MODELLING USER PROFILES FOR RECOMMENDER SYSTEMS

A THESIS SUBMITTED TO

THE SCIENCE AND ENGINEERING FACULTY

OF QUEENSLAND UNIVERSITY OF TECHNOLOGY

IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Wanvimol Nadee

Science and Engineering Faculty

Queensland University of Technology

2016

Copyright in Relation to This Thesis

Statement of Original Authorship

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

QUT Verified Signature

Signature:

Date: 26 February 2016

Abstract

Recommender systems are popular personalisation tools to assist users in finding relevant information based on the preferences maintained in their respective profiles. User profiles play an important role in the success of the recommendation process since the profiles represent the users' information needs. Therefore, every recommender system needs to develop or maintain a profile or model of user preferences in order to identify the needs of an individual user. The accuracy of each user profile affects the performance of the entire recommender system.

The explosive growth of the Internet has increased the volume and complexity of information. Users are faced with a variety and vagueness of information before they can isolate the content that fits their needs. Consequently, they are often uncertain regarding their information needs or do not know how to describe what they want. The most challenging task involved in building personalised recommendations is acquiring information on user needs and preferences when there is limited information about users. These problems make it difficult to profile users accurately with a view toward making quality recommendations.

The practice of building an item taxonomy can be used to obtain more information about each user and the relevance of each item to other users. Such information may also assist in determining users' preferences. Item taxonomy information is based on a set of categories or topics that can be used to classify and describe items in a hierarchical structure, from coarse-grained classes to fine-grained classes. However, using that taxonomy to identify each user's correct information needs is still a challenging task because of the complex relationship among concepts, items, and users.

In light of these challenges, this thesis proposes a new method for using taxonomy information to describe uncertain knowledge about a user's information needs or preferences in more detail. The proposed structure takes the form of a two-dimensional hierarchy that can represent both user profiles as a user concept hierarchy and items as an item concept hierarchy. It will significantly enhance the performance of personalised recommendations and will also offer an alternative solution to the cold-start problem. Additionally, this thesis develops a recommender algorithm based on a language model to allow the proposed structure to be implemented. The language model is applied to facilitate an understanding of how items are likely to generate what the user wants. The proposed approaches are evaluated using standard Amazon.com book and music datasets, and experimental results demonstrate that the approaches described in the proposed model outperform related baseline models.

Keywords

User profiling, user preferences, recommender systems, item taxonomy, personalisation, the cold-start problem, concept hierarchy model, language Model

Acknowledgments

I would like to express my sincere gratitude to my principal supervisor Professor Yuefeng Li, for his invaluable expertise, understanding, encouragement, trust, tolerance, full support, expert guidance, and helpful assistance throughout the journey of this research project. I am so lucky to have such a kind, modest, gentle, nice, and talented mentor. Without his incredible patience and timely wisdom and counsel, my thesis work would have been a frustrating and overwhelming pursuit.

I also express my appreciation to Associate Professor Yue Xu, Senior Lecturer Jinglan Zhang from QUT, and Associate Professor Raymond Lau from City University of Hong Kong, for their generous support and comments on my work during this candidature. Likewise, Likewise, I would like to thank the Thai Royal scholarship committee, and the employees of the Office of Education Affairs (OEA), Maejo University, and QUT. Without this scholarship and their support, I would not have had this opportunity to study at QUT in Australia. I would also like to thank all the anonymous reviewers of my papers and thesis for their valuable comments to improve my research. I wish to thank all my fellow students and friends for their friendship. I also would like to thank for my fellow Thai students and Qi Dong for supporting me in the creation of this thesis.

Finally, I would like to thank my parents and my cousins for their unconditional love and encouragement during the last four years.

Table of Contents

Nomenclature

Abbreviations

AP	Average Precision
BRP	Bayesian Personalised Ranking
CBF	Content-Based Filtering System
CF	Collaborative Filtering System
CTLM	Concept Taxonomy with Language Models recommendation Approach
D1	Book Dataset
D2	Music Dataset
F_1	F_1 measure
IF	Information Filtering
ILM	Inferential Language Model
Ip	Item popularity baseline model
IR	Information Retrieval
ITB	Item Taxonomy Recommendations baseline model
Item-Based CF	Item Based Collaborative Filtering
kNN	k-Nearest Neighbour
LM	Language Modelling
MAE	Mean Absolute Error

MAP	Mean Average Precision
MF	Matrix Factorization
MLE	Maximum likelihood estimation
MSE	Mean squared error
PopCs	The Item Popularity and
	Concept Hierarchies Recommendation Approach
RMSE	Root Mean Squared Error
SVD	Singular Value Decomposition
TF*IDF	Term-Frequency times
	Inverse Document Frequency
Top-N	A numer of item recommendations
User-Based CF	User Based Collaborative Filtering
UUCH	User-User Concept Hierarchies Recommendation Approach
VSM	Vector Space Model

Symbols		**Chapter**
B	A set of items or products	Ch3,4,5
b_k	An item	Ch3,4,5
$\overrightarrow{b_k}$	a vector of concepts for an item	Ch3,4
C	A set of concepts (or categories)	Ch3,4,5
c_h	A taxonomic concept in horizontal direction of the hierarchy	Ch3

c_i	A taxonomic concept	Ch3,4
c_v	A taxonomic concept in vertical direction of the hierarchy	Ch3
$cs(u_a, b_k)$	The user-item concept hierarchy similarities	Ch4,5
$cs(u_a, u_i)$	The user-user concept hierarchy similarities	Ch4,5
$cw_{H_{u_a}}(v, h)$	The concept hierarchy weight on level v at the order h for an active user	Ch3
$cw'_{H_{u_a}}(v, h)$	The normalised weight of the concept on level v at the order h on the hierarchy for an active user	Ch3
$cw_{H_{u_i}}(v, h)$	The concept hierarchy weight on level v at the order h for an existing user	Ch3
$cw'_{H_{u_i}}(v, h)$	The normalised weight of the concept on level v at the order h on hierarchy for an existing user	Ch3
$cw_{H_{b_k}}(v, h)$	The concept hierarchy weight on level v at the order h for an item	Ch3
$cw'_{H_{b_k}}(v, h)$	The normalised weight of the concept on level v at the order h on the hierarchy for an item	Ch3
D_{b_k}	A set of item descriptors	Ch3,4
d_x	An item taxonomic descriptor	Ch3,4
H	A concept hierarchy	Ch3
H_{u_a}	An active user concept hierarchy	Ch3,4
H_{u_i}	An existing user concept hierarchy	Ch3,4

H_{b_k}	An item concept hierarchy	Ch3,4
k	A number of similar users or items	Ch2,4
L_v	A level in a concept hierarchy	Ch3
$npop(b_k)$	The normalised item popularity	Ch4,5
$Precision(P)$	Precision	Ch4
$P(b_k)$	The probability of item	Ch4
$pop(b_k)$	The item popularity weight	Ch4,5
$P(c_i)$	The probability of concept in all relevant item hierarchies	Ch4,5
$P_{ML}(c_i\|M_D)$	The maximum likelihood estimation of the entire item collection LM	Ch4
$P_{INF}(c_i\|M_{b_k})$	The item inferential language model the entire item collection	Ch4
$P(d\|q)$	The probability of the relevance of document to a given query	Ch2,4
$P(d)$	The probability of document	Ch2,4
$P(t\|M_D)$	A language model built for the entire document collection	Ch2,4
$P(b_k\|u_a)$	The probability that the user concept of interest is generated by the items descriptors	Ch2,4
$P(t\|M_d)$	A language model built for each document collection	Ch2,4

