technique.
Feature augmentation	Results of a recommendation technique are used as input for another technique.
Meta-level	User model learnt by a recommendation system is used as input for another model.

Table 2-6 : Comparison of hybridization methods (Burke, 2002)

2.2.6.8 Mobile Recommender Systems

Mobile recommender systems could increase the usability of applications that run on smart phones and tablets. Travel and tourism applications are important application areas for mobile recommendation systems. Due to the mobile usage environment, these systems show certain drawbacks and advantages compared to systems designed to operate on personal computers. In many applications, the position of the system user is considered as an important source of information that determines recommendation generation and recommendation presentation.

Mobile portable systems offer their users mobility and wireless connectivity so that they have access to the same information in different locations. On the other hand, due to the physical characteristics of mobile devices, users need to browse the information on smaller screens, which makes the style of recommendation presentation even more important. Apart from that, mobile devices have certain limitations relating to supply power, data storage, and user interaction. As a consequence, traditional recommendation techniques cannot be applied to mobile recommendation systems directly. For instance, mobile devices have traditional peripherals including keyboards and a mouse as touch screens or in the form of physical dial pad with less control keys, so the user interaction becomes less convenient compared to personal computers.

A user would need to scroll down multiple times to discover an item that is listed in the lower parts of a ranked list. Most of the location-aware travel applications show possible relevant attractions including restaurants, museums, or hotels alternatively on a map-based interface to overcome this drawback. In addition to location awareness, mobile devices could also offer proactive recommendation techniques with the help of certain sensors. For instance, these sensors could detect the biometric data of the human body including heart rate and skin temperature and regulate their outputs. Running and cycling applications are good examples of this.

Route recommendation is another important application area of mobile recommendation systems. In addition to route recommendation with different preferences like fastest or shortest route, such applications also provide information about the timetable of public transport, traffic situations, and accidents (Ricci, 2010).

2.3 Service Recommendation

The first two parts of the state of art investigate the service ontology and recommendation systems. Intangibility, heterogeneity, inseparability, and perishability of services make them different from goods in terms of marketing and quality evaluation of service customers. Furthermore, due to these characteristics, recommender systems in service domains could have additional roles for service customers.

As mentioned in section 2.1.4, service marketing needs to be customer centric considering his active involvement in the service delivery. In addition to customer involvement, performance of the service employee can also influence overall service quality perception. Due to the process nature of services, service customers experience various service encounters that could elevate or decrease customer satisfaction. All these service properties require explicit consideration in terms of service recommendations, and assign additional roles to recommender systems that help service customers during their decision making process. This section investigates the state of art in service recommendation, and the properties that make service recommendations different than goods.

2.3.1 Role of Tangibles in Service Recommendation

One of the most important features of services is their intangibility. Due to intangibility of services, service customers cannot experience the service output physically. In other words, service customers cannot touch, see, or taste the purchased service (in contrast to goods). However, service customers have the opportunity to experience multiple touch points where they interact with different physical components. As shown in the figure above, level of tangibility depends heavily on the service domain. Financial services, consulting, or teaching can be considered relatively less tangible than services like car rental, heath care, or restaurants. The number and role of required tangible components influence directly service quality perception as mentioned in service quality models. In general terms, tangible components could give service customers a first impression about the service provider and determine their zone of tolerance by shaping customer expectations. For instance, a guest of a five-star hotel expects to see clean sheets and soft, hygienic towels in his room. In contrast, customers of a cheaper hotel have less expectations in terms of service delivery and service touch points.

Service providers try to impress their customers with various service touch points. Tangibles can be considered the vitrine of service providers, and they are relatively easier to change than the form of the service process. For highly tangible service domains, type and number of tangible components can be used as an important competitive factor. For instance, luxury hotels can have additional services like thermal pools, saunas and sport halls for their customers.

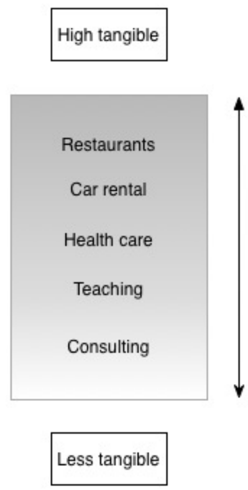

Figure 2-11 : Tangibility spectrum for some service domains

Analogous to service marketing, recommender systems could consider tangible components to be an important source of information, or as parameters for their utility function. Service entities can be described by varying tangible components as service attributes. In service domains where the level of tangibility is relatively high, tangible components can be significant decision factors for the users. *rentalcars* [16]is an online pull-based recommendation system that queries multiple car rental service providers for selected vehicle pick-up location and dates. Initially, items are listed in ascending order based on their rental price. The user could refine search results by providing further information about desired car type, supplier location, and fuel and transmission options. The system is a typical example of a utility-based recommendation engine. It uses multiple tangible attributes of car rental providers to utilize customer needs.

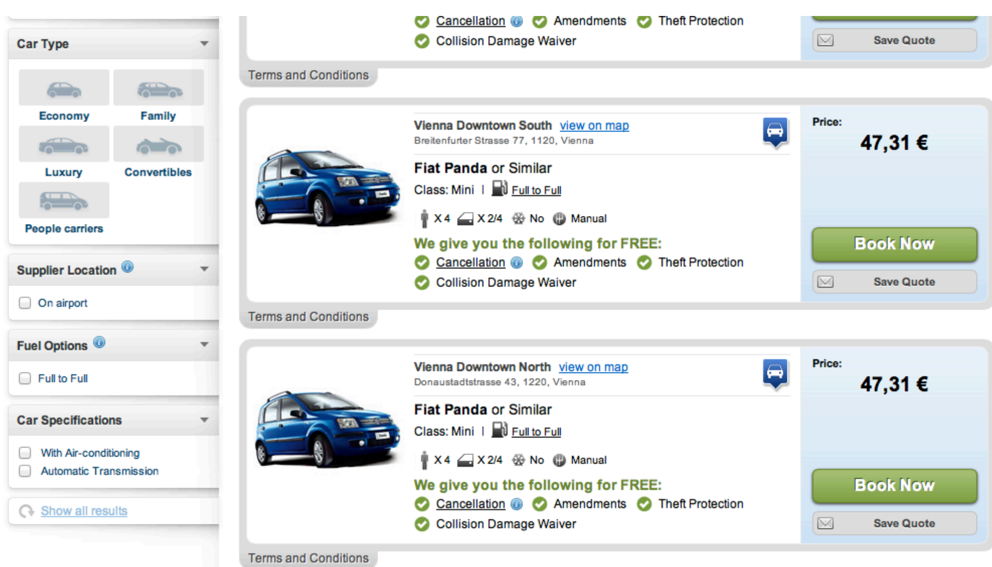

Figure 2-12 : Recommendation presentation of *rentalcars*

[16] www.rentalcars.com Last visited on 23.04.2013

Tourism and travel can be considered the most important application domain of service recommender systems. For instance, recommender systems utilized to recommend city attractions including museums, archeological sites, and other important venues could provide personalized recommendation to their users with supporting textual information or images. More advanced systems could even provide augmented reality views, 3D images, sound, and videos. Choosing a proper travel destination depends on various personal factors and travel features. The destination decision of the traveler could depend on multiple socio-economic and demographic factors including age, income, and place of departure. Additionally, travel features like travel purpose, duration, and type of transportation could also play a significant role in the decision. Considering the high number of input parameters for destination decisions, holiday booking can use recommendation systems to assist travelers during their decision making process.

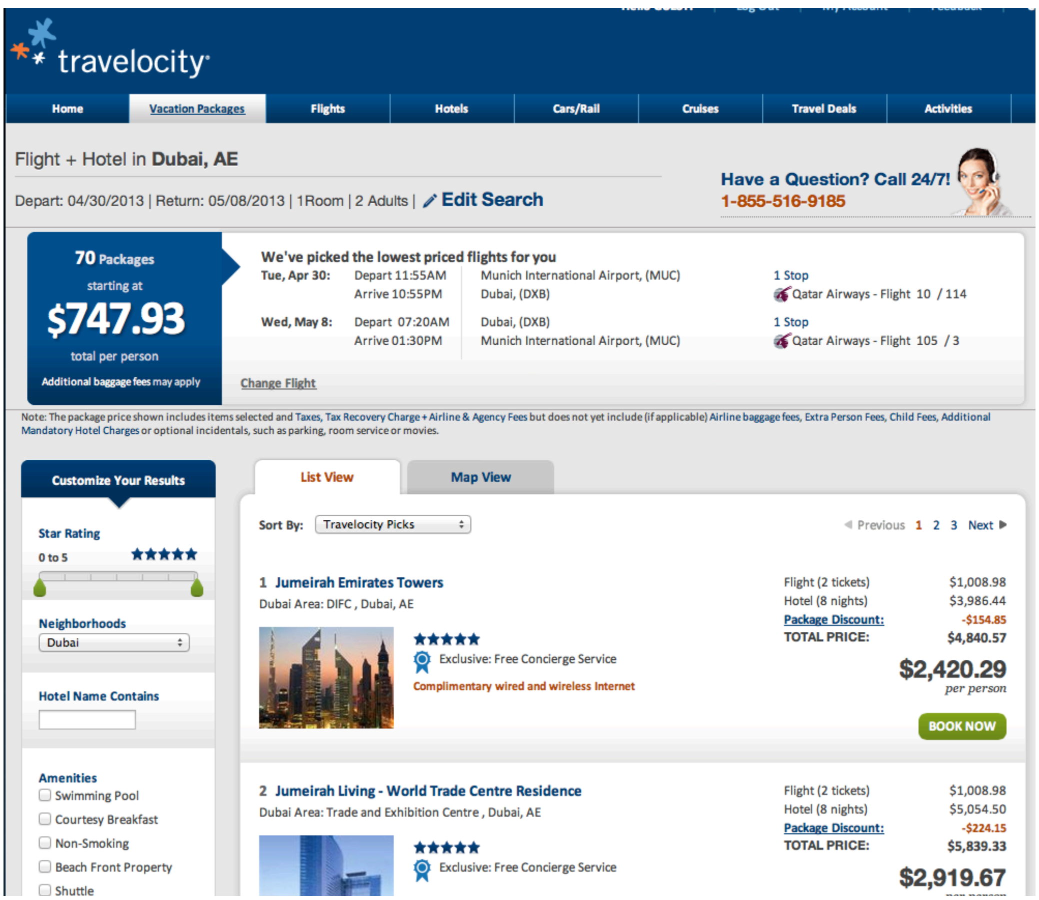

Figure 2-13 : Travelocity mimics functions of traditional travel agencies

Travelocity is an online platform that mimics all possible interactions that can be experienced in a travel agency. Users can get recommendations for their holidays analogous to counseling sessions at an agency. The user needs to choose the type of his desired holiday, and the system shows a wide range of travel and accommodation options that can be sorted or filtered based on his preferences. Recommended hotels include customer ratings in different dimensions including bed comfort, cleanliness, room quality, staff and service, and value for money. In addition, former guests can also provide textual reviews about their experience with the hotel. Textual customer comments are very common in the travel and dining recommendation domain and can be considered an important requirement. This requirement comes from the heterogeneous nature of services. Additionally, due to customer involvement in the service delivery, two travelers do not experience the same service even if they have booked the same vacation package and visited the same destinations. As a consequence, numeric ratings require further explanations to denote perceived customer experience.

As an addition to travel and accommodation recommendation, the system also gives advice about activities and available events for the desired destination. Registered users can set their travel preferences, including top preferred airlines, car rental providers, and hotel chains. Additionally, users can even set constraints about meal or seating preference in the airplane.

As seen in the examples above, tangible components of car rental and accommodation services can be taken into the utility function of service recommendation systems. Tangible components of services can be decision factors for service customers if they have active roles in the service process. Recommendation systems can model these tangible components as service specification attributes and generate their output with their values.

2.3.2 Services Cannot be Refunded

One of the major differences between services and goods is that purchased goods can be refunded, while this is not possible for services. Customers can return purchased goods in a given time without necessarily claiming a reason. In the case of e-shops, goods cannot be observed with detail before delivery. Due to this limitation, it is very common that e-shop customers will need to return purchased products. On the other hand, due to the perishable and inseparable nature of services, it is not possible for service customers to refund purchased services. The term “inseparability” denotes services that are produced and delivered at the same time. Furthermore, a service customer can only experience different dimensions of the service as he consumes it. A service customer perceives the service value while the service is perished. For example, a hotel guest cannot demand to return the accommodation service if he feels that his requirements have not been fulfilled. During his stay in the hotel, a service customer occupies a room, which could have been used by another hotel guest. Additionally, he participates in a set of activities with certain tangible components that can be consumed physically like food. Due to the inseparable and perishable nature of all these activities, it is not possible to refund the stay in the hotel. In contrast, purchasing a good transfers the ownership of the good from its distributer to the customer. As long as the purchased good is in acceptable condition, the customer can return it back to its seller within a defined period of time. The same good can be sold to another customer by the seller if its conditions are still acceptable.

Considering that services are not refundable and intangible, the decision making process of service customers is even more complicated than good purchases. A service customer makes his decision based on the service provider's promises that correspond to service specifications. The reliability of the service provider and its promises cannot be estimated over service specification. Due to that, a service customer can consider company image or reputation, or what he perceives through word-of-mouth communication. Additionally, he can also evaluate the tangible components, which could be pre-evaluated before service purchase. Recommender systems can help service customers to gain an overall impression about the reliability of a service provider. Tangible components of service providers can be presented to active users as images in certain service domains including tourism, accommodation, and restaurants. This would ensure the reliability of the service provider and increase the trust in the recommendation system.

2.3.3 Customer Involvement

Due to the heterogeneous nature of services, each service entity needs to be considered unique. There are different factors that make service providers and individual service deliveries variable, including location, unique tangible components, active customers and employee participation. Service employees can elevate customer satisfaction or decrease provided service quality. Additionally, the service customer needs to participate in certain activities during the service delivery. Therefore, in addition to the service employee, the role of the service customer is also very critical. For instance, a car mechanic needs to find out possible sources of problems by diagnosing the car, but the service customer also needs to describe where, when, and how the problem occurs. In this context, the technical knowledge of the service customer is required to determine the task specification.

In terms of service recommendation, customer involvement needs to be considered as an input parameter if it can result in a deterministic change in the service output. The car rental service is a good example where customer involvement is relatively high. The customer experiences various service encounters as he books the rental car, takes it from the pick-up location, and returns it back. Additionally, overall service satisfaction depends partly on how well the service customer can drive a car. If the service customer experiences difficulties or has negative impressions of the car due to his driving abilities, this will probably result in a negative overall service evaluation, even though it is not related to the service provider. A possible car rental recommendation engine could use as input parameter driving skills of active user to rank its recommendation results.

transportdirect is a journey planner and transportation recommendation system that can advise users about various transportations modes over different routes to desired destinations. This system compares possible journey options between two destinations based on number of transfers, journey duration, total distance, and carbon monoxide emission of the travel. Additionally, the user can compare costs of different journey options. Some of the suggestions also include partly walking routes, where the estimated route duration depends heavily on user involvement. The journey planner lets the active user set his average walking speed to get more accurate duration estimates for routes where they need to go by foot. Apart from route duration estimation, users can also get alternative route recommendations if their walking speed can compensate for the waiting time of a certain vehicle.

Moloskiing is another interesting recommender system for ski mountaineering. In this system, ski mountaineers provide information about their trips and level of trust to other system users. Ski mountaineering can be considered a risky extreme sport, and the experience and physical durability of the mountaineer is significant for success. The goal of the system is to make ski mountaineering trips safer by exploiting ski route conditions provided by other users. Based on this given information, the system only shows relevant and reliable information to the active user. Moloskiing models a trust network for each user and presents only routes considered secure and enjoyable by trusted users (Avesani et al., 2004).

Health care has also been an important application domain for various recommendation systems. Duan et al. (2011) proposes a recommender system based on previously given diagnoses of nurses to construct care plans for hospital patients. The main purpose of the system is to assist nurses in finding out if a patient requires a certain medical item on behalf of his therapy. In addition to common measures of confidence and support, this system also refers to the effectiveness of a novel measurement for the next iterations. Initially the nurse responsible for the patient selects required items explicitly based on the primary diagnosis and the system determines the required items as a ranked list throughout the therapy of the patient. The accuracy of presented items increases sequentially as nurses modify the ranked list of presented items and log health condition of the patient. In the long term, suggested care plan items are based on previously picked items.