$p\_score(u_a, b_k)$	The prediction score	Ch4,5
q	The query	Ch2,4
$Recall(R)$	Recall	Ch4
R_{ik}	The user explicit rating	Ch3,4,5
$\|Rec_{u_i}\|$	the set of Top-N recommended items for active user	Ch5
$sim(\overrightarrow{u_a}, \overrightarrow{b_k})$	the vector cosine similarity between an active user and an item	Ch4
$sim(\overrightarrow{u_a}, \overrightarrow{u_i})$	the vector cosine similarity between an active user and an existing user	Ch4
$tf(c_i, D)$	the occurrence frequency of concept in the entire item collection D	Ch4
$\|T_{u_a}\|$	The number of test items of the active user	Ch5
U	a set of users	Ch3,4
u_a	An active user or a given user or a new user	Ch3,4,5
u_i	An existing user	Ch3,4,5
$\overrightarrow{u_a}$	a vector of concepts for an active user	Ch3,4
$\overrightarrow{u_i}$	a vector of concepts for an existing user	Ch3,4
μ_{L_v}	The importance of concept in the vertical direction of the hierarchy	Ch3
α	Alpha	Ch2,3,4,5
β	beta	Ch2,3,4,5

$\prod$	Capital pi	Ch2,3,4
$\propto$	proportional to	Ch2,3,4
λ	Lambda	Ch3
$\in$	Set of element	Ch3,4,5
λ_{c_h}	The importance of concept in the horizontal direction of the hierarchy	Ch3

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 OVERVIEW AND MOTIVATIONS

In recent years, the explosive growth of the Internet has increased the volume and complexity of information. Consequently, users are faced with a variety of information before they can isolate the content that fits their needs. There are numerous options to consider, and it often takes considerable time to find the right information. The best solution to this problem is to use intelligent information systems to find and filter out unwanted information. Recommender systems are popular applications in Information Filtering (IF) and Information Retrieval (IR) systems because the suggestions they make assist users in seeking information [12]. Recommender systems provide easily accessible, high-quality recommendations to support users in their decision making, while simultaneously dealing with information overload. Recommender systems have been adopted widely by many e-commerce sites, e.g., Amazon, Netflix, Yahoo, and LinkedIn, to provide personalised services and products to customers.

Potentially, there are several ways in which recommender systems can be studied and improved upon, including developing new algorithms for making recommendations, utilising additional information sources to support user profiling and recommendation processes, improving recommendation problems such as information overload, and addressing other problems such as sparsity, cold-start, and scalability [67, 71, 111, 121].

Essentially, recommender systems can be seen as personalised information-filtering applications. The potential of personalisation lies in its ability to tailor information and present it to each individual user based on knowledge about his or her preferences and behaviour [34]. In the long run, this is likely to lead to improved customer satisfaction and higher profits [17].

The most fundamental step in creating a personalised recommendation is acquiring and learning user preferences [105]. Therefore, to provide users with the specific information that will satisfy their requirements, every recommender system needs to build a user profile or a model of user preferences to identify the needs of individual users [48]. In theory, the more accurate each user profile is, the more effective the recommendations will be. However, in the real world, interpreting various information about users or their interests involves fuzzy information or knowledge. Fuzzy knowledge can be defined as information or concepts that are vague, imprecise, uncertain, or ambiguous in nature [21]. For example, the concepts young, high, tall, good, cold and interesting are fuzzy; in other words, there is no single value that represents these concepts specifically. A further example is the concept 'fast', which is described by the fuzzy concept of speed. The problem here is that some define fast as 180 kmh whereas others might define it as 120 kmh; in other words, the concept 'fast' has unclear boundaries. This makes it difficult for users to identify the right information for their needs, and can reduce the quality of recommendations when users are confronted with uncertain information needs or do not know how to describe exactly what they want.

The challenge of making personalised recommendations is determining how to acquire user preferences or needs effectively. When there is limited information about new users this is called the cold-start problem. For example, a new Netflix user may only have rated a small number of items. With only a few ratings, it is very hard to find what the user wants [19, 102]. This problem makes it difficult to profile users accurately and to make quality recommendations. Various approaches to solve this challenging issue have been developed, such as discovering user preferences from interactions between users and items [7], maintaining a profile or a model of user preferences in order to identify the needs of an individual user [48], or using additional information about users (e.g., gender, age and geographical location) and items (e.g., genres, product categories, keywords, and product descriptions) to enhance the performance of recommender systems [32, 55, 75, 85]. However, it is very hard to find a suitable solution for the cold-start problem.

To address these recommendation-making problems, however, taxonomy information should also be exploited. Taxonomy information is another popular textual source of information and provides an alternative data source for learning user preferences [67]. In essence, an item taxonomy describes or classifies items according to sets of categories or topics; in other words, it is a set of topics that is designed by website experts or web managers based on their personal experience [46, 67]. Item taxonomies are used widely on e-commerce sites such as Amazon.com (http://www.amazon.com). Many different relationships between items are indicated by the item taxonomies, and this data is exploited to help users find their preferred products or items quickly and easily [46]. Taxonomies are commonly available for many domains, such as product categories and price attributes [4].

One of the main advantages of item taxonomy is that topic correlations within item taxonomies represent a hierarchical relationship between topics. There are also other advantages, including implicit feedback data, standard vocabularies, and the fact that a taxonomy is not vague and can be controlled by experts [67]. In addition, a taxonomy hierarchical structure can reflect user preferences from general topics, at the root nodes, to specific topics at the leaf nodes [52]. This enables user preferences in the items to be linked with taxonomic item information. In short, information on a user and his or her preferences can be learned from the item taxonomy information.

Recently, the leverage of item taxonomy information in recommender systems has been an increasingly popular focus of scholarly research. Some studies recommended the use of implicit item ratings and item taxonomy information to capture user preferences for solving the cold-start issue [4, 54, 73, 79, 111, 118, 121]. These studies provided techniques to represent items in terms of concepts or concept weight vectors. Research in this area has been driven largely by Ziegler et al. [121] and Weng et al. [111]. The researchers' approach is to alleviate the recommendation problem when user ratings are sparse and to enhance the performance of recommendations. Put simply, the recommendation method aims to exploit the relationship between users' item preferences and the taxonomic categories of the item. User ratings data can also be used to compute and capture users' preferences in a hierarchical ontology. The advantage of this approach is that it recommends items based on the importance of topics rather than explicit item ratings. However, they do not offer an effective way to represent or summarise this knowledge based on only a few rated items.

The effectiveness of recommendations can be improved in many different ways using machine learning techniques, data mining, approximation theories, and artificial intelligence. Many studies utilise a content-based filtering technique and the contents of items to solve the cold-start problem and enhance the performance of recommendations. In content-based filtering, recommendations are made by extracting the content of the items in which users have previously shown an interest. The system recommends new items that are similar to the users' preferred content patterns. However, content-based filtering approaches are based solely on item content and are not capable of profiling users [80, 110, 117]. Content-based approaches still have limitations in terms of new users, over-specialisation and restricted content analysis. Using content-based techniques and item taxonomy information to capture user preferences for items is still a challenging task in the field of recommender systems, and needs to be explored further.

The main objective of this research is to enhance approaches for making personalised recommendations. The research provides an effective way to summarise the useful knowledge based on user ratings of only a few items, which makes a significant contribution to the cold-start problem and situations when users are confronted with uncertain information needs. This study focuses on using taxonomy information as domain-specific knowledge and explores how to exploit the relationship between users and items according to the set of taxonomic categories utilised by the user, and relate them to the item. The purpose of this task is to obtain the user's needs or preferences and to profile users accurately. Furthermore, the study introduces a new way of representing user profiles and item descriptions in a meaningful way in the concept hierarchy.

1.2 RESEARCH PROBLEMS

Most research involving recommender systems and information retrieval (IR) has focussed on achieving accuracy in the retrieval of the most relevant items or documents that match the needs or personal interests of an individual user. Personal information about the user is important for personalised recommendation-making. Typically, users' ratings data is widely used to capture their preferences. Based on the cold-start problem, the recommender system cannot draw any relation between users and items if few ratings exist in the system, or if the user's personal data is difficult to obtain. This problem impacts the effectiveness of the recommendation-making. Many personalised recommender systems have to contend with having insufficient personal data to generate accurate recommendations, and the cold-start problem is one of the main limitations of recommender systems [19, 48, 76, 102, 111].

In addition to the cold-start problem, a further issue is how to help users filter fuzzy information in order to identify information that correctly matches their needs. In order to generate personalised recommendations, other user data sources and new approaches to obtain user preferences need to be explored. Many previous studies have tried to develop new strategies to make better recommendations by utilising the limited available user information, as well as other information sources regarding item content such as item taxonomy information [4, 33, 54, 67, 71, 77, 79, 111, 118, 120]. Although content-based approaches are beneficial in improving the performance of recommender systems, a more pressing question remains: how can we utilise the existing user information sources to improve the quality of recommendations and solve these recommendation-making problems?