Khan and Hoffmann (2003) propose a case-based diet recommendation system that assists doctors in creating menus for their clients. The knowledge base of the system contains the nutrient requirements of patients for better recovery from diseases or surgery. Analogous to the care plan recommendation system mentioned above, the doctor can update reasoning and the knowledge base of the system if it performs an unsatisfactory recommendation in the current iteration. Whenever the system is updated with explicit input, the doctor also needs to provide an explanation for the update. Through this approach, the system shows incremental improvements by reasoning the updates of its knowledge base.

The health care domain could be considered relatively more critical for service recommendation. Systems need to consider all possible cases, and considering the value of human life, no fault can be tolerated. Due to that, explicit expert feedback can be necessary to correct system recommendations. In this healthcare domain, patients' heath conditions can be considered as passive customer involvement.

3 Generic model for service recommendation

In the previous section, possible recommendation techniques have been discussed in detail, with their advantages and limitations. This section investigates requirements of a generic service recommendation model independent of the service domain. The requirement analysis investigates a proper recommendation technique, structure of ratings, and user and item entities separately.

As mentioned earlier, user, item, and rating entities need to be modeled considering the domain-specific properties of the system, and their structure can also vary depending on the recommendation technique. The proposed generic model focuses on the common features and dimensions of different service domains. As demonstrated later in section 4, application of the proposed model on a concrete service domain requires additional extensions and certain customization.

3.1 Recommendation Technique

The first important requirement of a service recommendation model is the individual treatment of service customers. It should be assumed that each service customer can have different preferences that can change in the short or long term. In real world scenarios, collaborative filtering is preferred due to its simplicity, but the popularity bias problem of CF violates the uniqueness of individual service customers. The collaborative filtering recommendation algorithm covers items only if there are enough users that liked them before. In other words, user preferences are dependent on other user ratings, which cannot be tolerated in many service domains. Another disadvantage of collaborative filtering is that recommendation results are presented to users as a black box, without the possibility of any reasoning. Service items in domains like travel, tourism, or health could have very complex structures with multiple item attributes and different kinds of types. From the perspective of the service customer, it might be valuable to understand generated recommendations and reason why and how they have been generated. This would not only increase trust in the system, but also help the service customer in his decision making process in a rational way.

In terms of services, calculation of similarity among recommendation items is easier than finding similarities between users. A family with children might have rated different kinds of travels in the past positive. It is very understandable if the destination of these travels and the facilities of the hotel show different values, since travelers do not tend to visit same destinations repeatedly. In this context, it could be necessary to reason recommendation results to the service customer and let him give feedback on the result set. In multiple iterations, recommendation results could be refined and the user would get more accurate result sets. This approach is only available if the user can reason how and why a certain recommendation item has been selected for the result set. This approach is only available if recommendation items have a structured attribute domain that describes their key features.

Considering the complex structure of service domains that require reasoning when the recommendation results are presented, in addition to limitations of collaborative filtering, a content-based recommendation technique would be more appropriate for service domains. In order to handle changing user preferences in the short term, a complementary utility or constraint-based module can improve the performance of the system. The complementary recommendation approach overcomes the cold start problem in addition to handing the short-term demands of service customers. A content-based approach makes it possible to inspect individual item attributes and reason recommendation results. As an alternative to long-term user preferences that are based on past ratings, service customers could adjust utility function of the model, which would result in different recommendation results that cover their temporary needs.

Possible extensions of the proposed model to overcome other limitations of the content-based recommendation approach are discussed in section 6 in scope of the offline experiment.

3.2 Multi-Criteria Ratings

Section 3.1 stresses that the content-based recommendation approach would be the most appropriate technique for a generic service recommendation model in service domains. This section investigates the requirements of this model from the perspective of user ratings.

A recommendation engine finds various user interests and behavior patterns based on past user interactions, which are then used as input for the prediction algorithm. As mentioned, different user interactions with the system could be used as implicit ratings depending on the item domain, application area, and implementation environment. The proposed model assumes that system users evaluate given services with numeric ratings. The required user preference model is calculated based on given user ratings. Numeric ratings are generally preferred in advanced enterprise systems since similarity measures including cosine or Pearson correlation denote similarities of items better in numeric scales. Additionally, ratings with higher ranges could present the quality evaluation in a more detailed manner than binary ratings.

As mentioned in 2.1.5, by means of service quality models, the evaluation of services could be considered a process that begins with the first service encounter and continues throughout the service delivery. Depending on the service domain, customers might have more frequent service encounters where a service customer experiences different service dimensions over various touch points. Assigning a single numeric rating to whole service performance would be the easiest approach from the prediction calculation perspective. Most of the recommendation systems operate with single dimensional ratings due to its simplicity where users have one-dimensional rating vectors. This approach fits well with systems where item domain is simple. Advanced recommendation engines need to consider more than one rating dimension if recommendation items have complex attribute domains or user utility functions. In terms of services, ratings need to represent evaluation of the service customer in different service dimensions. For instance, online restaurant recommendation site Zagat.com represents reviews with four dimensions (food, décor, service, and cost), while Amazon.com lets its users rate bought items with one single rating. In terms of recommendation engines, the computation overload of single-criteria ratings are much less than multi-criteria ratings, and such systems require less user input for the evaluation of a single item. However, multi-criteria ratings are more informative for the users so that they can estimate strengths and weaknesses of individual items. Another example of multi-criteria ratings can be found in tripadvisor.com, which is one of the biggest travel recommendation engines. Their users evaluate hotels in six different dimensions as demonstrated in the Figure 3-1: Multi-criteria hotel rating in tripadvisor.com

Figure 3-1: Multi-criteria hotel rating in tripadvisor.com

Service recommendation systems can require multiple ratings for different service dimensions, but to simplify the similarity calculation of service items, they could calculate an overall value of the individual ratings. As a simple approach for the hotel recommendation, the arithmetic mean of all rating dimensions could return the overall rating of hotels. On the other hand, some users could value the cleanliness of hotel rooms more than location of the hotel. These distinct rating dimensions originate from service quality models that divide overall service customer satisfaction into separate dimensions. Single-criteria ratings could denote the overall service quality, but they do not cover these important service dimensions separately. Evaluation of distinct dimensions separately is not only essential for service customers, but also for the service provider management team to improve service quality. As discussed in section 2.1.5, quality evaluation of services is a complex assessment process and it is necessary to distribute service item ratings to multiple criteria. Additionally, individual rating dimensions have different significance values depending on the service domain and active user preferences. In order to cover these differences, the generic model extends multi-criteria ratings with dimension weights.

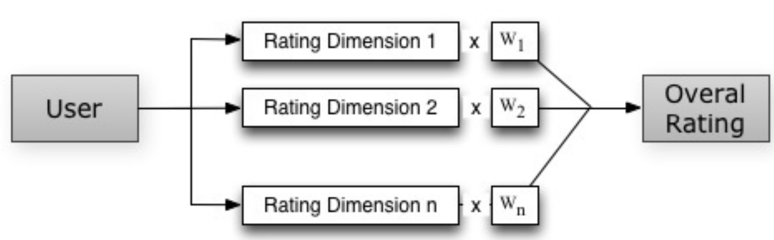

Figure 3-2: Weighted multi-criteria rating

As seen in Figure 3-2: Weighted multi-criteria rating, user needs to evaluate an item over n distinct dimension, and the overall user rating can then be calculated by multiplying the individual rating values with their corresponding weights. Another possible application area of the mentioned dimension weights is the utility function of the recommendation model. These values can be considered as a user's utility parameters to denote his preferences. For instance, if the user travels to a destination without a vehicle, hotel location becomes an important rating dimension that gets a higher weight factor.

As seen in the example of hotel booking, number and type of service dimensions is domain specific. The generic service model assumes that these service dimensions can be aggregated to generic dimensions by considering common features of services. Service quality models can be used to derive these generic evaluation perspectives, which could be taken as rating dimensions in the scope of a generic recommendation system.

According to *Three-Component Model (2.1.5.3),* service quality depends on three distinct dimensions, namely *service product*, *service delivery*, and *physical environment*. Three-component model is an essential quality model to determine abstract service dimensions. A service customer can, for instance, evaluate a certain service by answering the following questions associated with the corresponding service component:

Service Product	How good is the core service output? Do benefits fulfill expectations? How is price/quality relation?
Service Delivery	How is service provided? How is the attitude of personnel? Does service provider make me feel valuable and an important customer?
Physical Environment	How are internal and external service settings? Do tangibles elevate service quality? What are my first impressions about the provider?

Table 3-1: Service rating based on three-component model

Service product is the core benefit that is offered to a service customer. In other words, this is the output that a service customer is willing to pay for. In the case of accommodation services, customers pay for housing rights for a limited duration. During this period, the service customer also has the right to use the facilities and complimentary services provided by the service provider. In the case of a spa therapy or medical operation, the customer expects a positive physical and mental change. Another example from a different domain is digital cable TV service, which offers its customers broadcasts of various television channels and internal services including gaming and shopping channels. The service product of the cable TV service is the ability to watch digital channels on the customers' own television devices. The following table demonstrates possible examples of a service product in various service domains.

Service business	***Service product***
Car repair	Physical improvement/replacement of a corrupted part.
House insurance	Economic compensation in case of physical damage.
Medical services	Better health conditions.
Hair cut	Described hair cut.

Table 3-2: Service product in different service domains

It should be noted that the service product is not the value offered to the service customer. The service value is created with all service touch points and interactions throughout all service encounters. In other words, service value is perceived as a process while service output denotes the main service benefit. Three-component model points to the fact that perceived service output should be evaluated apart from the physical and environmental factors and service delivery. Due to the inseparable nature of services, it is impossible to distinguish service production and delivery in real world scenarios. As a consequence, the service product can be labeled as an expectation fulfillment level regarding the service specific promises of the service provider.

Service delivery in *three-component model* represents how well the service product is presented or delivered to the service customer. It includes all service encounters and the attitude of the service personnel during the customer-employee interactions. Level and type of service delivery is once again service dependent, but it includes all sub-processes beginning from service request through to end of service perception in the eyes of the service customer.

In addition to service encounters, service delivery also encapsulates the time dimension. The service customer does not only complain if the provided service product deviates from promises, but also if it cannot be delivered in the promised time. In this context, service duration should also be considered a service delivery component. In other words, in the case of a two-criteria ranking model, if the promised service is delivered with a delay, the user would evaluate the service delivery with a lower rating, rather than the service product. Unfortunately, this assumption about mentioned service dimensions is not symmetrical. In real world scenarios, most of the customers consider intentional or unintentional service product to be a significant decisive factor for positive service rating. The zone of tolerance of service customers is bound to a certain level of service product quality. Due to that, service product and service delivery is one-directional dependent. This means that if the promised service product cannot be achieved, customers will also rate service delivery in a negative manner.

Apart from service product and delivery, three-component model also refers to tangibles and all internal and external settings of the service in a distinct dimension. These components not only give service customers a first impression about the service, but are also considered an important part of the service evaluation. For instance, a transportation company can provide a friendly and safe service to its passengers without any delays or organizational problems. If the transport company does not maintain its vehicles, however, or keep them internally and externally clean and provide a comfortable environment to its passengers, this would indicate a lack of service settings. Just like service delivery, service environment can also cause successful service delivery to fail. For instance, passengers who travel in unmaintained vehicles an consider themselves in partly uncomfortable conditions, and this can result in customer dissatisfaction. For service domains where physical components are tightly coupled with service delivery, the physical environment dimension can be considered a significant success factor. On the other hand, tangibles that do not affect service product or delivery can be seen as a showcase of the service provider that makes a first impression about the service. In service domains where the service provider and service consumer reside in different physical environments, service customers can experience other service touch points. For instance, the design and content of the homepage of an insurance company can be considered an important starting point for the first impression of customers to evaluate service quality.

As seen in the examples above, the three dimensions of the three-component model do not refer to a completely independent service dimension. They are partly related to each other and each of them can elevate or damage impressions about another service dimension. Even so, it is important to model these dimensions as different ratings so that users can reason what could have gone bad or good.

The three-component model is a simplified model for service quality evaluation and its aggregation, especially about service environment needs and further extensions for service dimension ratings. For further considerations about the generic service recommendation model, SERQUAL or its extension multi-level hierarchical model described in 2.1.5.5 could be referred to.

In service domains where the participation of the service employee plays an important role in service quality, it could be notable to distinguish the performance of a service employee from the generic service delivery dimension. The same assumption is also valid for the involvement of the service customer. The service employee is a very significant factor and aggregates multiple dimensions relating to attitude and skills as described in SERVQUAL. The proposed model evaluates service personnel as a distinct dimension and separates it from service delivery. The attitude of service personnel during the service process is a notable factor for service quality, and customers expect service personnel to be polite, friendly, and helpful. The insufficient expertise or experience of service employees can result in dissatisfied service customers. The level of required skills and knowledge depends heavily on the service domain and can affect service quality from the perspective of service marketing, as described in 2.1.4.3. For instance, a restaurant guest expects a waitress to know about the meals listed in the menu, and expects to get proper answers to his specific questions. In the generic service recommendation model, service personnel needs to be a distinct rating dimension for services with a high level of service employee participation. The service delivery dimension refers to evaluation of service delivery time, service presentation, and any supplementary set of service activities that is not obligatory for the promised service output. Most of the time these supplementary activities involve certain tangible components like a welcome drink or special gifts.

The reliability dimension of SERVQUAL denotes whether promised service output can be achieved accurately and in a predefined time. A service provider is only reliable in the eyes of the service customers if the promised service product creates a value and the promised service could have been provided as described. Considering that service customers know what they are paying for, reliability of the service customer is evaluated through the rating dimension *service output* that was mentioned above. If promised service and actual output do not match, this would result in a low service output rating.

As mentioned above, the physical environment dimension of the three-components model needs further extension for more complex service domains. The proposed generic model derives from the physical environment dimension of the three-component model, two further sub-dimensions namely *accessibility* and *tangibles*. Tangibles are the physical components of service that the user experiences over service touch points. In SERVQUAL, the *accessibility* dimension denotes the approachability of the service provider towards the customer. The generic model refers to accessibility in a more generic way, with multiple aspects including physical reachability, availability of service, and policy of service provider towards special customer requirements or demands. Accessibility of a service is a controllable factor that can be changed by internal and external service settings and management decisions. The Table 3-3: Possible accessibility factorsdemonstrates possible examples of accessibility factors in different service domains.

Service domain	***Accessibility example***
Any	Parking place
Any	Working hours
Telecommunication	24-hour call center / coverage
Finance	Tele/net banking
Entertainment	Physical accessibility for handicapped customers

Table 3-3: Possible accessibility factors

The Figure 3-3 : Generic service rating dimensionssummarizes generic rating dimensions derived from the dimensions of the three-component model and associated with the sub-dimensions of the multi-level service quality model. Depending on the service domain, further rating dimensions can be derived from given the sub-dimensions. For instance, if it is important to underline waiting time for a certain service, *waiting time* can be taken as an additional rating dimension. It should be noted that an increased number of rating dimensions also has certain disadvantages for the recommendation model. Multi-criteria rating systems provide more detail about the item evaluation, but from the perspective of the recommendation engine it would mean more computation overload for the rating prediction. In addition, it requires more user input, which could be inconvenient from the perspective of the system user.