The primary goal of this thesis is to explore how item taxonomy information can be used to identify user informational needs or interests to support user profiling. A further goal is to devise an alternative solution to the cold-start problem and to dealing with uncertain information. The thesis also explores effective approaches regarding the use of item taxonomy information to improve the performance of making recommendations.

1.3 RESEARCH OBJECTIVES

In order to improve the accuracy and effectiveness of recommendations, the primary research objectives of the thesis are as follows:

Objective 1: Utilise additional information regarding item taxonomy to support user profiling and address personalised recommendation problems.

Item taxonomy information provides content information and other user information sources that can help in learning user preferences. The idea here is to investigate how to use item taxonomy information to obtain user preferences accurately, in cases where users are faced with uncertain information and there is limited information available on new users. In essence, the goal is to develop an effective approach that can identify these users' accurately in a new structure as a concept hierarchy. Accordingly, a new user profiling approach based on item taxonomy is developed and designed to support user profiling and provide an alternative solution to the cold-start problem.

Objective 2: Explore and develop new recommender algorithms to support the creation of personalised recommendations.

The goal of a recommender system is to suggest items matching user preferences or needs. This research seeks to explore and develop an effective recommendation approach based on item taxonomy information in order to enhance the personalised item recommendations task.

Objective 3: Verify that the proposed idea of using item taxonomy information in recommender systems can improve the performance of recommendations.

To ensure that the proposed approaches based on taxonomy information can improve the performance of recommendations, the effectiveness of the proposed user profiling and recommendation-making approaches will be evaluated using experiments, based on accuracy metrics. The effectiveness of the proposed approaches will be evaluated on a Top-N recommendation task. This evaluation will be done by comparing the proposed recommender systems to other related models and state-of-the-art models. If the accuracy of the recommendation-making approaches can be improved, then the effectiveness of the proposed user profiling and recommendation approaches will be verified.

1.4 MAIN CONTRIBUTIONS

Based on the research problems discussed above, we make several contributions in this thesis towards acquiring user preferences to improve user profiling and make better personalised recommendations. The contributions of this thesis are as follows:

1. The problem that we address in this research is how to acquire user preferences (or needs) effectively when there is a cold-start problem. This includes the issue of uncertain information regarding user needs and preferences. The research provides an alternative method of using existing available information resources to formalise the recommendation-making problems and to enhance user profiling. The contributions related to this work are as follows:
    1.1. This research provides an effective way to summarise useful knowledge based on the ratings of only a few items, which makes a significant contribution to the cold-start problem. The item taxonomy information is utilised as domain-specific knowledge, and explores how to exploit the relationship between users and items according to a set of taxonomic categories (or concepts) that are utilised by users.
    1.2. This thesis proposes a novel concept hierarchy model to gauge a user's preferences about the items, according to a set of taxonomic concepts that the user interacts with and the item has. This research contributes to more accurate

user profiling which captures the personal preferences and assists in tackling the cold-start problem. This model provides a better structure for new users to approximate their needs in a two-dimensional hierarchy. It also allows users to transmit their preferences in a concept hierarchy rather than by only using ratings. It provides a new way of representing a user's personal preferences as a user concept hierarchy and an item's features as an item concept hierarchy.

1.3. The user profile and item representation based on concept hierarchy are utilised to measure the similarity of concept hierarchy preferences between a user and an item. It contributes to improvement in the performance of the neighbourhood formation process, either in normal situations or in cold-start situations.

1.4. The user-item concept hierarchy similarities model is also designed to enhance the item recommendation-making approaches.

2. Ultimately, the research makes a contribution to the development of new recommendation functions. The combination of user profiling and item representation, based on taxonomy information, is applied to the recommendation algorithms of recommender systems to further enhance the quality of personalised item recommendations and to tackle recommendation making problems. The contributions of this work are summarised as follows:

   2.1. The item popularity and user-item concept hierarchy similarities are incorporated to create a new item recommendation function, in order to improve the quality of item recommendation making and tackle the cold-start problem.

   2.2. An effective item recommendation function is proposed to estimate the probability that a user is interested in an item, based on the underlying language model (LM) and a combination of item popularity, user-item concept hierarchy similarities, and the probability of concepts. It contributes to improving the performance of recommendations in both normal situations and cold-start situations.

2.3. We evaluate our proposed approaches with two real-world standard data collections: Amazon book and music datasets. The experimental results are compared with three different baseline models.

1.5 THESIS ORGANISATION

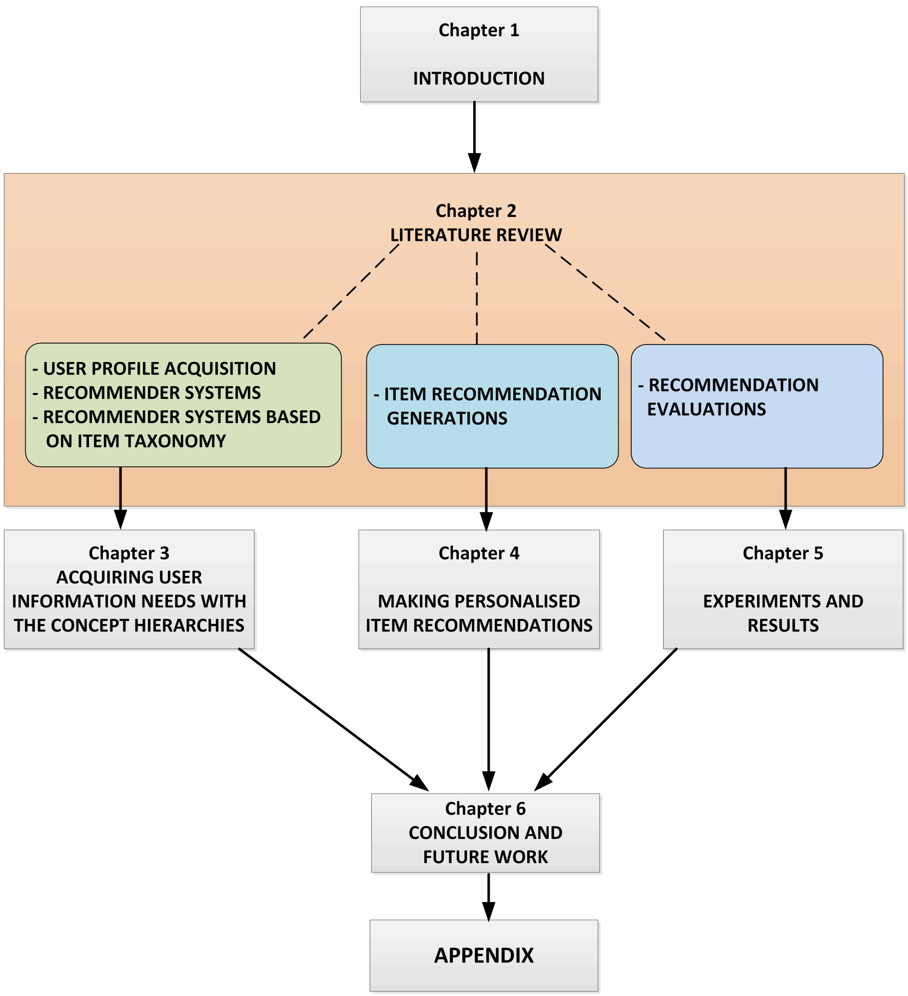

Figure 1.1: The structure of the thesis

Objective	Contribution	Chapter	Literature Review
1	1 and 2	Ch3	Section 2.1, Section 2.2
2	3.1, 3.2, and 3.3	Ch4	Section 2.2, Section 2.3, and Section 2.4
3	3.4	Ch5	Section 2.5

Table 1.1: Correlating the thesis objectives to the thesis contributions, and to the topics covered in the literature review.