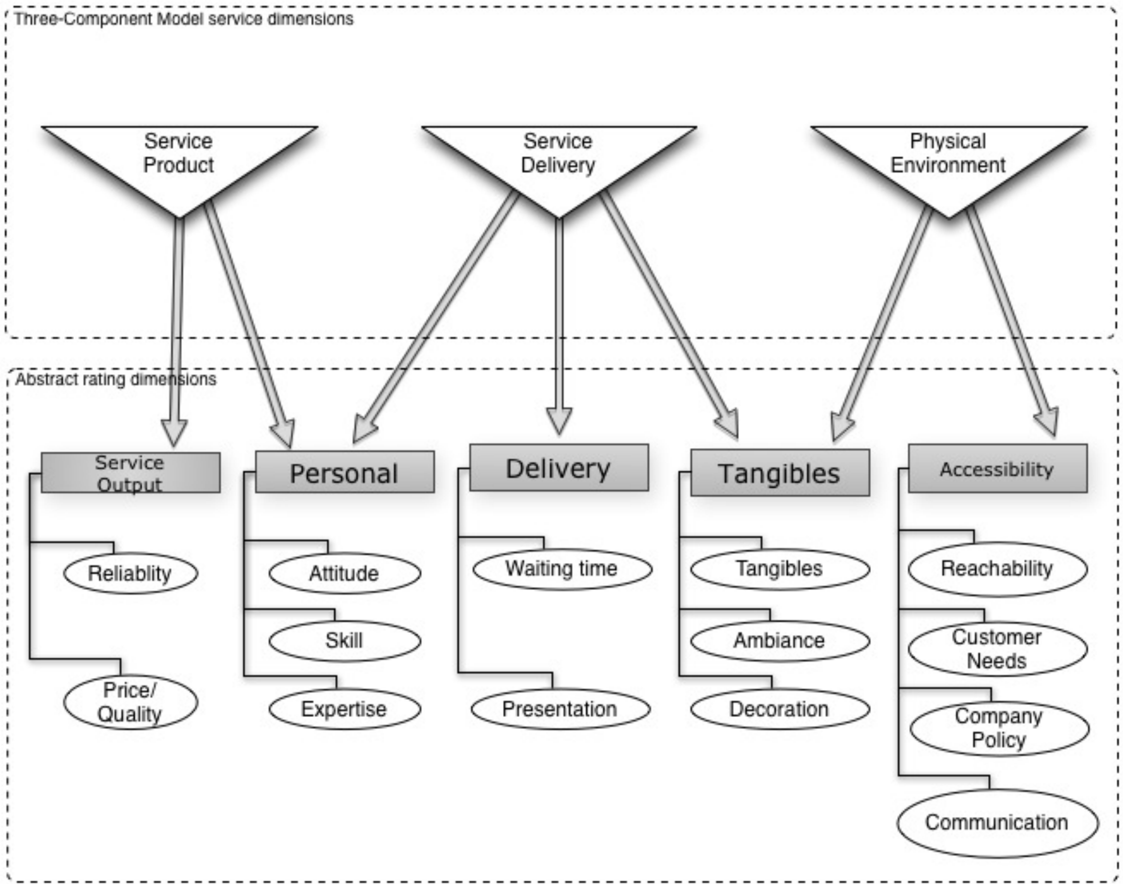

Figure 3-3 : Generic service rating dimensions

Parasuraman et al. (1988) proposed five service dimensions that service customers consider during the quality evaluation. As described above, some of these dimensions also represent the major rating dimensions of the generic service recommendation model. All of these service dimensions can have different levels of significance depending on the service domain, as mentioned with the example of hotel ratings. According to a SERVQUAL survey of Arlen (2008), the following chart demonstrates the importance distribution of these five dimensions. The survey has been conducted among different service sector costumers and denotes overall relevance of the individual dimensions.

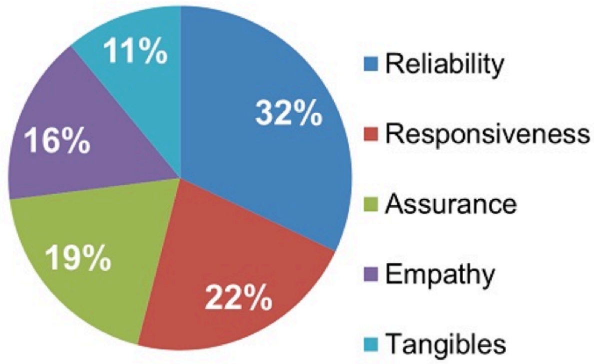

Figure 3-4 : Relevance weighting of service dimensions (Arlen, 2008)

As seen on the chart (Figure 2-1 : Product-oriented marketing triangle), service customers find the reliability of the service provider to be the most important factor, and expect to perceive the promised service. Apart from that, they expect employees to respond quickly and intently. Customers do not want to wait long for a specific service or service provider response. In addition, customers expect service employees to be experts in the service that they deliver. They want to interact with employees that have sufficient knowledge and the ability to perform desired tasks. Empathy does not seem to be as important as reliability or responsiveness, but without communicating with customers and understanding their needs it is not possible to establish a reliable service where contact persons respond quickly to customers needs.

The survey states that tangibles are the least important dimension in the quality assessment, but they might be important for customers to pre-evaluate the service quality and to give a good impression during service encounters.

As mentioned, this survey contains feedback of service customers from various service domains. Thus, individual significance values denote an overall opinion that could vary on specific service domains. For instance, tangibles are considered to be a complementary supporting dimension so that they have the least value among all dimensions. In contrast, services with a high level of physical components require more attention to tangibles. For instance, a customer of a bank could care less about tangibles, but a visitor to a theme park would care primarily about service touch points that maximize his joy.

No service customers like to wait for service delivery, and they expect service provider response as fast as possible. However, in the case of health services, patients have more flexibility to responsiveness, while assurance of the health personnel’s expertise would be a more important factor. Many physicians hang their certificates and diplomas in their offices to reassure their patients about their expertise. Apart from the expertise of the doctors, the patient would also expect a hygienic environment that would once again be listed under the tangibles dimension.

The following two charts demonstrate the results of other SERVQUAL surveys for two different service domains. The first chart shows the opinions of catering service customers and the second chart represents the SERVQUAL score of *National Criminal Justice Reference Service* in drug trafficking areas (The Oregon HIDTA).

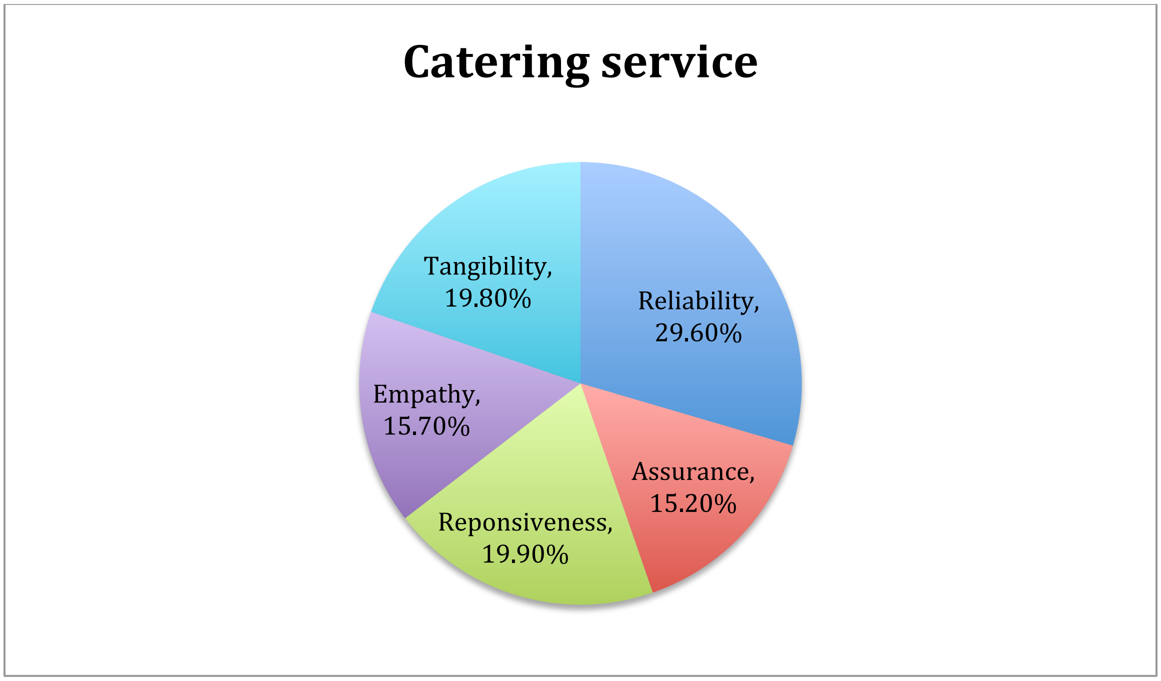

Figure 3-5 : SERVQUAL scores for catering service (Curry & Brysland, 2001)

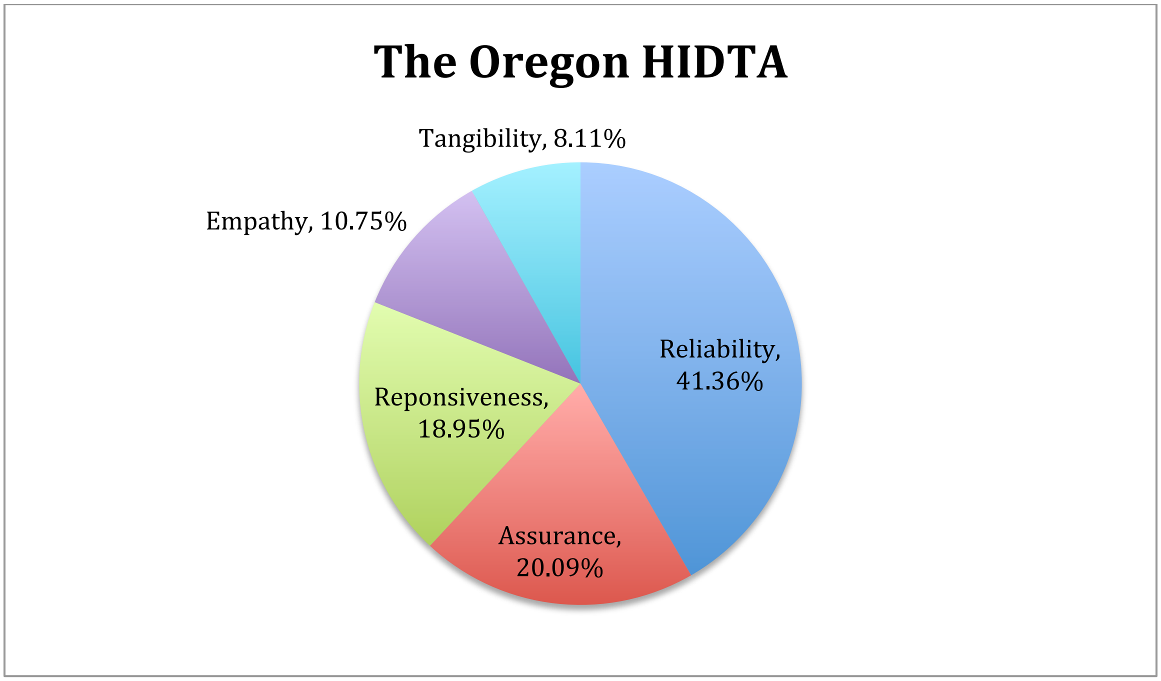

Figure 3-6 : Public service SERVQUAL scores (Gibson, 2009)

As seen in the mentioned examples, the relevance of the service dimensions is heavily dependent on the service domain. Even inside the same service domain, different service customers can have different preferences, and as a consequence different priorities about the service quality evaluation. From this perspective, in terms of the generic recommendation model it would not be accurate to assign the same weight to all service dimensions. A dynamic rating dimension weight not only brings scalability to the recommendation engine; it also enables a level of personalization for ratings and a level of unique interpretability to multi-dimensional item ratings. In other words, same numeric ratings can result in different overall rating results if two service customers have different rating dimension weights. Self-assigned dimension weights, on the other hand, represent the short-term utility function of the recommendation engine in addition to preference weights.

The proposed generic rating dimensions could be applied to any kind of service business. Depending on the service domain, number, and type of service encounters, this generic rating framework could be extended with further dimensions as proposed by *Integrated Quality Model* in section 2.1.5.6. It should be noted that the increased number of dimensions is not very desirable in terms of recommender systems due to the curse of dimensionality problem, as mentioned in 2.2.5.4.

3.3 Users

This section investigates the requirements of a generic recommendation engine in terms of user entities. Required user-specific information depends on the type of recommendation technique. Collaborative filtering systems do not require any user-specific information to produce a rating prediction. In this approach, users are represented only with their past ratings, and depending on the algorithm similarity between users or items, are only calculated by considering the given user ratings. In content-based recommendation systems, item attributes are studied and a recommendation engine creates a user profile based on the found patterns. This profile represents the tendency of the active user. Just like collaborative filtering, content-based systems do not require explicitly demographic user specific information in order to operate.

As mentioned earlier, in addition to the content-based approach of the proposed model, a supplementary utility-based recommendation engine helps to cover changing user preferences in the short term. Utility-based recommender systems for services like accommodation booking sites could list recommendation results based on different features including price, location, presence of breakfast, and so on. For certain guests, hotel location could be more important than price, while some could prefer to have breakfast. UBRS or KBRS do not necessary require user specific demographic information, but demographic information could be used to improve recommendation results. For instance, tripAdvisor.com distinguishes hotel reviews into five different groups including family, couple, solo, business, and friend reviews. If a user logs in with his Facebook account, he could list feedback of his Facebook friends. In general terms, demographic information provided by social networking sites could be used to refine recommendation results. Over this approach, a recommendation engine has access to the latest information about the users without any need to store or maintain it.

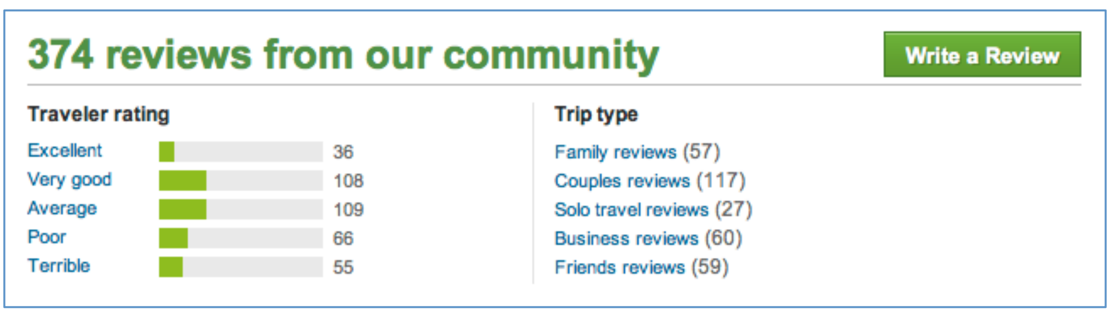

Figure 3-7 : *tripAdvisor* groups user reviews based on demographic information

The proposed model does not require any user specific information to operate, and models user interests based on past ratings and attribute values of rated items. From the perspective of data modeling, users could be presented only with unique ids.

3.4 Items

Content-based, utility-based, and knowledge-based recommendation systems require information about item attributes in order to operate. In content-based recommendation systems, items are represented as n dimensional attribute vectors. Individual dimensions represent significant item attributes that help the recommendation engine to distinguish similar items. As a golden rule of data modeling, item vectors should not contain redundant data in order to avoid the curse of dimensionality problem. Information retrieval systems are simple forms of content-based recommender systems. These items are represented by certain keywords, and the corresponding dimension gets the value 1 if a given keyword is present in the document and 0 otherwise. In other applications areas of content-based recommendation systems, item attributes can have any kind of value range, including ordinal, categorical, nominal, binary, and textual. Processing and interpreting textual attributes for recommendation systems is not in scope of this master's thesis. The proposed model assumes that individual service items are represented with structured types depending on the service domain.

Item attributes should be as informative as possible, and should be distinguishing. Attribute entropy values can be considered for primary selection. Entropy denotes how much information an item attribute value contains, or in other words much many bits are required to code a given attribute value based on the relative frequencies of individual values and the cardinality of the attribute. For instance, a binary attribute has less entropy value compared to another attribute with a range of 10. The major problem with services compared to goods in terms of attribute modeling is the heterogeneity of service entities.

Modeling a recommendation item means elimination of unnecessary features and properties and representing it in a simpler form. Recommendation items are therefore bound to given attribute values and can only be compared over these values. A content-based recommendation engine considers two different recommendation items as "same" if they own the same attribute values. As seen in the table below, *Good A* and *Good B* have exactly the same attribute values, which implicates the same economic value for the system user. Both items would get the same distance to the user preference considering only these three attributes. Same distance to user preference indicates same rating prediction and possibly same ranking in the recommendation list. From the perspective of the user, both items are identical considering their attribute values. A deterministic recommendation approach would assign the same rating to both of these items.