To date, various research approaches have been proposed in the recommender system field, including surveys, empirical analysis, and experimentation and evaluation methods [44, 47]. This thesis focuses on developing new approaches to generating a user profile and representing items based on item taxonomy information and new recommendation approaches, and then evaluates the results in the area of recommendations. Figure 1.1 shows the structure of this thesis, and Table 1.1 shows a thematic roadmap that correlates the thesis objectives, contributions, and topics covered in the literature review. This will help the reader grasp the diverse disciplinary scope of the thesis. The remainder of this thesis is organised as follows:

- **Chapter 2:** This chapter is a comprehensive review of existing research that is relevant in the area of recommender systems. It identifies and justifies the research context and the problems from which the research questions were derived, as well as the advantages and disadvantages of existing related work.

- **Chapter 3:** This chapter discusses how the item taxonomy information can be used to obtain user information needs or preferences in a two-dimensional hierarchy. It will also describe how to represent a user and an item based on concept hierarchy.

- **Chapter 4:** This chapter discusses how to utilise the user profiles and the representation of items, based on the concept hierarchy proposed in Chapter 3, to make recommendations. The two item recommendation-making approaches will be presented in this chapter.

- **Chapter 5:** This chapter discusses the evaluation of the proposed approach using taxonomic concept hierarchies and recommendation approaches, which includes verifying the effectiveness of the proposed approaches to improve recommendation accuracy and address recommendation problems.

- **Chapter 6:** This chapter concludes the thesis and outlines the direction of future work in this area.

- **Appendices:** The appendices of this thesis include the sample category taxonomy information of the data sources, the general structure and components of the system in this thesis, and the database structure design.

The relevant publications based on this thesis are as follows:

- W. Nadee, Y. Li, and Y. Xu. Acquiring user information needs for recommender systems. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, 2013 IEEE/WIC/ACM International Joint Conferences, volume 3, pages 5-8, Nov 2013.

- L. Zhang, Y. Li, C. Sun, and Nadee, W. Rough Set Based Approach to Text Classification. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, 2013 IEEE/WIC/ACM International Joint Conferences, volume 3, pages 245-252, Nov 2013.

- W. Nadee, Y. Li, and Y. Xu. Personalised Recommendations Based on Item Taxonomy. Submitted to *ACM Transaction on the web (TWEB)*, 2015.

- W. Nadee, Y. Li, and Y. Xu. A Concept Hierarchy Model for User Modelling. Submitted to *ACM User Modelling, Adaptation and Personalisation (UMAP)*, 2016.

Chapter 2

LITERATURE REVIEW

This chapter will provide a review of the literature relevant to the field of study and the research methodology. The literature provides a thorough background understanding of the area of study and provides details to support the research in this thesis. This review covers the basic ideas and concepts, as well as various challenging problems and technologies.

Section 2.1 discusses previous studies about the acquisition of user profiles and related concepts. In this research, we provide a new hierarchy structure for the acquisition of user profiles. In the previous studies, we found no suitable way of acquiring user profiles for taxonomy data.

Section 2.2 reviews the recommendation approaches and the limitations for recommender systems. In this research, we provide new recommendation-making approaches based on hybrid approaches to improve the performance of standard recommender systems, and to help alleviate the limitations of recommender systems, such as the cold-start problem.

Section 2.3 investigates the literature regarding recommender systems based on item taxonomy. In this research, we introduce a new model for using taxonomy information to alleviate the recommendation-making problems and obtain the user's preferences for an item. In contrast to other studies covered in the literature review, the new model provides a better structure for new users to approximate their needs in a two-dimensional hierarchy.

Section 2.4 reviews source materials focused on using language models for the creation of item recommendations. In this research, a new item recommendation function is provided by considering the features of the item such as a set of taxonomic categories. The relevance of an item to a user as a kind of language model is described, to clarify how users, items, and concepts are integrated to generate what the user wants.

Section 2.5 reviews the evaluation measures for recommender systems, both in the Top-N recommendations and rating predictions. In this research, the precision and recall at N, MAP (Mean Average Precision), and $F_1 measure$ in Information Retrieval (IR) are applied to evaluate the accuracy of the proposed approaches in the Top-N recommendation tasks.

2.1 USER PROFILE ACQUISITION

User profiles play an important role in recommendation processes since their models represent the user's information needs. Most personalisation systems need to build a user profile or a model of user preferences in order to identify the needs of individual users [48]. The initial step in providing personalised recommendations is to learn about user interests and preferences in order to generate a user profile. Users' preferences can be gleaned from their past interactions with the systems in question [92]. These user interactions consist of either explicit or implicit information about the user's preferences or interest in items. The user profile allows users to be modelled, which can be described as the process of building personal preferences [10]. In other words, the user model is generally represented in the form of a user profile which captures the personal preferences of the users in terms of the user's knowledge about the object or subject in which they are interested [10, 67, 77]. User profiles can represent the interests or preferences of both an individual user and a group of users: an individual user profile provides only one user's interests and information, whereas a group user profile describes the common interests or goals of a group of users [67].

User profiling is the process of gathering information about the topics or subjects in which a user is interested [34]. The accuracy and effectiveness of user profiling affects the performance of recommender systems. The crucial aspect of user profiles is their ability to represent users' current interests. According to Gauch et al. [34], the user profiling process consists of three main phases:

- **The first phase involves information collection. This process is utilised to gather raw information about the user**

The first step to create a recommendation system is to gather the user information. To generate a user profile, the system needs relevant information about the user's preferences or interests. There are different types of user information sources and techniques that can be used to discover users' preferences or interests. Basically, the systems can gather user interests or preferences from user feedback. This feedback can be explicit or implicit, as explained further in subsection 2.1.1.

- **The second phase is profile construction and representation**

An essential of the personalised recommender systems is how to build user profile, which involves the information needs and preference of user and has great impact on the performance of recommendations. One important consideration when constructing a user profile is that more accurate the user profile is, the more effective the recommendations will be. User profiles are constructed using different techniques based on the user profile representations. User profiling is either knowledge-based or behaviour-based [77]. Knowledge-based approaches emphasise explicit domain knowledge about items and implicit knowledge about the users. The approaches are rule-based in proposing items which exactly match rules to users, for example using decision rules to classify users' personal interests or preferences based on their demographic characteristics [5].

Most recommender systems use behaviour-based approaches to construct the user profile. Behaviour-based approaches use the users' behaviour as a model and discover the useful patterns of that behaviour by using machine learning [77]. User behaviour can be represented via several types of patterns, such as frequency patterns, sequential patterns, neural network models, and graph models [67]. The profiles can likewise be generated from either implicit user data (e.g., sets of keywords, web usage data, content and structural information about visited web pages, user ratings data, and demographic information) or explicit user data (e.g., questionnaires or interviews with the user).

Based on the different user information obtained, user profiles can be represented as sets of weighted keywords, topics, concepts, or ratings. The most common representation for user profiles is a set of keywords/terms. Each keyword represents a topic of interest to the user. These keywords can be extracted from the content of products or provided directly by the user. The degree of user interest in the keywords can be weighted using the *tf-idf* method (term frequency-inverse document frequency) in the vector-space approach [34, 69].

Other approaches taken to represent user profiles include the history-based model (which uses purchase histories and user ratings), the vector space model (which uses feature vectors to represent items), weighted n-grams (which represents items as a net of words, with weights in the nodes and edges), or semantic networks (where the profiles may be represented by a weighted semantic network in which each node represents a concept). Further options are weighted associative networks (which use the associations between a set of concept nodes to represent each user profile), user-item ratings matrices (which employ user ratings for items to represent the interest of the user in each item or the user's item preferences) and demographic features (which create user profiles through user characteristics) [80].

In most recommender systems, user ratings for items are widely utilised to indicate their item preferences; this is called a rating-based user profile [67, 80]. However, the difficulty of obtaining users' rating data affects the performance of the traditional collaborative recommender systems when making recommendations or predictions. Hence, the content features or descriptions of items include keywords, phrases, category names, or other textual content embedded as meta-information in the content of web pages. They provide additional knowledge about users, which can be used to represent user profiles based on content-based analysis. Zhang and Koren [119] proposed the new idea of using analytical based on the Bayesian hierarchical linear models in order to enhance content-based user profiles. The benefit of using item features is that it can solve the cold-start problem. The preferences of the users can be obtained even when there is no rating data in the systems.

Moreover, user profiles can be represented via a concept-based, which is similar to the keyword-based method, except the data is presented as vectors of weighted features. Initially, the concept hierarchy was utilised to represent the content of web pages. More recently, some studies have been used concept hierarchy to represent user profiles that reflect the content of a given user's interest in the hierarchical structure [83, 100, 115]. Singh et al. [100] introduced a system used for news filtering in which the user's interests are modelled by a user interest hierarchy based on explicit user feedback. Generally, the basis of the concept-hierarchical profile is constructed from a reference taxonomy or thesaurus.

Many research studies refer to the concept-hierarchy as an ontology which has an 'is-a' relationship with the concepts. The hierarchical relationship one concept has with another is described as either a super-concept or a sub-concept; every member of the sub-concept is a member of the super-concept. A set of concepts and relationships is extracted and weighted to form the user profile. A set of concepts associated with a user is called a user profile, whereas a set of concepts associated with an item is referred to as an item profile [48]. The users' preferences or interests can be viewed as concept vectors in this method. The use of concept hierarchy to represent users' interests is explained further by Kim and Chan (2008) [52]. Nanas et al. [83] introduced a methodology for constructing a concept hierarchy under documents' topic to represent the users' topic interests. Based on the relationship between topics or concepts in a hierarchical structure, many studies also reviewed the new approaches to construct user profile, such as using the hierarchical relationships between topics in taxonomy to represent user's taxonomic topic interests or preferences [67, 79, 111, 118, 121]. Taxonomy-based user profiles have been used in some recommender systems, which will be discussed further in Section 2.3.

Nowadays, the development of Web 2.0 and Semantic Web technology provide bountiful textual content information, multimedia content information, and network information including tags, review, comments, posts, pictures, tweets, videos, audio clips, and social networking-which provide a valuable resource when constructing and representing user profiles [67, 104]. Li and Chang [65] introduced technique of information fusion to construct user profile. Yu et al. [115] constructed user profile based on concept and relation to represent user's real-time preference for Web personalised services.

In recent years, tag or folksonomy information has been popular textual content information in Web 2.0, and has become an important research focus. Based on tag information, user profiles can be represented by sets of tags associated with users and items. The research on tag information mainly focuses on the semantics of words to improve the quality of the traditional collaborative filtering recommendation, as well as to alleviate obstacles such as the cold-start problem [70]. The main idea of using tags or folksonomy is to profile users' topic interests or preferences when the amount of available user ratings is too small [31, 67]. With these new web developments, we are able to exploit additional information to construct and represent user profiles.

- **The third phase involves exploiting information in a user profile to provide personalised services**

After a user profile is constructed, it is then used to provide personalised services in different areas, such as personalised recommender systems, personalised searches, queries, and trust-aware recommender systems. There are three main methods utilised in recommender agents for user profile exploitation: content-based, collaborative, and hybrid methods, all of which will be further reviewed in Section 2.2. In addition, sets of topics, keywords, or concepts are utilised to provide personalised searches in this phase.

2.1.1 User Information Collection

The first phase of user profiling is to collect information about the user. To be able to identify the needs of users, the recommender system needs to know something about the user. Therefore, acquiring user information about needs or preferences is a fundamental task for making personalised recommendations. User preferences are learned from users' interactions with items. These interactions consist of explicit and implicit information, usually referred to as explicit feedback and implicit feedback [49]. The following subsection will discuss explicit and implicit feedback in detail.

2.1.1.1 Explicit user information collection

Most explicit user information collection approaches rely on users inputting personal information. This information is acquired directly via registration forms or questionnaires, or by asking users to rate items, or by tracking users' queries [49]. In other words, users are required to supply some information themselves. Many sites collect user preferences by providing personalised services to users and then directly asking them to give personal information to create a profile. This explicit user feedback reflects the actual user needs. For example, eBay asks users to provide their opinions and to give ratings for the services and products on offer. The company then utilises this information to improve the personalised recommendations that it gives to customers.

Explicit information includes demographic information (e.g., gender, educational background, age, location, and occupation), data about interests and preferences (e.g., topics of interest, tastes, preferred products and brand preferences), and opinion-based information (e.g., reviews, comments, and feedback) [34, 67, 80]. In recommender systems, explicit ratings data is widely used to profile users' preferences [7, 29, 49]. Some websites operate by using explicit ratings data, such as Netflix, which utilises movie ratings to generate popular movie suggestions for customers [89].

Although explicit feedback is effective and easy to collect, and has less noise, there are some drawbacks. First, users have to invest time and effort in expressing their preferences or interests through actions [67, 79]. In other words, explicit feedback places an additional burden on the user. If users do not provide personal information, user profiles cannot be built. Another problem relates to privacy concerns; users may not be willing to share their personal information or to give accurate information to the system. These problems affect the performance of recommender systems and make it difficult to profile users precisely. Therefore, encouraging users to provide sufficient explicit information is a challenging task.

2.1.1.2 Implicit user information collection

Explicit information is not always available and does not always contain enough detail for an adequate user profile to be built. For this reason, implicit user information is often collected. Most implicit user information is based on user behaviour. The implicit user information or implicit user feedback can be collected through web usage logs, click streams, browsing histories, purchase records, and content or structural information from visited web pages [27]. Browsing histories are a common source of implicit information [34]. Yu et al. [115] extracted user's browsing contents of each web page in a session to compute user's real-time preference. The main benefit of implicit information collection is that it does not require any extra effort on the part of users during the process of constructing profiles [48, 79]. It also allows easy and continuous access to data; it is automatically updated when users interact with the system in question.

However, it is very difficult to convert user behaviour into user preferences, as the accuracy depends on whether the user behaviour is interpreted correctly. For example, users might buy items such as music for someone else. In other words, they might not personally like all the music that they have bought. Another drawback of this information collection method is that it requires the development of high-quality applications or plug-ins, which web developers have to install.

Implicit user information is a rich source of data that allows personalised recommendations to be made. For example, Mnith [79] exploited implicit feedback such as item taxonomy and implicit rating in order to improve the predictive performance of recommendations. Ziegler et al. [121] and Weng et al. [111] proposed the use of item taxonomy in representing users' topic interests, and in making item recommendations. Kim and Chan [52] suggested that keywords or topics should be captured from users' browsing histories in order to assess their interest (using the concept-hierarchical method).

Moreover, with the development of Web 2.0, some new kinds of user information can be used as implicit user information, such as tags, comments, images, videos, posts, and click-streams. This data provides rich information about the relationship among users, items, and content and they also imply user interests or preferences. To take the tagging of a user as an example, the keywords on tags can be used to capture the user's topic interests. Amazon.com uses the usage logs of users to recommend books to their customers [48].

The data sources discussed above are very important, but there remain some obstacles related to data gathering. Privacy concerns may cause some users to withhold information or behave differently when logged in to the system. However, the advantage of user profiling lies in the access to both implicit and explicit user preference information. There are several features of user information that can be collected and utilised to retrieve items that are of interest to users. However, determining new users' preferences is challenging because limited information is available, and even that may be inaccurate, as discussed above. This is an important topic of research in relation to recommender systems and personalisation systems.

In Section 2.1, we reviewed the process of constructing user profiles. This thesis aims to enhance the traditional approach to making personalised recommendations. The thesis focuses on using taxonomy information to construct more accurate user profiles, resulting in higher accuracy of recommendations and improving the cold-start problem in recommender systems. However, the challenging task in this area is the construction and representation of a profile that accurately reflects the user's preferences. In contrast to the aforementioned techniques, this research provides a novel method, using the category taxonomy information of the item to obtain the users' preferences. We learn the user's preferences level by level in a two-dimensional hierarchy, via a set of taxonomic concepts, to generate a more comprehensive concept hierarchy preference profile for that user. The new structure provides a better way for the users to describe their needs. In Chapter 3, the details of how to acquire user information needs based on a concept hierarchy model will be discussed.