User Preference	***Value***
Attribute A	B
Attribute B	3
Attribute C	NO

Good A	***Value***
Attribute A	B
Attribute B	3
Attribute C	NO

Good B	***Value***
Attribute A	B
Attribute B	3
Attribute C	NO

Table 3-4: Identical item attribute

Due to the heterogeneous characteristic of services, mentioned assumption has different implications for service customers. A user might have rated a *service S1* with the maximum rating value while he has given for another *service S2* the lowest rating just because he was dissatisfied because of an employee attitude. This customer evaluation behavior is quite common in many service domains and causes an important anomaly in recommendation engines. Just like service customers, different service providers are also unique, and they do not provide the same service. From the perspective of the recommendation model, two service providers cannot be distinguished if they share exactly the same attribute values. Theoretically, both service items are supposed to be the same, but the user could evaluate them totally differently. Such ratings break the patterns in past interaction history, or in a most simplified form they decrease the accuracy of ratings for items of the same category. In the long term, this results in higher RMES values, which indicates how much predicted and actual ratings deviate from each other. To cope with such cases, a recommendation engine could ignore the outliner ratings or require an enumerated explanation for extreme values of user ratings to exclude them from user preference calculations.

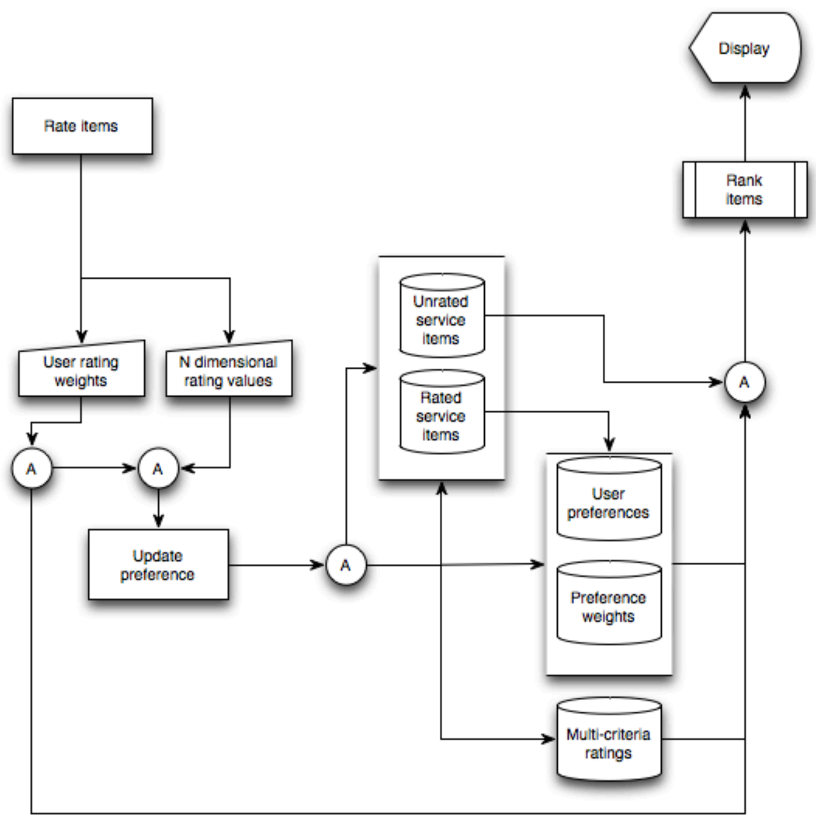

Figure 3-8 : Generic recommendation model overview

Figure 3-8 : Generic recommendation model overviewdemonstrates the information flow inside the generic service recommendation model. *Rating an item* activity requires from users n dimensional rating weights and rating value vectors. Each given rating updates the user preference model in two dimensions. A new item rating primarily updates the set of rated/unrated items. If given user rating weights deviate from the previous settings, the model needs to iterate also present ratings to calculate new overall rating values. After the new overall ratings are calculated, the model needs to update user preference. When the user preference is ready, the model presents to the active user his preferences as attribute values and their weights. Once again, the user has the ability to change preference weights based on his demands. When the active user requests recommendation, unrated items are patched from the database and each item gets a rating prediction based on the preference settings. The recommendation items are sorted on this rating value and presented to the user as a ranked list.

4 Service Recommendation for Restaurants

Section 3 proposes a generic recommendation model for all service domains. In the empirical part of the thesis we introduced a generic model that will be evaluated in the example of restaurant recommendation. This section discusses application of the proposed model on a concrete service domain and explains the recommendation algorithm, similarity, preference, and preference weight calculation in detail.

Designing and building a recommender system is a interdisciplinary work of many computer science fields including machine learning, data mining, information retrieval, artificial intelligence, etc. (Ricci et al., 2011).

The following restaurant recommendation model is a simplified version of a content-based recommendation model that does not cover all aspects and problems of enterprise recommendation engines. Additionally, the proposed model is extended with a utility-based recommendation engine to react to one-time demand changes. The main purpose of the model is the application of the proposed generic model on a concrete service domain with specific service dimensions that can be evaluated in the scope of an offline experiment.

Restaurants are exceptional examples of services, with many tangible components and the fact that they lie in the middle of the good-service continuum, being apart from pure services. This is caused by the involvement of many different types of service touch points during service delivery. Due to their complex structure, they provide a very rich number of service dimensions that can be considered as rating dimensions. Considering possible types of restaurant customers, each user could have his unique evaluation preferences. Primarily, customers evaluate the consumed meal as a core service, but other service dimensions are also considered for the quality evaluation. The quality perception of restaurant customers can vary depending on the customer stereotype. For instance, some restaurant customers love certain places just because of their ambiance or the attitude of servants.

A restaurant offers its customers ready-to-eat food in a physical place, cuisine related supporting equipment, and ambiance as service. The ambiance and the physical settings of the restaurant depend on the category and class of the restaurant. While customers consume their food, they also require durable goods like tables, chairs, knives, handkerchiefs, etc. Restaurant customers experience various service encounters during a restaurant visit. In luxury restaurants, a valet parking attendant can welcome the guests, and afterwards another person assists him at the entrance of the restaurant for checking the reservation and guiding the guests to their table.

Although restaurants are exceptional services, their quality evaluation is processed like any other services in multiple parallel or sequential dimensions. Primarily, a good restaurant must provide its customers the same level of taste, which is not only dependent on the chef, but also other factors including ingredients from its wholesaler and the equipment and capacity of the kitchen. Secondly, it is not possible to satisfy the guest by providing only delicious meals, since guests would like to have their meal in a friendly and hygienic environment. An unclean plate or unkind waitress will affect the value transition in a negative way, which might result in dissatisfaction.

4.1 Service Dimensions in Restaurants

A restaurant guest experiences many service encounters and has the opportunity to judge various service dimensions and touch points with his various senses and tastes. In this section, possible service encounters during a restaurant visit are investigated with the help of a service blueprint. The following service blueprint considers only a sub-set of possible customer and employee actions, and could vary from one restaurant to another. All the considerations about the quality evaluations of the restaurant customers are taken from various restaurant evaluation forms (Anonymous, 2012), (Roundtable, 2012).

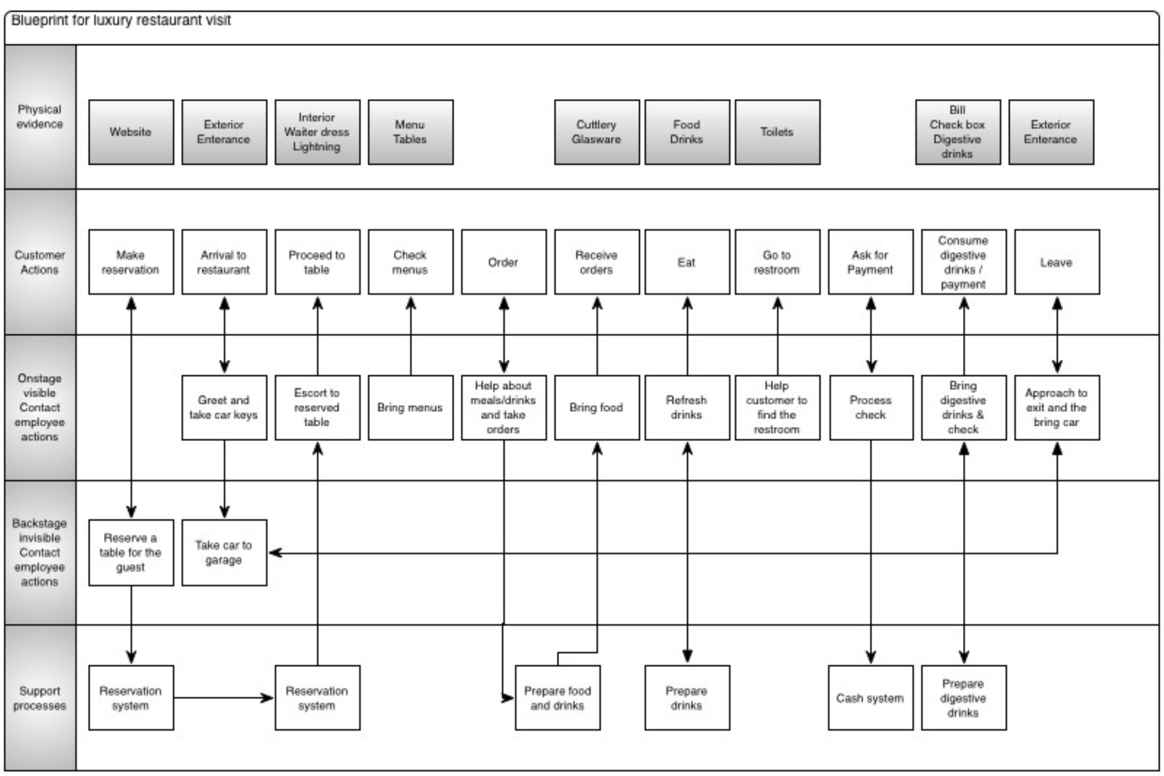

Figure 4-1: Service blueprint of luxury restaurant visit

Reservation

The website of the restaurant can be considered as the first touch point for the customer. Restaurant users experience their first service encounter as they make their reservations over the web service of the restaurant. The website itself is not necessarily the "*moment of truth*" in the eyes of the restaurant guests, but a well designed professional looking site could easily impress visitors and will increase the image of the restaurant. If restaurant users experience difficulties or any problems during the reservation process, this will leave a negative first impression.

Arrival to Restaurant

Physical accessibility can be named as a relevant factor or service feature to determine how a customer reaches the physical location of the restaurant. At first glance, restaurant customers can observe the physical environment or entrance of the restaurant. They judge if the entrance and the signboard of the restaurant make an inviting impression with its visibility, lighting, and appearance.

Restaurant visitors with vehicles consider availability of parking slots, and for more luxury restaurants valet parking can be seen as an important customer requirement. As a support process, cars of restaurant guests can be washed in the garage. Handicapped restaurant guests demand easy and safe wheelchair access to operation, and they can also mind the condition of the sidewalks in front of the restaurant from the perspective of a person who rides a wheelchair. A service provider is responsible for providing easy and safe access to its operation.

For many services, operating hours can be a significant persuading factor and an important competitive attribute. The proposed model considers all restaurants as having the same operating hours to keep the database modeling simpler.

When the customer enters the restaurant, a waiter that greets him in a friendly manner and guides him to an available table indicates a warm, hospitable attitude of the service personnel. In the case of restaurants that demand reservations, the approach of the personnel when asking for a customer's name or the way in which he denies customers without reservations when there are no available tables is absolutely important. If the service personnel can convey to a visitor that he is a valued customer, this will also elevate the evaluation of other service encounters in a positive way.

As mentioned before, possible customer interactions and observations during the first service encounter play a significant role in the judgment of service quality. These user impressions can be distributed to multiple rating dimensions from section 3.2, including accessibility, tangibles, service delivery, personnel.

To be Seated

This activity has a rather short duration compared to others, but during this service encounter, the customer has an opportunity to observe many factors that are related to the *tangible* dimension of the service provider. In certain restaurants, a waiter can escort visitors to a table. During this time, customers can judge cleanliness of the floor, smell, lighting, decoration, and if present, music level and gender that make up the ambiance.

Sometimes a restaurant guest does not approve the first proposed table, and can have optional requests in regards to size and position of the table. If this demand cannot be fulfilled for some reason, it should be explained to the customer in a friendly, hospitable manner. It should be noted that each restaurant visitor is unique and can have many different kinds of special requests to feel confortable and convenient.

During this short service encounter, customers can evaluate the restaurant tangibles and have opportunity to experience the attitude of service personnel during guidance to a table or towards special customer requests.

Order

After being seated, the customer can observe the ambiance and further tangibles inside the restaurant while he waits for his menu. He judges if the table and chairs are comfortable, as well as the hygiene of the linens, glassware, and silverware on the table. The customer expects to receive a restaurant menu in an acceptable amount of time and in a hospitable manner by the waiter. When the menus are brought, the customer primarily evaluates its physical condition, considering its cleanliness and appearance instinctively.

During wine or meal selection, the customer expects the waiter to answer inquiries in a helpful manner. The waiter is supposed to know about the contents of the menu, and about meal ingredients. He also needs to be knowledgeable about beverage and wine selection. For instance, he should be able to suggest a proper wine depending on the selected meal, or offer seasonal or daily restaurant specialties. Meal order process is an important service opportunity for the waiter to show his expertise. While taking the orders, the waiter is expected to be polite towards customers, with proper eye contact and hospitable speech.

Serving

Customers expect ordered meals to be served in a reasonable amount of time, and also expect the simultaneous arrival of main dishes. Customers assume that delivered food and beverages are prepared and served as requested or described in the menu. Meals and drinks are supposed to be brought in at proper temperatures.

A customer also evaluates the waiter's professional serving techniques, attitude, and appearance, including cleanliness of his uniform. After the desired meals have been served, a waiter should keep paying attention to table conditions and the special needs of customers. Customers expect the waiter to ask for reorder of the drinks or refills of wine glasses without any customer prompt. Any special customer requirements, including condiments, dips, or additional glassware and silverware should be satisfied promptly. Furthermore, the waiter should ensure customer satisfaction on the delivered dish and make sure that tangibles on the table are still proper.

It is important to exclude evaluation of this service encounter from the taste of consumed meals and render it under rating dimension *service*. In most of the restaurants, preparation and serving of meals is performed by different actors. A badly prepared meal cannot achieve complete customer satisfaction with only the good serving of the waiter, and a well prepared meal can get a bad rating due to insufficient service or a long waiting time.

Restrooms

Restrooms are also important places in restaurants, and customers demand them to be hygienic and ventilated properly. Restaurant management should make sure that restroom components are cleaned regularly and supplies like toilet paper, soap, or light bulbs are always present. In case of possible mechanical or physical defects, they should be repaired as fast as possible. A negative impression or experience with restrooms could easily change the attitude of a customer, which would have a negative influence on the overall quality evaluation.

Ambiance can be considered as a property that can attract customer attention or create antipathy. Hygiene is in contrast more than a restaurant property and rather an obligation, which is also inspected by government authorities. A restaurant could be operated vey cleanly, but the interior decoration and supplementary equipment could be old. On the other hand, a new trendy restaurant might be operating under unhygienic conditions due to lack of personnel or other external factors.

Payment

When the check is requested by the customer, it should be processed and presented in a frinendly manner. Optionally, a digestive beverage, chocolate, or candy can be served as a kindness, which will elevate customer satisfaction and overcome any possible negative experiences occurred during prior service encounters. As when entering the restaurant and approaching the table, guests also like to feel like a a valuable customer while they proceed to exit.

Food Quality As Service Product

The mentioned service encounters are just possible examples that can be experienced in most of the resturants. The perception of the service quality and relevance of the mentioned dimensions are dependent on the customer stereotype. The service product remains in some service domains ambiguous, as in the case of restaurants, since the provided service product includes a tangible component. Theoretically, the service output is not the served meal itself, but rather the consumption of the served food in a physical environment with supplementary tangibles including equipment and accessories. However, the meal is evaluated as an important factor of the service output, and in terms of the restaurant recommendation, food quality is considered as a service product. The service product indicates food quality as a service dimension, and models the freshness, taste, temprereture, and portion size.

4.2 Application Data Model

The proposed restaurant recommendation model is implemented in terms of a standalone desktop application with pre-given input data. User preferences are not stored persistently for later use and need to be computed on demand. Apart from that, recommendations are also generated with a memory-based algorithm of the model, and they are not stored in the database of the application for later queries.