2.2 RECOMMENDER SYSTEMS

Recommender systems have become a popular topic of research, as the quantity and range of information available on the Internet is enormous. This excess of information means that it can be difficult for users to make decisions. Recommender systems are popular applications within information filtering (IF) and information retrieval (IR) systems because the suggestions that they make assist users with information seeking, by making personalised recommendations in relation to information, products, and services. Recommender systems try to predict what items would be interesting to users and to match their needs. The systems make recommendations and predictions based on information about users, such as user profiles, user preferences, user modelling, user interaction history, item features, or other users.

Nowadays, recommendations are based not only on user ratings, but also on knowledge about users and products. This knowledge-based approach to generating recommendations involves reasoning about which products or topics will meet users' requirements. According to Chen et al. [19], recommender systems are composed of three main elements: (1) a user acquisition module, which is used to collect user information; (2) an analysis module, which analyses user preferences; and (3) an algorithm module, which generates recommendations. There are two tasks involved in recommendations: rating predictions and Top-N recommendations.

The next section will introduce the basic recommender algorithms widely and successfully used in recommender systems. During the discussion, the problems with these recommender systems will also be outlined.

2.2.1 Recommendation Approaches

Recommender algorithms can be divided broadly into three categories: collaborative filtering approaches, content-based filtering approaches and hybrid approaches. These categories are examined in more detail below.

2.2.1.1 Collaborative Filtering Approaches

Collaborative filtering (CF) approaches have been used in various applications for recommender systems [74, 113]. Popular sites that use this approach include LinkedIn, Facebook, Twitter, Google, Netflix and Amazon. They incorporate collaborative filtering with recommendation engines to recommend jobs, friends, groups and/or companies in which users might be interested [95]. The main tasks in CF are rating predictions and Top-N recommendations [29, 44]. CF approaches use past information about the opinions, or behaviour of existing users in the community, to predict topics or products that a current user might like.

Basically, CF techniques make recommendations or predictions about the interests of a user based on similarities between the preferences of users in a system. The underlying idea of CF is that users who have rated the same items are more likely to have similar tastes. The techniques utilise user ratings against other users to determine the relationship between user and item, and convert the preference of a user for items to a user-item ratings matrix [13, 43, 54, 93, 102]. The input information used in CF can be either explicit ratings or implicit ratings. The most successful recommendation techniques in CF are the neighbourhood method and the latent factor model [50, 57]. CF approaches can be classified according to their algorithm techniques into two classes: memory-based CF (or neighbourhood-based CF) and model-based CF [39].

1) Memory-based CF algorithms

These algorithms utilise the entire collection of items previously rated by a user to make recommendations [3, 67]. Memory-based collaborative filtering algorithms are commonly referred to as neighbourhood-based algorithms [76]. They can be divided further into user-based CF algorithms and item-based CF algorithms [29, 94]. User-based CF makes recommendation based on the similarities between an active user and other users, while item-based CF makes recommendations based on the similarities between a target item and other items [94].

In memory-based algorithms, a user's preferences for an item are evaluated based on the ratings data of other users who have similar behaviour to the user. These k-Nearest-Neighbour (kNN) techniques are widely used in CF based algorithms to identify a group neighbourhood of users and items that are similar to a user or an item. The algorithms use the given rating data by similar users for many items to predict missing ratings or create a Top-N recommendation list for the active user. To form a neighbourhood for the active user, a similarity measure is required. The top-k neighbour users and items for the active user can be selected by calculating the similarity between the active user and all other users or all other items. The similarity measure can be calculated by various kinds of proximity computing approaches. The most common methods utilised for determining the similarities between users or items are the Pearson correlation and vector cosine similarity measures. There are also several other similarity measures used in the literature, including adjusted cosine similarity, Euclidean distance and the Jaccard coefficient [19, 44, 102].

The rating data plays an important role in CF techniques to form the neighbourhood. Bell and Koren [7] proposed the neighbourhood-based approach to improve the accuracy of kNN approaches without meaningfully affecting running time. However, when the amount of user direct rating data in the system is too small, and would resulting in poor neighbour formation and recommendations. Besides using explicit ratings and implicit ratings, the similarities between users, items, or user-items can be measured based on other features, such as users' topic interests or users' tagging behaviour [67, 81, 111, 118, 121]. Weng et al. [111] utilised the taxonomy information of the item incorporated with the existing user's explicit rating in the neighbourhood formation, instead of using only the rating data.

2) Model-based CF algorithms

These algorithms use a collection of ratings to learn the pattern of ratings, and then make intelligent rating predictions based on the learned models. The models are developed using data mining techniques and machine learning algorithms to explain the rating pattern. In contrast to memory-based CF algorithms, model-based approaches are not subject to heuristic prediction rules. There are many model-based CF algorithms, including Bayesian network-based models, clustering models, linear regression models, latent factor models, linear regression, singular value decomposition models (SVD) and matrix factorization (MF) models. A key advantage of the model-based approach is that it improves prediction performance.

Recently, the use of matrix factorization and latent factor models has become popular in recommender algorithms, both for implicit and explicit feedback. Typically, matrix factorization classifies both items and users via factor vectors that are gleaned from item rating patterns. Many studies have utilised matrix factorization and neighbourhood methods to improve the performance of Collaborative Filtering and alleviate or solve the cold-start problem [23, 55, 103].

Most memory-based CF approaches can be used for both rating predictions and Top-N recommendation tasks, whereas model-based CF approaches focus on rating predictions. Standard CF algorithms, including user-based and item-based examples, are popular benchmark baseline models. These includes latent factor and matrix factorization models, which have emerged as state-of-the-art methodologies in recommender techniques [76]. The advantages of collaborative filtering are that it is easy to implement and incorporate with other information sources, but the next section explores its drawbacks.

- Challenges and limitations of collaborative filtering

CF poses several challenges, such as scalability, synonymy, the gray sheep problem, shilling attacks, privacy protection, diversity and the long tail, sparsity, and the cold-start problem [3, 54, 56, 102]. These issues can reduce the quality of recommendations and thus of the CF approaches overall. In order to produce high-quality predictions or recommendations, CF algorithms need the ability to deal with these challenges.

The sparsity problem occurs when the number of products is extremely large [37, 38, 85]. Users tend only to have provided ratings for a small percentage of the items in a dataset, leading to a small number of ratings per item. As a result, the user-item matrix used for CF will be highly sparse; this could reduce the effectiveness of the CF systems when making predictions or recommendations. Hence, CF algorithms must have the ability to deal with highly sparse data.

The cold-start problem is the main difficulty for recommender systems. The problem occurs when a new user or new item first enters the system, and the recommender system cannot draw any relation between users and items because of the absence of information about the user or item present (e.g., insufficient reviews or ratings) [19, 38, 76, 102]. Recommender systems need to gain some details about users and items before reliable and accurate recommendations can be provided via recommendation algorithms [85]. This problem can largely degrade the performance of the traditional recommender systems in making personalised predictions, particularly when forming a neighbourhood for the active user which is based on users' ratings data. The underlying causes of the cold-start problem include [19, 28, 102, 111]:

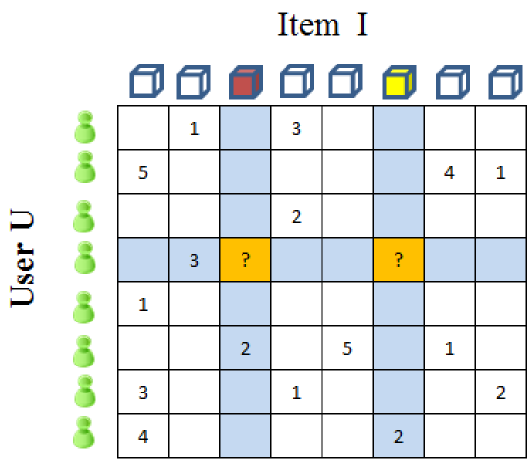

Figure 2.1: The cold-start problem.

- **The fraction of explicit ratings data in the system is small**. Figure 2.1 shows a situation where the fraction of the users' explicit ratings data is sparse. This occurs when there are only a few ratings in the system. Consequently, the system may not be able to identify users similar to the target users, thus affecting the quality of recommendations, particularly in collaborative filtering approaches which depend on ratings data to make predictions.
- **The target user provides very few ratings of the items**. When a new user has just entered the system, this user will have only rated a small subset of the overall items in the database. The recommender system cannot produce accurate recommendations for new users. Similarly, when a new item is added to the system, the system cannot recommend the new item to any user because very few users have rated or purchased this item. In this case, the quality of recommendations

to the target user may be poor. Therefore, the active users have to rate a sufficient number of items before the recommendation algorithms are able to provide accurate recommendations.