User preferences and corresponding preference weights need to be calculated on wire by inspecting present user ratings when recommendations are requested. The data model of the application contains four main tables, namely users, user ratings, restaurants, and rating weights to indicate rating utility of the active user.

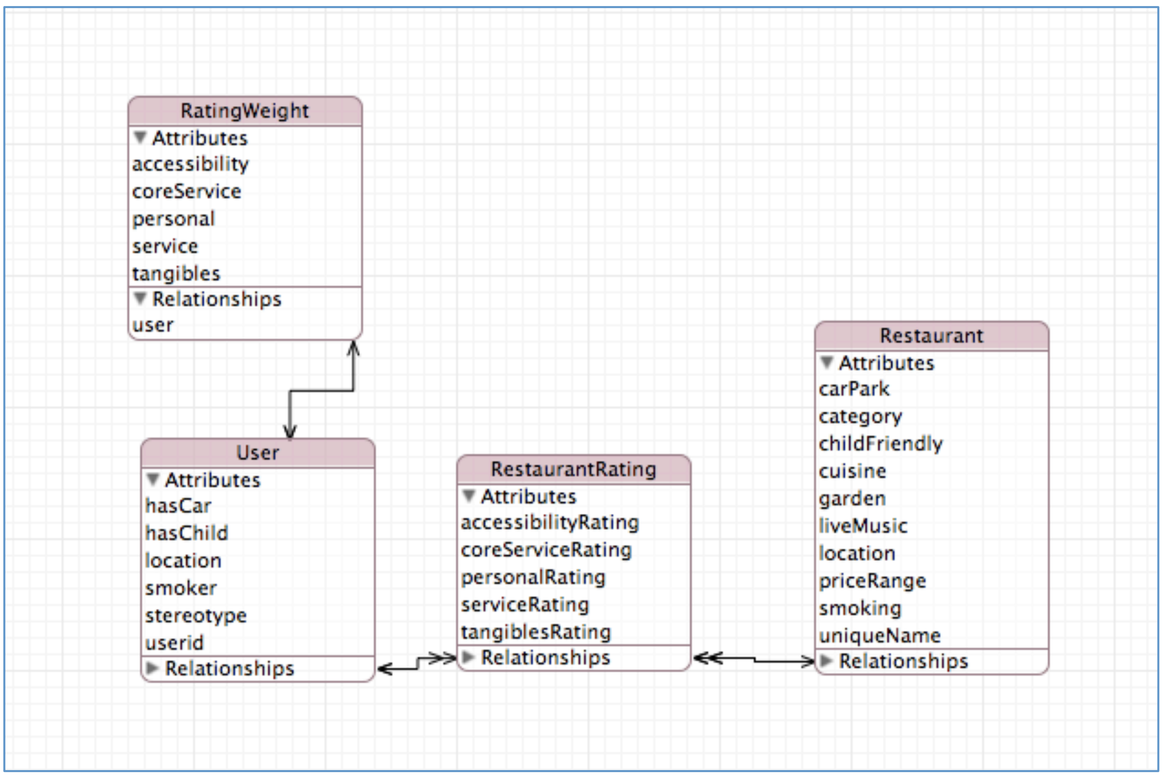

Figure 4-2: Data model of the recommendation application in *Core Data*

As mentioned earlier, the recommendation engine does not require any information from users considering the operation technique of content-based recommendation. The user table is only required for generation of the test users with different preferences as explained in section 5. This table is not relevant for recommendation generation, and the model operates mainly on restaurant, restaurant rating, and rating weight tables.

4.3 Rating Dimension Weights

As mentioned earlier, restaurant items are evaluated over five distinct dimensions so that each restaurant rating corresponds to a five-dimensional vector. As seen in the data model (Figure 4-2: Data model of the recommendation , each system user owns a separate table where corresponding rating dimension weights are stored. Rating weights are changeable in the short term, and are required to convert five-dimensional restaurant ratings into a single overall rating.

$$R_{i\ overal} = \begin{pmatrix} R_{Service\ product} \\ R_{Delivery} \\ R_{Tangibles} \\ R_{Accessibility} \\ R_{Personal} \end{pmatrix} x \left(W_{Service\ product} \quad W_{Delvery} \quad W_{Tangi.} \quad W_{Acc.} \quad W_{Pers.}\right)$$

Formula 4-1: Overall rating calculation

Overall restaurant ratings of the active user can be compared with the utility matrix of recommendation systems with single-criteria rating systems. For computation of user preferences, overall ratings are taken into account rather than considering the ratings of different rating dimensions to keep the calculation simple. By assigning proper weight factors, the user has the ability to distribute the weights to five different service-rating dimensions. Rating dimension weights could be compared with the utility function of the user to denote his evaluation attitude. Corresponding rating dimension tables store the normalized rating weights so that their sum makes 1. A user could change his preference values by assigning different weight factors to different dimensions that would affect the overall rating of restaurants. The following example demonstrates the consequence of different weight values on the overall restaurant rating value.

Restaurant X	
Service product:	8
Delivery:	6
Tangible:	2
Accessibility:	3
Personal:	8

Table 4-1 : An example restaurant rating (Overall)

Weight	User A	User A
Service product	0.25	0.15
Delivery	0.15	0.15
Tangible	0.2	0.35
Accessibility	0.1	0.25
Personal	0.3	0.1
Overall	6.0	4.35

Table 4-2 : Weighted rating for restaurant X

It should be noted that users rate restaurant items in five dimensions with ratings in a range of 0 to 10. As seen in the first table (Table 4-1 : An example restaurant rating (Overall)), user ratings of restaurant *X* for dimensions *accessibility* and *tangibles* are significantly lower than service *product* and *personnel*. As a consequence, the user gives an overall rating of 6.0 for this restaurant since he has a weight factor of 0.25 for the dimension *service product* and 0.3 for personnel. The same user changes his utility so that personnel, service product, and delivery become less important, and he increases weights for tangibles and accessibility. As a consequence, for the same restaurant his overall rating changes from 6.0 to 4.35. In the application model, this adjustment would change multiple overall ratings, and as a result user preferences would get different values. As a consequence of changed preference values, the active user would get different recommendation results.

4.4 User Preference Calculation

User preferences or user models in the context of recommendation systems denote the most preferred item attribute values. The goal of building user preference is extracting most favorable attribute values for item features as n dimensional vectors. The number of dimensions in the preferences vector conforms to number of restaurant attributes in terms of the proposed model. The user preference shows patterns and tendencies of active users' ratings, represented by the individual recommendation item attributes with corresponding numeric values.

Attributes

Attribute	Type
B carPark	Boolean
S category	String
B childFriendly	Boolean
S cuisine	String
B garden	Boolean
B liveMusic	Boolean
N location	Integer 16
N priceRange	Integer 16
N smoking	Integer 16
S uniqueName	String

Figure 4-3 : Restaurant attribute types

As seen in the screenshot above, restaurant entities have various qualitative and quantitative attributes with different value ranges. Restaurant category and cuisine are examples of unordered nominal attributes with a limited set of possible values. *carPark*, *childFriendly*, *liveMusic,* and *garden* are examples of Boolean attributes. Another categorical attribute *smoking* has three possible values, including "smoking allowed," "smoking permitted," and "separate saloons."

In real word scenarios, user preferences are stored and updated after each new user rating for reuse. The proposed model's memory-based recommendation approach computes the required user preference model on demand considering the rating weights and restaurant ratings. This approach makes it possible to update a user preference instantly when the utility values are changed over rating weights or directly over user preferences. This gives the model the flexibility to cope with temporary requirements that deviate from past user ratings.

There are different approaches to calculate user preferences depending on the recommendation technique. Some systems consider only items with a rating above a certain positive rating threshold. The proposed model iterates restaurant attribute values and inspects past ratings with the given attribute value. For calculation of the preference value for the given attribute value, the preference algorithm computes both average of past ratings with the given restaurant attribute value and relative frequency of restaurants with investigated attribute value in the whole restaurant set. This calculation would be too costly in enterprise systems where there are thousands of items and ratings, so only a subset of total ratings could be investigated rather than the whole entities. Several optimization possibilities for preference calculation are discussed in the evaluation part of the thesis.

As mentioned earlier, the output of the recommendation process could be a list of recommended items with descending predicted ratings. The proposed model follows the same approach during preference calculation, where for each restaurant attribute, a list of its possible values is sorted on preference ratings. The preference rating of a certain attribute value denotes the rating prediction of a restaurant by considering only the given attribute value. For this approach, the model does not only need to know about the favorite restaurant attribute values, but it also needs to associate individual attribute values with ratings.

Smoking	
Attribute Value	***Preference Value***
NO	[1-10]
YES	[1-10]
YES/NO	[1-10]

Table 4-3 Preference values for smoking attribute

As seen in the table above, with smoking attributes, the preference value represents the attribute value in the rating scale. The attribute value with the highest preference value is considered to be the most favorable value for the given attribute. As mentioned earlier, for calculation of preference rating values, the model iterates user ratings of the active user with this being a certain attribute value. For instance, in order to find corresponding preference values for smoking restaurants, the model primarily retrieves all user ratings where the rated restaurant has the value YES in its smoking attribute. Since the ratings contain separate values for all rating dimensions, the model computes the overall rating based on user rating dimension weights to make the preference calculation easier. In the simplest form, it is possible to take the arithmetic average of all restaurant ratings that have smoking value YES for the preference value. In order to assign a significance value to individual attribute values, the proposed model also considers a number of given ratings in addition to the arithmetic mean of the ratings. This approach could be considered as a type of significance correction measure of high but less frequent ratings for certain restaurant attribute values. As seen in Equation 4-1: Preference calculation for attribute x – value 1, preference value for value n of attribute X ($P_{User\ a}(Attribute\ X_{Value\ n})$ is calculated by weighted sum of *rating-based* and *count-based* preference values (Karaman, 2010).

$$P_{User\ a}(Attribute\ X_{Value\ n}) = P_{RBP_{Value\ n}} * (\beta) + (1-\beta)P_{CBP_{Value\ 1}}$$

Equation 4-1: Preference calculation for attribute x – value 1

$$P_{RBP}(Attribute\ X_{Value\ n}) = \frac{\sum_{i \in\ value\ n} R(u,i)}{number\ of\ ratings_{Attribute\ X-Value\ n}}$$

Equation 4-2: Preference calculation based on past ratings

$$P_{CBRP}(Attribute\ X_{Value\ n}) = \frac{number\ of\ ratings_{Attribute\ X-Value\ n} - \min number\ of\ ratings_{Attribute\ X}}{\max number\ of\ ratings_{Attribute\ X} - \min number\ of\ ratings_{Attribute\ X}}\ x\ 10$$

Equation 4-3 Count based preference value

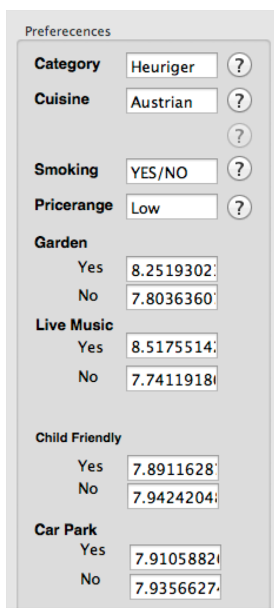

Figure 4-4 : User preference box of restaurant recommendation application

The recommendation application presents the user preference as lists of values with corresponding preference rating values. The first four restaurant attributes have cardinalities higher than 2, so given text fields show only the most preferable attribute values, as seen in the Figure 4-4 : User preference box of restaurant recommendation application. Remaining preference values could be observed as ranked lists by clicking the "?" button beside the text fields of most preferred attribute values. The rest of the restraint attributes are binary, so both preference values are presented above each other.

As seen in the screenshot above, values for the binary attribute *child friendly* are very close, while values for the attribute *live music* differ more. By observing the preference rating values, it is possible to interpret the relevance of individual attributes without calculating the required weights. For instance, the mentioned user has a tendency toward restaurants with live music since preference value for restaurants with live music is 8,52 while restaurants without receive only 7,74. On the other hand, there is not enough evidence about the child friendly attribute since values of this attribute differ only by a 0,05 preference rating value.

Rank	FavoriteCategory	Value
1.	Heuriger	7,946154
2.	Food Court	7,428889
3.	Sea Food Resta...	7,069136
4.	Fast Food Rest	7,02381
5.	Steak house	7,02029
6.	Pizzeria	7
7.	Running Sushi	6,924638
8.	Coffeehouse	6,874667
9.	Cantina	6,86087
10.	Fine Dining	6,739785
11.	Osteria	6,506173
12.	Counter Service	6,372222
13.	Bistro	6,271795
14.	Bakery	6,196079
15.	Brew Pub	5,979711
16.	Ouzeria/Tavern	5,969231
17.	Snack Bar	5,864865
18.	Pub	5,790196
19.	Take-Out	0,0

Figure 4-5 : Preferences value list of restaurant category attribute

The example data set above does not contain any ratings for a restaurant with category *take-out*, and the equation 4-1 returns 0 in this case. Without any rating, the model cannot calculate preference value of the attribute category for take-out restaurants. In terms of the recommendation model, missing preference values are interpreted as negative values so restaurants with category value "*take-out*" are considered to be a less preferred restaurant category.

As mentioned earlier, results of content-based recommendation systems are self-explanatory. In terms of the ranked list demonstrated in the screenshot above, one can say that a restaurant would get a rating prediction of 7,95 by ignoring the preference weights and any other restaurant attributes. Depending on the preference ratings and weights of other restaurant attributes, the overall rating prediction of an unknown restaurant would be below or over this value.

4.5 Preference Weighting

As discussed in section 4.4, individual preference calculation of attribute values considers that all restaurant attributes have the same significance for the active user. For most of the users, certain restaurant attributes are more important than others. Apart from that, as an important requirement of utility-based recommendation engines, the active user should be able to adjust his needs by setting relevance factors for individual attributes. The proposed model gives the active user the opportunity to set his rating dimension weights and rating attribute weight. This feature gives the system an important flexibility in terms of changing user interests in the short term. This section describes the calculation of weighting factors for individual attributes as part of the user preference model based on past ratings.

*tf*idf* weight (Section 2.2.6.2.2) represents in the information retrieval relevance of keywords to describe content of a text document. This approach fits well with domains where the item attribute values are only binary. In the application domain of information retrieval systems, feature vectors of recommendation are keywords of the document collection. If a particular keyword appears in a document, the corresponding dimension gets the value 1 (and 0 otherwise). As mentioned in previous sections, restaurant attributes like category, cuisine, smoking, or price range have higher cardinalities. Thus, their weight factors cannot be expressed with *tf*idf*.

For generation of user preference weights, the model refers to Martinez and Barranco (2010) in that it proposes an approach to cope with multi-valued item features. This approach considers the amount of information (entropy) that a feature contains and correlation or contingency between item features and item ratings to calculate feature weights.

Entropy corresponds to the average amount of information, or in other words required number of bits to encode the information. In the context of feature weighting, features with higher entropy are more informative and interesting for the user, and they should also get higher feature weights.

$$H_j = -\sum_{k_j} \frac{f_{k_j}}{n} \log_2\left(\frac{f_{k_j}}{n}\right)$$

Equation 4-4 : Entropy of item feature j

k_j is a feature value and f_{k_j} is the frequency of this feature value among the whole set of restaurant entities. n corresponds to total number of restaurants in the data set. The sum of entropy values for individual restaurant attribute values makes the entropy of the restaurant attribute.

In addition to entropy, Martinez and Barranco (2010) also consider the linear relationship between user ratings and the feature values of those rated items. Correlation between ratings and features indicate that the given feature is significant for the user and requires higher weight. For quantitative attributes, Pearson correlation is used, and for qualitative features, Cramer coefficient[17] could be used to denote the dependency. The weight of attribute *j* for user *u* is expressed as a product of dependency coefficient *DC* and entropy *H* of attribute *j*.

$$w_j^u = DC_j^u \, H_j$$

Equation 4-5: Feature weight of attribute j for user u

After calculation of most favorable item features with corresponding weights, the user profile could be expressed as a matrix (Table 4-4: User preference matrix) where the first column corresponds to restaurant features and the second column denotes the weights for these feature values. As described earlier, user preference does not only hold values of most preferred restaurant attributes, but rather ranked lists of individual attributes values. In this context, rows of the first column represent the ranked list of attribute values. Rows of the second column are the normalized preference weight values for restaurant features.

$$P\,(u) = \begin{bmatrix} Category & W_{Category} \\ Cusine & W_{Cuisine} \\ Price\ range & W_{Price\ range} \\ Smoking & W_{Smoking} \\ Car\ park & W_{Car\ park} \\ Childfriendly & W_{Childfriendly} \\ Garden & W_{Garden} \\ Live\ music & W_{Live\ music} \end{bmatrix}$$

Table 4-4: User preference matrix

[17] Cramer V is used to denote the association between nominal variables.

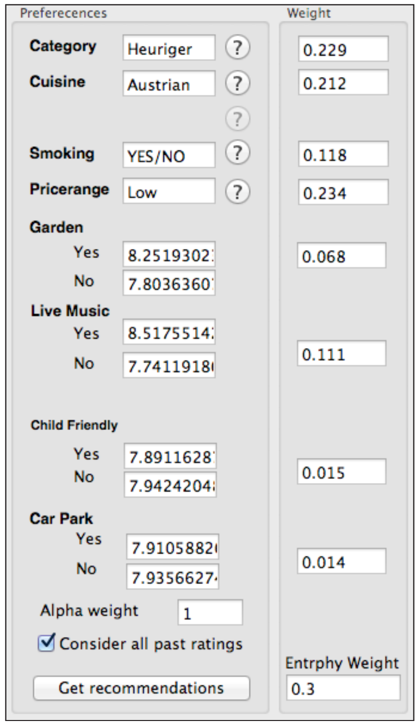

Figure 4-6 : User preference box in recommendation application

As seen in the Figure 4-4 : User preference box of restaurant recommendation application the recommendation application presents the attribute weights beside the preferences box. Individual weight could be changed manually to simulate parameters of utility function of utility-based recommendation engines. Martinez and Barranco (2010) consider attribute entropy and the dependency of attribute values with ratings equally important. Considering the restaurant category with cardinality of 19 and cuisine 15, binary attributes could be dominated due to their lower entropy values. In order to avoid the mentioned drawback of the referred approach, overall weight function is scaled with an entropy relevance factor. Entropy weight could be set at the bottom of the weight box in the ratings tab of the recommendation application.

$$w_j^u = DC_j^u * (1 - AlphaWeight) + H_j\,(AlphaWeight)$$

Formula 4-2 : Scaled preference weight function

4.6 Recommendation Generation

Generation of recommendations depends heavily on how the recommendation results are presented. The proposed model presents recommendation result as a descending list of restaurant items with their attribute values and prediction ratings. In certain domains, already rated items are filtered out from the set of potential recommendation items. The proposed model assumes that a rated restaurant can be re-recommended and does not filter out any restaurants for the recommendation generation.
In order to limit the number of recommendations, recommendation models could consider a positive rating threshold value. Thus, it can show all recommendation items that are considered to have positive rating predictions. Alternatively, Top-n technique can be used to limit the recommendation results with a fixed number. For evaluation reasons, the proposed model lists all recommendation items sorted on the rating prediction.

The recommendation algorithm iterates all potentially interested restaurant items and patches required preference and weight values from the user's preference model. As described earlier, the user preference model is not saved in the data model and calculated on demand. This consideration assumes that before each recommendation request, the user an change his rating dimension weight and utility weights. In order to optimize recommendation generation, user preference is calculated once for the first request and cached in memory as a dictionary (*NSDictionary)* object (NSDictionary Class Reference, 2013).

NSDictionary class associates textual keys to corresponding value objects and returns the required object without iteration. The debugger output demonstrates the preference dictionary of a sample user (Code snippet 1: User preference as NSDictionary. The outer dictionary has a total of eight sub-dictionaries that stand for all restaurant attributes. Furthermore, each sub-dictionary contains an array of all possible attribute values. Binary attributes are represented by a dictionary that holds an array of two elements. The first element corresponds to preference value *NO* and the second element is the value *YES*. Attributes with higher cardinalities are represented with the help of helper classes, which are again sorted in arrays on their preference values. For instance, the second dictionary holds a key named "category" that holds an array of 19 elements, where each element holds the category name and category preference value for the active user.

```
(NSMutableDictionary *) $0 = 0x0000000105b2a7f0 {
    CarPark =     (
        "7.496392726898193",
        "7.62640905380249"
    );
    Category =     (
        "Name:Fast Food Rest , Value 8.037037",
        "Name:Running Sushi , Value 7.916667",
        "Name:Pizzeria , Value 7.901235",
                       . . .
        "Name:Counter Service , Value 7.203704",
        "Name:Cantina , Value 7.064815",
        "Name:Bistro , Value 6.937037",
);
    ChildFriendly =     (
        "7.526196956634521",
        "7.526749134063721"
    );
    Cuisine =     (
        "Name: American , Value 8.222222",
        "Name: Chinese , Value 8.111111",
        "Name: Mexica , Value 8.049383",
                       . . .
        "Name: French , Value 7.227053",
        "Name: Spanish , Value 6.888889",
        "Name: Argentina , Value 6.574074",
);
    Garden =     (
        "7.394394397735596",
        "7.90170955657959"
    );
    LiveMusic =     (
        "7.385361671447754",
        "8.266203880310059"
    );
    PriceRange =     (
        "Normal : 7.822222",
        "Low : 7.754630",
        "Above Average : 7.438998",
        "Luxus : 6.589744"
    );
    Smoking =     (
        "NO : 7.800926",
        "Seperated : 7.651652",
        "YES : 7.238367"
    );
}
```

Code snippet 1: User preference as NSDictionary

$$R^{*}_{Restaurant\ R} = \begin{bmatrix} [R_{Cuisine}] \\ [R_{Category}] \\ [R_{Price}] \\ [R_{Smoking}] \\ [R_{Car\ Park}] \\ [R_{Childfriendly}] \\ [R_{Live\ music}] \\ [R_{Gardern}] \end{bmatrix} x \begin{bmatrix} W_{Cuisine} & \cdots & W_{Garden} \end{bmatrix}$$

Formula 4-3 : Rating prediction

As mentioned earlier, individual attribute values in the preference dictionary own a preference value that represents the overall ratings of all restaurants with that attribute value. As seen in the Formula 4-3 : Rating prediction, $[R_{Attribute}]$ stands for preference value representation of the restaurant attribute, which could be found in the user preference dictionary. This representation is self-explaining and the active user could reason individual recommendations by investigating the corresponding attribute preference values.

As mentioned earlier, generation of recommendation items corresponds to finding a proper rating prediction for restaurant items considering their attributes values. Individual attribute values of restaurant items are matched with preference values of attributes in the user model. In this context, the recommendation engine does not require any similarity between restaurants.

5 Offline Experiment

The offline experiment part of the thesis targets generation of test data as input for the model and evaluation of the recommendation results. This section describes synthetic data generation, which is required for recommendation model evaluation discussed in the next section. All recommendation items (restaurants) and service customers (users) are created randomly by a data set generation algorithm that fulfills certain constraints to be able to create proper user prototypes and categorized restaurants.

Through an offline experiment it is possible to measure the accuracy of predicted recommendation ratings. A recommendation algorithm computes recommendations with hidden user ratings, and afterwards prior known ratings are compared with the generated rating values.

As mentioned above, data entities of the restaurant recommendation model are mainly users, restaurants, and five-dimensional restaurant ratings with meal, service, tangibles, personnel, and accessibility dimensions. The synthetic input data is generated by following a set of linear rules. These constraints define users' rating behavior, preferences, and interests. Systematic generation of the test data is necessary since random generated data would result in the absence of patterns and the recommendation engine would fail to predict reasonable recommendations. In real world scenarios, users can like and dislike entities of the same category, but in general terms each user has a sense of taste and a tendency to only a limited number of preference values. For instance, if it is known that a customer is a non-smoker, it is understandable that he rates non-smoking restaurants positive or slightly better than smoking places. Under random data generation, the user has the same opportunity to like a smoking and non-smoking restaurant, and it is not possible find out any pattern about the attitude of the active user against smoking.

The data generation algorithm assumes that restaurant customers could be grouped into various user segments as demonstrated in Table **5-1**: Restaurant customer segments for prototype generation. Users of the same stereotype are supposed to have similar expectations and preferences. These expectations can be formulized as simple *if – else* rules. The goal of creating stereotypes and assigning certain constraints is to be able to fill the utility matrix systematically with reasonable ratings. It should be noted that these stereotypes are not taken into calculation in the recommendation prediction algorithm. User prototypes are required for the generation of restaurant ratings in a systematic manner.

The proposed model does not consider general problems of recommendation systems, including challenges in explicit or implicit user feedback collection, data sparsity, or cold start problems. Additionally, required restaurant attributes are considered to be previously known by the system. A utility matrix is considered to be non-sparse in order to evaluate prediction results with accuracy measures.

Segment	Assumptions about restaurant rating generation
Students	Lower price range is preferred. 90% without car. Without children and prefer non-child friendly restaurants. 50% smoker and smoking attribute is an important preference. Live music elevates ratings.
Ambiance lovers	Tangibles are important for them. Presence of car park, live music, and garden elevate their ratings. Tendency toward high priced restaurants. They like unique places like ouzeria, brew pub, osteria, or fine dining.
Gourmets	Cuisine and category is not significant and they are open to different tastes. Luxury restaurants are preferable. Ratings are very "*sharp*" (higher level of rating randomness). Smoking restaurants are not even considered.
Families	Travel with car and require car place. If they have children, child friendliness is an important preference. Tendency toward non-smoking restaurants.
Tourists	Like mostly local cuisines in heuriger, coffee house, etc. Prefer comfortable places like food restaurants, pizzerias, or counter service. No restrictions about price range.

Table 5-1: Restaurant customer segments for prototype generation

Additionally, the data generator also defines certain constraints for generated restaurants. Restaurants are clustered into classes based on their cuisine and category and other attribute values information. Firstly, generated restaurant entities are labeled with 19 distinct category values. Secondly, 16 distinct cuisine values are assigned to these restaurant entities with known category values following the rules described in the cuisine-category constraint table. Restaurant categories are matched with possible cuisine values randomly. This consideration is optional, but is necessary to simulate realistic restaurant category-cuisine matching. For instance, matching a restaurant with category running sushi with Irish would not be realistic.

Category	Cuisine			
Bakery	Austrian	French	Turkish	
Bistro	French			
Brew pub	Austrian	Irish		
Cantina	Austrian	Italian	French	Spanish
Coffee house	Austrian	Turkish		
Counter service	World			
Fast food restaurant	American	World		
Fine dining	French			
Food court	World			
Heuriger	Austrian			
Osteria	Italian			
Ouzeria/Tavern	Greek			

Pizzeria	Italian			
Pub	Austrian	English	Irish	
Running sushi	Chinese	Japan	Korea	
Sea food restaurant	Greek	Italian	Spanish	Turkish
Snack bar	Austrian	Chinese	Thailand	Turkish
Steak house	Argentinian	Mexican		
Take-out	Austrian	Chinese	Italian	Turkish

Table 5-2: Restaurant cuisine-category matching

In addition to restaurant category-cuisine matching, remaining restaurant attributes are generated following further simple rules demonstrated in Table 5-3 : Conditional probabilities of price given category.

Category	**Price**			
	Low	Normal	Above average	Luxury
Bakery	% 25	% 50	% 25	% 0
Bistro	% 25	% 50	% 25	% 0
Brew pub	% 25	% 50	% 25	% 0
Cantina	% 0	% 60	% 20	% 20
Coffee house	% 0	% 60	% 20	% 20
Counter service	% 25	% 50	% 25	% 0
Fast food restaurant	% 20	% 60	% 20	% 0
Fine dining	% 0	% 30	% 35	% 35
Food court	% 20	% 50	% 30	% 0
Heuriger	% 20	% 30	% 30	% 20
Osteria	% 20	% 30	% 30	% 20
Ouzeria/Tavern	% 20	% 30	% 30	% 20
Pizzeria	% 20	% 50	% 20	% 10
Pub	% 30	% 35	% 23	% 12
Running sushi	% 20	% 30	% 30	% 20
Sea food restaurant	% 0	% 20	% 30	% 50
Snack bar	% 40	% 60	% 0	% 0
Steak house	% 0	% 30	% 30	% 40
Take-out	% 30	% 49	% 21	% 0

Table 5-3 : Conditional probabilities of price given category

Category	**Smoking**		
	NO	YES	Separated
Bakery	% 40	% 60	% 0
Bistro	% 50	% 50	% 0
Brew pub	% 5	% 70	% 25
Cantina	% 40	% 40	% 20
Coffee house	% 20	% 60	% 20
Counter service	% 30	% 30	% 40
Fast food restaurant	% 90	% 5	% 5
Fine dining	% 100	% 0	% 0

Food court	% 100	% 0	% 0
Heuriger	% 0	% 50	% 50
Osteria	% 0	% 70	% 30
Ouzeria/Tavern	% 0	% 50	% 50
Pizzeria	% 50	% 20	% 30
Pub	% 0	% 80	% 20
Running sushi	% 70	% 0	% 30
Sea food restaurant	% 20	% 40	% 20
Snack bar	% 0	% 100	% 0
Steak house	% 50	% 0	% 50
Take-out	% 50	% 50	% 0

Table 5-4 : Conditional probabilities of smoking attributes given category

Category	Attribute			
	Live Music	Garden	Child friendliness	Car park
Bakery	% 0	% 25	% 0	% 10
Bistro	% 0	% 10	% 0	% 10
Brew pub	% 30	% 25	% 0	% 10
Cantina	% 15	% 10	% 30	% 10
Coffee house	% 10	% 30	% 30	% 20
Counter service	% 0	% 10	% 50	% 10
Fast food restaurant	% 0	% 60	% 50	% 10
Fine dining	% 35	% 30	% 35	% 30
Food court	% 0	% 30	% 50	% 40
Heuriger	% 70	% 80	% 80	% 70
Osteria	% 15	% 30	% 30	% 10
Ouzeria/Tavern	% 70	% 30	% 0	% 20
Pizzeria	% 0	% 30	% 50	% 10
Pub	% 30	% 30	% 0	% 10
Running sushi	% 10	% 10	% 50	% 20
Sea food restaurant	% 20	% 40	% 30	% 40
Snack bar	% 0	% 0	% 0	% 0
Steak house	% 15	% 20	% 30	% 30
Take-out	% 0	% 0	% 0	% 10

Table 5-5 : Conditional probabilities of Boolean attributes given category

Once the synthetic restaurants are created, user ratings are generated in regards to the predefined preferences of user stereotypes as described in Table 5-6 : Rating generation rules. During rating generation, restaurant attributes are compared with predefined user stereotype preferences. Each restaurant attribute is evaluated with a rating of 0 to 10, which is then assigned to one of the service dimensions. Restaurant attributes and service dimensions have different weight factors depending on the stereotype. In order to create a level of randomness in user ratings, a dimension penalty is added or subtracted from the generated ratings depending on the result of the random number generator and user stereotype.

Stereotype	Child friendly		Garden		Live Music		Car park	
	YES	NO	YES	NO	YES	NO	YES	NO
Students	0	10	10	8	10	6	10	4
	0	10					7	7
Ambiance lovers	0	10	10	5	10	3	10	0
	0	10					9	7
Gourmets	0	10	10	8	10	6	10	4
	0	10					7	7
Families	10	0	10	8	10	8	10	4
	4	7					7	7
Tourists	10	10	10	8	10	9	10	4
	10	8					10	10

Table 5-6 : Rating generation rules based on Boolean attributes

Stereotype	Price Garden				Smoking		
	Low	Normal	Above average	Luxury	YES	NO	Separated
Students	10	9	7	3	10	6	10
					0	10	10
Ambiance lovers	7	8	10	10	10	6	10
					0	10	10
Gourmets	0	8	9	10	0	10	6
					0	10	6
Families	3	8	10	7	10	7	9
					0	10	9
Tourists	8	10	9	5	10	7	9
					5	10	9

Table 5-7 : Rating generation rules based on multi-valued attributes

Once individual restaurant attributes are evaluated with ratings depending on the given linear rules, they are assigned to corresponding service dimensions as described in the following table. The final value of the service dimension ratings is determined by the rating weights and randomness value of the individual stereotypes that vary on each rating generation.