The challenge with the cold-start problem is to find a way to deal with new users and new items when purchase information is not yet available in the system. Other areas for CF improvement include looking for novel techniques for making predictions, other than finding similar users, and learning how to exploit a small amount of ratings data in order to make viable recommendations. Hang et al.[40] proposed the method to solve the cold-start problem based on the implicit information of the new users and multi-attribute rating matrix. This thesis focuses on using the features of items in terms of their taxonomy information to formalise the problem. Items can be classified by a set of taxonomic categories, and this can be used to represent users' preferences.

Scalability is another challenge for collaborative filtering-based recommender systems. This problem occurs when the existing algorithms of the recommender systems suffer due to a large number of users and items that continues to grow [8, 16, 56, 102]. The algorithms cannot respond immediately to online requirements or make recommendations for all users. Chen et al. [18] proposed an algorithm based on the probability model to tackle the scalability problem. The main purpose of this research to improve the effectiveness of recommendations and solve the cold-start problem. However, this literature review will provide some brief background to this problem. Basically, the computation efficiency of collaborative filtering is between $\mathcal{O}(m + n)$ and $\mathcal{O}(m^2 n)$, where m is the number of users and n is the number of items [111]. The analysis of the efficiency of the proposed CTLM recommender algorithm using the Big-O notation concept will be discussed in Chapter 5.

Another problems, *Synonymy*, relates to ambiguous item names or keywords or items with different names, resulting in the recommender systems being unable to discover the accurate similarities. For example, *love movie* and *love film* are actually the same item [102].

2.2.1.2 Content-Based Filtering Approaches

Content-based filtering approaches (CBF) utilise user profiles and the contents of items as domain knowledge, and compare information from the new items with the user's profile. Those items which are similar to the user's profile are recommended. [5, 88, 106]. Content-based approaches build a model or profile of user interest based on the description of items that the user has previously preferred or rated [82]. The users' interests or preferences can be described in terms of the interest in item characteristics such as topics, attributes, or categories.

Therefore, a user profile may consist of the user's preferences, needs, and implicit or explicit interests (such as sets of items, topics, concepts, or ratings) [69]. For example, the users' preferences can be represented by the users' topics preferences, which can be generated from the content-related information on items that the user rated, clicked, browsed, or bought. CBF recommendations are made by utilising these profiles to find other items with similar content to the items preferred by users. Along with as using explicit and implicit ratings to reflect user's preferences, the affinity between a user and an item can be established by determining the content of the items.

An item is represented by a set of features used to describe its content. These features correspond to a set of keywords, topics, terms, or concepts. Most content-based approaches are developed using information retrieval techniques [69]. These techniques try to match query words and/or other user data with item features [96]. There are two popular approaches to making item representations: heuristics-based and model-based (e.g., the Vector Space Model, VSM) approaches. In the heuristics-based approach, the profile of an item and user is represented as a vector of weights for each feature. The importance of words associated with an item can be determined using a weight function. The most widely-used weighting functions is *tf-idf* [28, 69, 88]. Cantador et al. adapted the VSM and ta to make use of us and item profile based on content-based

A user's interests or preferences for an item can be calculated using cosine similarity. In the model-based approach, user preferences are learned through probabilistic methods and naïve Bayes, language models, machine learning, decision trees, and/or linear classifiers [28, 88, 119]. The advantage of content-based based recommender systems is that they do not require users' ratings data. Instead of representing users' preferences with rating data, user preferences can be represented using the content of the items. However, there are also some limitations and challenges to content-based filtering recommender systems, as discussed below.

- Challenges and limitation of content-based filtering

The limited content analysis occurs when content-based techniques have a limit in the number of features that are associated, resulting in unsuitable suggestions [88]. Although content-based filtering techniques do not require rating data to learn and model user's interest, no content-based recommender systems can provide accurate recommendations without enough available information to analyse and classify the item. For example, there is not sufficient keywords to model the user interests in books.

Moreover, the *over-specialisation problem* occurs when a user is only recommended items that are similar to items that were rated or bought before. For example, suppose that at one stage in a customer's life she is interested in the everyday life of people living in the medieval age. She buys a book on this topic and then does not use Amazon.com again for a year. After coming back to the system she sees a large number of books on medieval life, but little else in the recommended products section. Not only is this person being poorly understood by the system, but her interest in this topic passed long ago. Indeed, it was satisfied by that one book, and she does not need any more. The over-specialisation problem has occurred and no useful recommendations are being made to the customer. Abbassi et al. [1] examined the over-specialisation problem based on item regions.

This is because content-based techniques have no inherent method for generating or finding items different from items users have seen before; they can only recommend items that score highly against a user's profile [3, 53, 80, 102]. In other words, they have no inherent method for generating serendipitous suggestions. To give another example, a user whose profile contains no experience with Thai food would never receive a recommendation for a Thai restaurant, even if it was their favourite kind of food. One way to solve the over-specialisation problem is to cause diverse and serendipitous items to appear on the recommendation lists [69]. A serendipitous recommendation helps the user to discover unexpected yet interesting items with a high degree of novelty that the user might not have gone looking for independently [47, 107]. Ziegler et al. [122] proposed a taxonomy-based technique to generate diverse item recommendations based on the diversity of category taxonomy.

Another issue is the *new user problem or cold-start problem*, as also seen with collaborative filtering methods. Content-based techniques are less affected than the collaborative filtering techniques by the cold-start problem [111]. Information about the user and the item, such as item taxonomy information, folksonomy information, are needed in order for recommendations to be made [3, 69]. In the context of content-based recommendations, further examination is needed of the techniques that can be used to extract item descriptions and thus to recommend items that match a user's interests. Another important question concerns how the known characteristics of items can be harnessed in order to make useful recommendations. Based on the structural information about taxonomy in a hierarchy, it can be used to find other items with similar content to the new item.

2.2.1.3 Hybrid Approaches

A hybrid recommendation system is composed of two or more diverse recommendation techniques, including collaborative filtering, content-based filtering, knowledge-based techniques, and demographic techniques [14, 48, 76, 87]. The main goal of this approach is to improve performance in terms of recommendations and to overcome some of the issues that plague recommender systems, such as the cold-start and sparsity problems. Many hybrid techniques are based on traditional collaborative filtering merged with content-based filtering [15, 112]. For example, incorporating collaborative filtering into a content-based approach to overcome the cold-start problem [24, 35, 40]. Woerndl et al. [112] applied a hybrid recommender system which integrated for mobile applications. Barragáns-Martínez et al. proposed a hybrid approach using content-based and item-based collaborative filtering methods, using singular value decomposition in order to recommend TV programs [6]. In other hybrid systems, latent factor models and item taxonomy information have been combined to facilitate the development of more personalised recommendations, thus solving the sparsity and cold-start problems [4, 50, 79, 118].

However, the cold-start and sparsity problems still remain a challenge where recommender systems are concerned. Several methods have been proposed for dealing with these issues, including dimensionality reduction of the user-item matrix (using singular value decomposition, the latent factor model, and/or matrix factorization) and the application of associative retrieval techniques [19]. Content-boosted CF approaches can also be used to gain additional information about items, with a view to computing meaningful similarities between them [32, 75]. Still, there are a few weaknesses of hybrid approaches. First, sometimes there is insufficient contextual information to model users and items well enough to make predictions. Secondly, the scalability problem still exists, since the number of users and objects grows rapidly and the systems need extra time for computation [5]. Therefore, the performance of hybrid approaches does not rely on only integrating the different techniques together. Instead, good performance and a high quality of recommendations are based on the use of appropriate knowledge from the context in the recommendation process, i.e. the content and the user behaviour.

The challenge in hybrid recommender systems is how to incorporate both content-based and collaborative filtering techniques together, resulting in high-quality recommendations. This thesis proposes two approaches based on hybrid systems in order to solve the cold-start problem, aiming to improve the effectiveness of recommender systems. We formalise these problems by using the relationship between users and items according to a set of taxonomic categories (or concepts), rather than by using item ratings. The details of populating the user profiles and different methods for achieving item-taxonomy-based recommender systems will be discussed in Chapter 3 and 4.