Dimension	Attributes			
Accessibility	Child friendliness	Smoking	Car park	
Core service	Category	Cuisine	Price	
Personnel	Category	Price		
Service	Category	Price	Live Music	
Tangibles	Car park	Garden	Live Music	Child friendliness

Figure 5-1 : Assignment of individual attribute ratings to service rating dimensions

6 Evaluation of the Proposed Model

Recommended Restaurants

Restaurant	Category	Cuisine	Place	Price	Smoking	Garden	Music	Car Park	Child Frdly.	Rating	Ranking	Rating*	Ranking*
141	Heuriger	Austrian	11	Above...	NO	YES	YES	YES	YES	9,5	10	7,5	1
445	Heuriger	Austrian	6	Above...	NO	YES	YES	YES	YES	9,3	26	7,5	2
829	Heuriger	Austrian	12	Above...	NO	YES	YES	YES	YES	9,7	6	7,5	3
54	Heuriger	Austrian	3	Above...	NO	YES	NO	YES	YES	9	36	7,5	4
341	Heuriger	Austrian	10	Above...	NO	YES	NO	YES	YES	9,5	15	7,5	5
356	Heuriger	Austrian	6	Above...	NO	YES	NO	YES	YES	9,5	12	7,5	6
292	Heuriger	Austrian	10	Luxus	NO	YES	YES	YES	YES	9,7	5	7,4	7
783	Heuriger	Austrian	15	Normal	NO	YES	YES	YES	YES	9,2	35	7,4	8
877	Heuriger	Austrian	13	Normal	NO	YES	YES	YES	YES	9,5	16	7,4	9
985	Heuriger	Austrian	6	Normal	NO	YES	NO	YES	YES	9,6	9	7,4	10
326	Heuriger	Austrian	2	Low	NO	YES	YES	YES	YES	9	40	7,4	11
437	Heuriger	Austrian	7	Low	NO	YES	YES	YES	YES	9,3	23	7,4	12
643	Heuriger	Austrian	15	Low	NO	YES	YES	YES	YES	8,7	44	7,4	13
928	Heuriger	Austrian	1	Low	NO	YES	YES	YES	YES	8,8	42	7,4	14
942	Heuriger	Austrian	9	Low	NO	YES	NO	YES	YES	8,6	50	7,4	15
900	Heuriger	Austrian	10	Low	NO	NO	YES	YES	YES	9,2	33	7,4	16
146	Food Co...	World	11	Normal	Sepe...	YES	NO	YES	YES	9,7	4	7,3	17
57	Heuriger	Austrian	3	Above...	YES	YES	YES	YES	YES	9,3	25	7,3	18
770	Heuriger	Austrian	15	Above...	YES	YES	YES	YES	YES	9,3	27	7,3	19
827	Heuriger	Austrian	7	Above...	YES	YES	YES	YES	YES	9,5	19	7,3	20
870	Heuriger	Austrian	2	Above...	YES	YES	YES	YES	YES	9,3	28	7,3	21
881	Heuriger	Austrian	14	Above...	YES	YES	YES	YES	YES	8,9	41	7,3	22
984	Heuriger	Austrian	4	Luxus	YES	YES	YES	YES	YES	8,3	80	7,3	23

Figure 6-1 : Recommendation presentation of proposed application

The proposed model is a simplified content-based recommendation engine that could be adjusted over multiple parameters including rating dimension weights and short-term utility weights to cope with service customer requirements. As mentioned, the recommendation model shows predicted recommendation results in a ranked list as seen in Figure 6-1 : Recommendation presentation. The rank of the recommendation restaurant in the list corresponds to the relevance level of the item among the whole set of items.

Due to the limitations of the offline experiment, it is not possible to measure the influence of the recommendation engine on the decision of the active user. Therefore, the evaluation part of the thesis mainly focuses on the prediction accuracy of the proposed model. The experiment does not cover the problem of data sparsity or cold start of expertise systems. Evaluation results represent the accuracy of the proposed model based on multiple instances of a synthetic data set. Given results are the average of multiple test iterations and could vary depending on the generated synthetic data.

The following table gives an overview of the development and test environment of the recommendation application.

Development & test computer	Mac Book Pro 2.4 GHz Intel Core 2 Duo, 8 GB Ram
Opera System	Mac OS X Lion 10.8.2
Development Language	Objective – C
IDE	*XCode* 4.5.2
Persistent data model	Core Data with XML persistent store
Required external library	GSL – GNU scientific library

Table 6-1 : Recommendation application development and test environment

Gunawardana and Shani (2009) introduce various dimensions and properties of recommendation systems as evaluation aspects and comparison of different recommendation techniques. Some of these properties are used as a test benchmark for the evaluation of the proposed model in this section in the scope of the offline experiment.

User preference and weight calculation
Rating prediction accuracy
Coverage
Confidence
Trust
Novelty
Serendipity
Diversity
Utility
Risk
Privacy
Adaptively
Scalability

Table 6-2: Properties of recommender systems (Gunawardana & Shani, 2009)

6.1 User Preference and Weight Calculation

One of the most important properties of recommender systems is how the user preferences are calculated and processed. Additionally, a reasonable recommendation requires assignment of proper preference weights in addition to preference calculation. The proposed model assigns preference scores to restaurant attributes as user ratings considering the number of past ratings and the average of these ratings.

Preference values and preference weight factors of the active user are computed by taking all rated items into account[18]. Rating scoring assumes that the user rates restaurant items that he is not interested in with lower ratings and more preferable restaurants with higher scores. In real world scenarios, users mostly do not rate items that they have not experienced so far. It may be valuable information for other users or the recommendation

[18] Calculation of preferences and corresponding weights is described briefly in section 4.4.

system itself to know why the active user rates a certain item very high, or in the opposite case very low. This would be a very important improvement for the user preference and weight calculation since *unsystematic*, exceptionally low ratings decrease user preference values of individual attributes. In the proposed model, the user has the ability to rate a restaurant with five different dimensions so that if one of the service dimensions did not satisfy the user, he does not have to rate the service performance of the other dimensions with a low rating. The proposed system iterates past ratings with all dimensions to compute primarily overall rating and use this overall rating to compute preference values. In exceptional cases, one of the five rating dimensions could be an outliner value, which could elevate or decrease the overall rating.

Preference weights are very significant for the proposed model since their value determines the final rating prediction. Individual weight values are calculated by considering the linear correlation or Cramer's V value of ratings with attribute values and entropy of restaurant attributes. The proposed model uses a scale factor to eliminate domination of attributes with higher entropy values.

6.2 Rating Prediction Accuracy

Prediction accuracy is considered to be one of the most important properties of recommendation engines and their underlying recommendation algorithm. In the case of collaborative filtering, accuracy is related with number of users and total available ratings. In content-based recommendation systems with dynamic utility settings, number of rated items does not influence the accuracy of the predictions. However, users need a high enough number of ratings so that their preference model can be calculated properly. Rating prediction accuracy can be measured in the scope of an offline experiment by comparing the prior known user ratings with the predicted ratings of the recommendation engine.

Accuracy

$$\boldsymbol{RMSE = \sqrt{\frac{\sum(R *_{u,i} - R_{u,i})^2}{number\ of\ observed\ ratings}}}$$

Formula 6-1: Root Mean Squared Error

$$\boldsymbol{MAE = \sqrt{\frac{\sum\left|R *_{u,i} - R_{u,i}\right|}{number\ of\ observed\ ratings}}}$$

Formula 6-2: Mean Absolute Error

Root Mean Squared and ***Mean Absolute Error*** are commonly used measures to denote recommendation accuracy. Compared to *MAE*, *RMSE* penalizes large errors. These measures depend on the magnitude of the errors made. Recommendation engines can have various rating semantics and value ranges. For instance, an RMSE of 2 can be still understandable in a rating scale of 10, while a model with a five-point rating system cannot be considered as reasonable with the same RMSE value (Gunawardana & Shani, 2009).

Number of Restaurants	RMSE	MAE
100	0,7794	0,7875
500	0,7478	0,7743
1000	0.8465	0.8227

Table 6-3: RMSE and MAE evaluation of proposed model with rating range of 10

Usage prediction

In many applications, recommendation systems do not assign numeric or categorical values to recommendation items to represent the level of interest. This approach is especially important for domains where the ranking of the recommendation item is not relevant. Such systems present only potentially interesting items and need to filter out irrelevant items. Due to that, the model needs to limit the number of recommendation items presented to the user. The recommendation model can take the highest N recommendation results, or all recommendation results that have prediction values higher than a certain threshold. A recommendation result can be classified as *true positive* if it is interesting for the active user and the model represents it to the user. *True negative* recommendations are the ones that are filtered out, and as expected are not interesting for the user. The undesired case *false positive* occurs if the model shows an uninteresting item to the user. The case *false negative* occurs if an interesting item is filtered out.

	Recommended	Not recommended
Interesting for user	True positive (tp)	False negative (fn)
Uninteresting for user	False positive (fp)	True negative (tn)

Table 6-4: Recommendation result classification of an item for active user

$$Accuracy = \frac{N_{True\ positive} + N_{True\ negatice}}{Total\ recommendation\ items}$$

Formula 6-3: Accuracy

To simulate the recommendation result classification in the eyes of synthetic users, a threshold value is used to indicate zone of tolerance for the active user. All restaurants with predicted ratings above the threshold value are considered to be interesting, and the ones below are filtered out. Through this approach, it is possible to denote the number of recommendation results that fall into each recommendation result category.

Usage prediction can be expressed over the measures *precision*, *recall,* and *false positive rate*. *Precision* is an important measure *that* defines the ratio of relevant items to selected items.

$$Precision = \frac{N_{For\ user\ relevant\ Items}}{N_{Selected\ Items}} = \frac{N_{True\ positive}}{N_{True\ positive} + N_{False\ positive}}$$

Formula 6-4: Precision

Recall denotes the ratio of selected relevant items to total number of relevant items.

$$Recall = \frac{N_{For\ user\ relevant\ Items}}{N_{Total\ relevant\ items}} = \frac{N_{True\ positive}}{N_{True\ positive} + N_{False\ negative}}$$

Equation 6-1: Recall

There exists a trade-off between precision and recall. An increased number of recommended items would improve the recall but also reduce the precision. The following tables show the prediction accuracy of the proposed model by considering precision and recall values under different positive threshold values.

Number of Restaurants	***Precision***	***Recall***
100	0.948052	0.196970
500	0.962264	0.153153
1000	0.666667	0.144649

Table 6-5: Precision/recall values within a defined positive rating threshold (8/10)

Number of Restaurants	***Precision***	***Recall***
100	0.830303	0.938356
500	0.800878	0.932312
1000	0.794234	0.803757

Table 6-6: Precision/recall values within a defined positive rating threshold (7/10)

As seen in the tables above, the recall value is relatively low when the positive rating threshold is eight. When the threshold value is set to seven, recall value increases dramatically. This is related to the underestimation of the user ratings by the recommendation system. In long-term positive ratings near the value, eight remains under the threshold and as a consequence the number of false negative values increases. Similarly, this decreases the number of true positive ratings and in the long run this ends up with a low recall value. When the positive threshold value is set to a lower value like seven, the number of false negatives is relatively low so that the recall value increases.

F_1-Score or F-Measure is another term to denote accuracy performance in the field of information retrieval. F_1-Score is the harmonic mean of precision and recall that could be interpreted as a weighted average of them.

$$F = 2 * \frac{presicion * \ recall}{presicion + recall}$$

Formula 6-5: F-Measure

6.3 Ranking of Recommended Items

The proposed model presents the recommendation results as a vertical list of items. Despite shown predicted rating values, items presented at the top of the list attract more attention compared to items shown in lower positions.

For the evaluation of a recommendation system considering ranking of the presented items, one requires a reference ranking for all recommended items. In the context of the restaurant recommendation model, real ranking of the restaurant corresponds to the position of the mentioned item in the descending sorted array based on the overall rating.

In other words, the highest rated restaurant gets the highest ranking and the least preferred one gets the lowest ranking. Restaurants that share the same rating are considered tied.

Normalized Distance-Based Preference Measure (NDPM) introduced by Yao (1995) denotes the item ranking performance of models by comparing original and predicted rankings as item pairs.

$$NDPM = \frac{C^{-} + 0.5C^{u0}}{C^{u}}$$

Formula 6-6: NDPM score

C^{-} is the number of item pairs whose rankings assert the wrong order. An item pair falls into this group if an item *i* is shown in front of its pair *j* while according to user ratings the positioning should be inverse. C^{u0} is the number of pairs that have tied ranking according to user ratings but are not represented as tied in recommendation representation. C^{u} denotes the total number of pairs in the data set. An ideal system is expected to have an NDPM value of 0 where all items are shown in the right place.

Number of Restaurants	*NDPM*
100	0,435786
500	0.444280
1000	0.460481

Table 6-7: NDPM score of proposed model

6.4 Coverage

Coverage is a measure that exposes the proportion of recommendation items that can be discovered by the model and recommended to the user. In collaborative filtering, coverage is similar to accuracy in the richness of user profile, and additionally with distribution of users with similar preferences. In other words, unique items with too few ratings will not be discoverable due to the popularity bias problem of CF, and this will result in lower coverage. To avoid the cold start problem, some CF systems like *movielens* can ask their user initially to rate a minimum number of items before starting to generate recommendations. The cold start problem not only concerns the number of given user ratings, but also newly added items.

In the proposed model, the discoverability of individual items is related directly to proper user utility settings and preference weights. Default preferences and weights are calculated based on past ratings, but the user can change his preferences anytime and the recommendation result will cover different items with different rankings. A cold item does not pose a problem as long as it has complete attribute values. The mentioned experiment does not consider the scenario that a new restaurant is added to the system with certain missing attribute values. In this case, the missing values could be predicted with the help of a proper classifier algorithm.

6.5 Novelty and Serendipity

As mentioned in section 2.2.6.2, a recommendation model is novel if it can suggest an unknown item to the active user that he might have discovered by himself. In the case of restaurant recommendation, if the active user has rated many restaurants with same category and cuisine, a recommended restaurant with these attributes is considered novel. From the perspective of the user, this recommended item would not be surprising. Content-based recommendation aims to overcome the problem of popularity bias and targets individual unique preferences or interests of service customers. Through this approach, it generates novel recommendations, but a user never gets a relevant surprising item as a recommendation (which is known as the serendipity problem).

In general terms, serendipitous recommendations run against the underlying algorithm of content-based systems. CBRS tries to find relevant items with possible small distances to the user preference, but a serendipitous item needs to have a predefined distance to prior recommended items or to the preference model of the user. This drawback is considered to be an important weakness of CBRSs. The proposed model generates recommendations with the content-based approach, but the utility-based module has the ability to change the short-term preferences and generate different recommendation results. An active user could set the weight factors of his preferences or refine restaurant attribute utilities. Even though the user would be aware of the changes that he applied to his utility settings, generated recommendations cannot be interpreted as serendipitous.

6.6 Scalability

Recommendation models also need to be tested in terms of computation time and complexity regarding memory and space. The following table demonstrates the computation performance of the proposed model in time space for any application user. User preference calculation assumes that all given user ratings are taken into consideration. The number of recommendation items is not limited by any constraints so the user gets all available restaurants as a ranked list.

Number of Restaurants	***User Preference calculation***[19]	***Recommendation generation***[19]
100	0.1036	0.4025
500	0.3281	10.7029
1000	0.6945	38.1950

Table 6-8: Computation time in seconds for

In the scope of the offline experiment, the recommendation application computes the user preferences in volatile memory, and required persistent data including users, restaurants, and user ratings are stored in the *Core Data* framework due to its simplicity ("Introduction to Core Data Programming Guide", 2012).

In an enterprise recommendation application, a relational database to hold persistent data with corresponding Data Access Objects and internal caching mechanism would return better results in terms of time.

[19] In seconds, tested on device given in Table 6-1 : Recommendation application development and test environment

7 Future Work

One of the most important drawbacks of the proposed model is the over-specification of the recommendation results. In order to overcome this limitation, the proposed model could be extended by a knowledge based-recommendation module. In terms of restaurant recommendation, it could track similarities between cuisines and restaurant categories independent of user ratings and generate additional recommendations. Through this approach, the model could achieve a level of randomness. This approach would require knowledge engineering on the cuisine and category data set, but the generated recommendation would remain reasonable in contrast to the collaborative filtering option. For instance, it is possible to calculate similarities between restaurant cuisines in a generic manner by considering gastronomic attributes including popular spices, geographical location, countries, or number of shared meals.

Another possible attempt to achieve serendipitous recommendation could be combining the content-based model with an extensional collaborative filtering. As a subject of future research, social filtering could also be used to overcome the over-specification problem by expanding the recommendation result set with preferable restaurants of similar users according to social network connections of the active user. Additionally, restaurants could be evaluated with regard to demographic or locational factors. Rather than keeping track of demographic information, a recommendation model could patch required information from an external source like a social networking site. Through the same approach, similarity could also be extended with the closeness level in the social network. For instance, a Facebook friend with whom a user has been tagged multiple times in different hotels would probably have similar interests of accommodation.

The collaborative filtering approach gives a level of randomness to the recommendation engine but could decrease the accuracy of the results. Optionally, the recommendation model could assign a confidence value to its recommendations to determine the system's trust in its predictions to inform the active user about random generated recommendations informally.

In an era of smart phones and mobile applications, location-aware mobile devices could be used as an important source of information for implicit rating interpretation. A recommendation model with a mobile front end could make location aware suggestions or track active user interactions with the device to derive implicit likes, including posting pictures to social networking sites, tagging friends in the same location, and so on. In terms of the mobile recommendation approach, restaurant items could be presented on a map based interface rather than ranked lists (Ricci, 2010).

The proposed restaurant recommendation model was evaluated in the context of an offline experiment considering primarily the recommendation accuracy. Unfortunately, due to the absence of user interactions and feedback, it is not possible to measure the recommender's influence on user behavior. Based on the findings about the accuracy of the model, ***RMSE*** could be considered relatively low. This value would increase in the case of an online experiment with a sparse utility matrix with a few thousand recommendation items rated by real system users.

This is related to the heterogeneous characteristic of services, which make individual service items unique even if they are modeled same considering their attribute values. In other words, a user might have rated a big percentage of restaurants with certain attributes very high, while he could have rated other items with the exact same attribute values low. Those items with lower ratings decrease the overall utility score of restaurants with the same attribute values. As an extension for the proposed model, outliner rating dimension values could be kept out if a certain rating pattern is detected. This approach assumes that individual dimension rating values are inspected before being taken into preference calculation. The system could detect a pattern in rating dimensions and take measures to avoid the undesired effects of deterministic rating values. For instance, if the model detects that personal rating is relatively low compared to other dimensions, this value could be exceptionally ignored. In other words, a service employee that made a customer happy or unhappy with his service should not influence his cuisine preference in the system. Apart from that, for outliner ratings an additional description could be requested from the user. If the user explains what has gone so well or poorly during service delivery, this information could be considered to update the user preference, or shared with other users in a possible CF approach.

Another option to increase the accuracy of the model in the scope of an online experiment could be letting the active user give relevance feedback to improve recommendation results. If one of the already rated items is shown intentionally on the list and active users could correct its overall rating, this corrected valued could be used to refine preference utility score, which would produce more precise rating values.

8 Conclusion

Services domains have many unique characteristics that make them different than goods. As a consequence of these special features, service recommendation systems have certain requirements in the modeling, user preference building, and recommendation generation phases. Recommendation techniques used for recommending tangible goods need to be adapted and extended in many dimensions in order to cope with unique service characteristics. This master's thesis focuses on generic service features that need explicit consideration during service recommendation, as well as additional roles of service recommendation systems for service customers.

The state of art part of the thesis reviews available recommendation techniques and their possible application fields, including their advantages and weaknesses. Based on the findings, a generic recommendation model is proposed that is evaluated in the restaurant recommendation domain. A generic model assumes that service quality perception is a long-lasting process that is distributed to multiple dimensions, and these dimensions should not be aggregated to a single rating. This requirement comes from the process nature of services. Evaluation of the whole service with a single numeric rating cannot represent user preferences and perception of quality from multiple service dimensions. The overall service quality consists of multiple dimensions, so these should be denoted separately in the user rating. Due to the shortcomings of single dimensional ratings, the proposed model assumes that services need to be evaluated over multiple dimensions including accessibility, tangibles, delivery duration, and service personnel. A recommendation model with a multi-criteria rating system is more informative for the active user and optionally for other system users. Additionally, multi-criteria ratings can show possible weaknesses and strengths of service delivery to optimize or maintenance for service providers. The multi-criteria ratings are important requirements of services since different service dimensions need to be evaluated separately. The evaluated model demonstrates important dimensions of restaurants that can also vary depending on the application domain. Dynamic rating dimension weights make the model flexible for different customer types. Unique service customers do not only have a unique preference model, but also different priorities about different dimensions of restaurant ratings. Service blueprints can be used to derive important service touch points and dimensions to determine the distinct rating dimensions for the recommendation system.

In order to individuate unique restaurant customer preferences, the proposed model uses a content-based recommendation technique and investigates restaurant attributes to build the user preference model. Content-based recommendation systems generate rating predictions by taking various significant features of items into calculation and exploring similar items that the active user has liked so far. Collaborative filtering ignores the content of the items and estimates similar items depending on the ratings of similar users. Due to the nature of collaborative filtering, the active user is bound to other user ratings, and users with unique preferences cannot perceive novel recommendations. As mentioned, service philosophy assumes that individual service customers should be treated separately. Thus, building user preferences based on similar user ratings would violate the customer centric service philosophy.

Another important reason for the application of the content-based recommendation technique to the service model is that recommendation results, which can be reasoned and explained by the model, will increase trust in the system. This would not be possible in collaborative filtering, since recommendation results remain as a black box. Reasonable recommendation results are informative and helpful for the user to make decisions about heterogeneous service items.

To accomplish changeable user preferences in the short term, model gives the active user the ability to change his default preferences and preference weights, which are computed in normal operation mode considering past user ratings for preferences. This ability could be considered to be a utility-based recommendation extension for the mentioned content-based approach. In general terms, utility-based models are preferable in many service domains, including accommodation, flight, and other restaurant recommendation systems.

Apart from that users, without enough past ratings do not suffer the cold start problem of content-based recommendation systems since they can determine the utility function considering their own needs.

Due to their inseparable nature, services cannot be refunded, so the reliability of the service provider becomes even more significant for the service customers. In addition to their traditional roles, service recommendation systems need to build trust in the service provider in the eyes of service customers. Intangibility of services means the absence of material product, but on the other side, certain services have various tangible components that can be used to reflect company image or reputation. As mentioned earlier, services might have common attribute values.

A recommendation model considers two recommendation items to be the same if they share the same attribute values. Unfortunately, this modeling approach violates the variable nature of services, given that the restaurant recommendation model contains many restaurant entities with the same attribute values that could be rated differently by the same test user. Service customers can even perceive the service value of the same service provider differently in different points of time due to various uncontrollable factors. Therefore, enterprise service recommendation systems can confirm user ratings with textual feedback to denote the reason for satisfaction or dissatisfaction.

9 Bibliography

Avesani, P., Massa, P., & Tiella, R. (2004). Moleskiing: a Trust-Aware Decentralized Recommender System.

Adomavicius, G., Manouselis, N., & Young, O. K. (2011). Multi-Criteria Recommender Systems. In *Recommender Systems Handbook.*

Anand, R. K. & Jeffrey, D. U. (2011). *Mining of Massive Datasets.* Stanford Univ.

Anderson, C. (2011). Recommender systems for e-shops.

Anonymous. (2012, 10 20). *Experience evaluation form.* Retrieved from http://theocexperience.com/yahoo_site_admin/assets/docs/The_Experience_Evaluation_Form.318112728.pdf

Arlen, C. (2008). *The 5 Service Dimensions All Customers Care About.* Retrieved from http://serviceperformance.com/the-5-service-dimensions-all-customers-care-about/

Burke, D. R. (2002, 11). Hybrid Recommender Systems: Survey and Experiments. *12* (4), pp. 331–370.

Burke, D. R. (2000). Knowledge-based recommender systems.

Baker, K. (2005). Singular Value Decomposition Tutorial.

Bitner, M. J. & Booms, B. H. (1981). Marketing strategies and organisation structures for service firms. *American Marketing Association* .

Bitner, M. J., Booms, B. H., & Stanfield, T. M. (1990). The Service Encounter: Diagnoising Favorable and Unfavorable Incidents. *Journal of MArketting, 54*, 71–84.

Bitner, M. J., Ostrom, A. L., & Morgan, F. N. (2007). Service Blueprinting: A Practical Technique for Service Innovation.

Bhasin, H. (2010, 09 19). *Heterogeneity in service marketing.*

Brady, M. K. & Cronin J. Jr. (2001). Some New Thoughts on Conceptualizing Perceived Service Quality: A Hierarchical Approach. *The Journal of Marketing , 65* (3), 34–49.

Brown, S., Gummesson, E., & Edvardsson, B. (1991). *Service Quality Multidisciplinary and Multinational Perspectives.*

Curry, A. & Brysland, A. (2001). Service improvements in public services using SERVQUAL.

Czepiel, J. A., Solomon, M. R., & Surprenant, C. F. (1985). The service encounter: Managing employee/customer interaction in service businesses.

Chackraborty, R. (2010). *Fundamentals of Neural Networks.* Artificial Intelligence lecture notes.

Chase, R. B. & Dasu, S. (n.d.). Psychology of the Experience: The Missing Link in Service Science.

Chesbrough, H. & Spohrer, J. (2006). A Researh Manifesto for Services Science. *Cummunication of ACM, 49* (7).

Clatworthy, S. (2011). Service Innovation Through Touch-points: Development of an Innovation Toolkit for the First Stages of New Service Development. *International Journal of Design , 5* (2).

Duan, L., Street, W. N., & Xu, E. (2011, 05 02). Healthcare information systems: data mining methods in the creation of a clinical recommender system.

Dabholkar, P. A., Thorpe, D. I., & Rentz, J. O. (1996). A measure of service quality for retail stores: Scale development and validation. *Journal of the Academy of Marketing Science*.

Debnath, S., Ganguly, N., & Mitra, P. (2008). Feature Weighting in Content Based Recommendation System Using Social Network Analysis.

Dongmin, C., & Yau, S. S.-T. (2005). Categorical Clustering by Converting Associated Information.

Fuller, R. (2010). The perception Learning Rule Tutorial.

Guzzoni, M.-L. (2005). Models of Service Marketing – A Study on the Marketing Functions of Project Workers in Knowledge.

Gunawardana, A. & Shani, G. (2009, 12). Evaluating Recommendation Systems. *Journal of Machine Learning Research*.

Gummesson, E. (2007). Exit Services Marketing – Enter Service Marketing. *The Journal of Customer Behaviour , 6* (2), 113–141.

Gales, M. (2011). Single Layer Perceptron & Estimating Linear Classifiers.

Galkin, I. (n.d.). Crash Introduction to Artificial Neural Networks.

Garden, M. & Dudek, G. (2005). Semantic feedback for hybrid recommendations in Recommendz. *IEEE International Conference on e-Technology, e-Commerce, and e-Service.*

Gershenson, C. (2003, 08 20). Artificial Neural Networks for Beginners.

Gibson, C. (2009). Using SERVQUAL to Assess the Customer Satisfaction Level of the Oregon HIDTA ISC Analytical Unit.

Grönroos, C. (1990). Service Management and Marketing: Managing the moments of truth in service competition.

Grönross, C. (1998). Marketing services: the case of a missing product. *JOURNAL OF BUSINESS & INDUSTRIAL MARKETING, 13* (4/5).

Introduction to Core Data Programming Guide. (2012). Retrieved 03 13, 2013, from https://developer.apple.com/library/mac/#documentation/cocoa/conceptual/CoreData/cdProgrammingGuide.html

Herlocker, J. L., Kostan, J. A., Terveen, L. G., & Riedl, T. J. (2004). Evaluating Collaborative Filtering Recommender Systems. *22*.

Hill, P. (1977). On Goods and Services Review of Income and Wealth. *23* (4), 315–338.

Hill, P. (1999). Tangibles, intangibles and services: a new taxonomy for the classification of output. *Economic Commission for Europe April.*

Jaccard Index. (2013). *Jaccard Index.* Retrieved 05 20, 2013, from Wikipedia: http://en.wikipedia.org/wiki/Jaccard_index

Jianming, H. & Wesley, C. W. (2010). A Social Network-Based Recommender System (SNRS). *Data Mining for Social Network Data* , 47–74.

Jonathan, B. & Nash, L. (2003). Customer satisfaction: Applying concepts to industry-wide measures. *Cornell Hotel and Restaurant Administration Quarterly , 44* (4/5), 173–183.

Khan, A. S. & Hoffmann, A. (2003). Building a case-based diet recommendation system without a knowledge engineer. *Artifical Intelligence in Medicine* .

Lai, K. & Cerpa, N. (2001). Support vs Confidence in Association Rule Algorithms.

Leung, M. K. (2007). Naive Bayesian Classifier.

Lin, A. & Lin, A. (2010). Building a predictive model.

Lindsay, S. I. (2002). A tutorial on Principal Components Analysis.

Lovelock, C. (2004). Frontiers in Services Conference. Miami: University of Miami.

Lovelock, C. & Wirtz, J. (2004). Services Marketing.

NSDictionary Class Reference. (2013, 05 18). Retrieved from https://developer.apple.com/library/mac/#documentation/Cocoa/Reference/Foundation/Classes/NSDictionary_Class/Reference/Reference.html

Martinez, L. & Barranco, M. J. (2010). A Method for Weighting Multi-Valued Features in Content-Based Filtering.

Mitchell, T. & McGraw, H. (1997). *Machine Learning.*

Oliver, R. L. (1980). A Cognitive Model of the Antecedents and Consequences of Satisfaction Decisions. *JMR, Journal of Marketing Research*.

Pazzani, M. J. & Billsus, D. (2007). Content-Based Recommendation Systems.

Parasuraman, A., Zeithalm, V., & Berry, L. (1988). SERVQUAL: A Multiple-Item Scale for Measuring Consumer Perceptions of Service Quality. *Journal of Retailing, 64* (1), 12–40.

Pedge. (n.d.). *Sample restaraunt evaluation.* Retrieved 10 20, 2012, from http://www.pedge.com/reports/pemsplrrpt.pdf

Sarwar, B., Karypis, G., & Konstan, J. (2001). Algorithms, Item-Based Collaborative Filtering Recommendation.

Schafer, B. J., Konstan, J., & Schafer, J. R. (1999). Recommender Systems in E-Commerce.

Shu, X. (2010). *A Hierarchical Model of the Chinese Mobile Communications Market: An Empirical Analysis.*

Shostack, L. G. (1985). Planning the Service Encounter.

Rust, R. T. & Miu, C. (2006). What academic research tells us about service. *Cummunication of ACM Vol 49. No 7 2006, 49*.

Rust, R. T. & Oliver, R. L. (1994). *Service quality: New directions in theory and practice.* Sage Publication.

RathMell, J. M. (1966). What is meant by service? *Journal of Marketting*.

Reichheld, F. F. & Sasser, E. W. (1990, September). Zero Defections: Quality Comes to Services. *Harward Business Review* .

Ricci, F. (2010). *Mobile Recommender Systems.*

Ricci, F., Rokach, L., Shapira, B., & Kantar, P. B. (2011). *Recommender Systems Handbook.* Springer.

Roundtable. (2012, 10 20). *www.dining.csus.edu.* Retrieved from www.dining.csus.edu: http://www.dining.csus.edu/wp-content/uploads/2012/07/roundtable.pdf

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). *Introduction to Data Mining* (First Edition). Addison Wesley.

Tryfos, P. (1997). Cluster analysis.

Wang, L. (2000). High Dimensional Data Analysis.

worldwidewebsize. (n.d.). *www.worldwidewebsize.com*. Retrieved 08 2012, 12, from www.worldwidewebsize.com: http://www.worldwidewebsize.com/

Wörndln, W. (2009). Lecture notes on Collaborative Filtering Recommender Systems. Technical University of Munich.

Wyckham, R. G., Fitzroy, P. T., & Mandry, G. D. (1975). Marketing of Services – An Evaluation of the Theory. *European Journal of Marketing, 9* (1), 59–67.

Yao, Y. Y. (1995). Measuring Retrieval Effectiveness Based on User Preference of Documents.

Zanker, M. & Jannach, D. (2010). Introduction to Recommender Systems. *ACM Symposium on Applied Computing*.

Zeithalm, V. A., Berry, L. L., & Parasuraman, A. (1993). The nature and Determinants of Customer Expectations of Service. *Journal of Academy of Marketing Science, 21*.