2.3 RECOMMENDER SYSTEMS BASED ON ITEM TAXONOMY

As mentioned earlier, acquiring user preferences is an important task in personalised recommender systems and web personalisation [105]. Recommender systems can be constructed using different types of input data. Typically, a recommender system can infer user preferences using explicit feedback, such as user ratings, or implicit feedback, such as browsing history or search patterns [57, 79]. User rating data is the most popular as it directly relates to users' personal preferences. However, since user rating data is sometimes difficult to obtain, it may reduce the performance of the recommender system, especially when new users face the cold-start problem discussed above.

According to Gantner et al. [32], the cold-start problem occurs when there are no purchases, ratings, or clicks, or when there is little collaborative information available for the system to compute predictions for the given users and items. In such a situation, additional user information (e.g., gender, age, and location) and item features (e.g., genre, product categories, and keywords) is utilised to enhance the performance of the recommender systems.

Item taxonomy information is another direct way of capturing a user's interest in an item and can also be used to solve the cold-start problem. Many sites (including Google, Yahoo! and Amazon) exploit taxonomy information to organise items, thus allowing users to find the items that they desire easily. The items are classified by a set of categories or topics within the item taxonomy. Taxonomies are used by websites all over the world and include everything from price attributes to product categories. They are usually provided by web managers or website experts based on their personal experiences.

Correlations between categories within an item taxonomy can be organised into a tree or a hierarchical structure, representing sub-concepts and super-concepts and the relations between them [48]. The hierarchical structure of the taxonomies can be used to describe or classify the items from coarse-grained through to fine-grained classes. The root category or the top category of the hierarchy is the most general of categories, and the categories become more specific as one moves toward the leaves [20].

The taxonomic tree structure can also reflect users' interests in topics, from general topics of interest at the root node to specific topics of interest at the leaf node [52, 64]. There are also other advantages, including implicit feedback data, standard vocabularies, and the fact that the structure is not vague and can be controlled by experts [67]. This enables users' interests in items to be linked with taxonomic information. In short, information about a user and his or her preferences can be learned from item taxonomy information. A user who has searched for a children's book about Iron Man might find value in having a superhero toy recommended to them. *Superhero* is higher up in the taxonomy tree than the more specific *Iron Man* topic, and toys may be an implicit interest inferred by the interest in children's books.

Recently, the usage of item taxonomy in recommender systems has been emphasised by many researchers. For example, Koenigstein et al. [54] proposed a novel usage of music taxonomy and matrix factorization to tackle the item sparsity problem [54]. Mnih [79] used taxonomy information to generate features that relate music to users' ratings histories. Ahmed et al. [4] have employed a taxonomy of item attributes together with the latent factor method to determine the factors related to a user's interest in an item. Zhang et al. [118] have also elaborated a system for discovering a user taxonomy for online shopping data.

Other studies of note here include those by Ziegler et al. [121] and Weng et al. [111], who proposed the use of taxonomy-driven product recommendations to solve the cold-start problem and enhance recommendations. This approach combined the content-based filtering and memory-based collaborative filtering approaches linearly in order to make recommendations. The aim was to reduce the problems associated with the collaborative filtering method when user ratings are sparse. Weng et al. [111] integrated items' taxonomic descriptors with implicit and explicit ratings, utilising tree-structured product taxonomy to find users' topic interests and the relevant topics related to items.

Ziegler et al. [121] decayed the weight of the taxonomic topic node on the item taxonomy tree, based on its number of leaf nodes and its length, in order to generate topic preferences and thus make item recommendations. As a result, users' topic interest profile vectors were generated using the scores for each topic. These vectors were then exploited to identify similarities between users, which were identified using the Pearson correlation method. Mostos-Junior et al. [73] used category taxonomy from the content of the web page with content-based techniques to recommend a set of books to the user. Hong et al. [45] studied incorporating item taxonomy into collaborative filtering to quantify users' exact levels of interest, thus facilitating the most effective personalised recommendations.

Most current studies map the taxonomic interests of target users against those of other users, or against taxonomic information about items [111]. The methodology used in such studies is likely to involve content-based filtering techniques. However, there are still many promising ways in which to utilise item content, item taxonomy and content-based techniques in recommender systems. The advantage of the above approaches is that they use topics or concepts rather than item ratings. How to effectively represent or summarise the knowledge in a few rated items remains a challenging problem.

This thesis takes the taxonomy-based approaches to the development of effective recommender systems, as it uses a concept hierarchy model with two-dimensional hierarchy to obtain users preferences. This has the advantage of summarising user preferences given information on their responses to only a few items. The thesis explores how to design an effective recommendation algorithm based on the proposed concept hierarchy and other user rating data if it is available. The details of constructing the proposed taxonomy-based concept hierarchy model will be discussed in Chapter 3.

2.4 ITEM RECOMMENDATION GENERATIONS

Creating a Top-N list can be seen as the second step in recommender systems after rating predictions [29, 39, 51]. Item recommendations involve generating a personalised ranking for a set of items. Usually, each item is given a score that reflects the user's predicted preference for that item. The items are then ranked according to these scores [92]. Several novel item recommendation techniques that utilise both explicit and implicit feedback have been proposed and developed by researchers.

Traditionally, the most popular Top-N recommendation method in recommender systems is k-Nearest-Neighbour (kNN) collaborative filtering [20, 42]. Here, a user's ratings are compared with those of other users who have similar interests, and then a ranked list of items that are likely to be relevant to the user's needs is produced. In recent years, the use of matrix factorization (MF) has become popular in relation to both implicit and explicit feedback. The basic idea of MF is to find the unknown ratings associated with users and items in the matrix, and then to sort the ratings to select the Top-N items. The models map both users and items to a latent factor space of dimensionality. This technique characterises both items and users by vectors of the factors inferred from item rating patterns [55].

Another useful method is Bayesian Personalised Ranking (BPR), which uses implicit positive-only feedback and tag prediction [92]. Although ranked list recommendation is not the focus of this research, it is surveyed and discussed. BPR is a personalised-ranking approach that optimises approximations of the area under the ROC curve for each user. It utilises an optimisation criterion and a gradient-based learning algorithm to generate personalised item recommendations. The goal is to create a ranking function for each user, through which more relevant items can be ranked higher than non-relevant items. This model is widely used as the baseline approach, which is involved in the personalised ranking and the ranked list recommendations [4, 50, 118].

Recently, Information Retrieval (IR) techniques have been exploited in several studies in the field of recommender systems in order to enhance item recommendation tasks. The concept of relevance has been widely used in IR methods to describe relevance ranking problems in item recommendation tasks. The relevance-based language model was proposed by Lavrenko and Croft [61]. It is a model-based query expansion approach in the language modelling framework. A relevance model involves a distribution of words in the relevant class for a query [66]. Based on the original relevance model approach, some researchers have sought to adapt the principles of ranking for relevance by utilising the vector-space IR model for item ranking in collaborative filtering [9], statistical language models (with the probability item ranking for collaborative filtering) [108], $tf - idf$ (with the user-item relevance model) [109], and the relevance-based language model (with a probabilistic clustering technique to perform neighbour selection) [86].

This thesis proposes a novel item recommendation approach which is designed under the principle of language modelling (LM). LM is a general, formal approach that is used in IR for text retrieval. Basic LM approaches have been developed from the traditional probabilistic approach in IR, in which two types of user information needs or query representations and document representations are utilised to indicate how well documents satisfy users' informational needs [72, 90]. The relationship between the query language and the document language is used to build query generation by estimating the probability of generating queries according to the document model.

In the language model, the model provides different approaches to document ranking. The basic and most common approach for using language models in IR is the *query likelihood model* [72, 90]. Documents are ranked according to the probability of the document in a language model M_d and the model generates a set of words t that are relevant to the query q. The intuition of the LM is that the user has a document in mind, and generates a query based on words that appear in that document. The model can be used to work out the probability $P(d|q)$ of the relevance of document d to a given query q using the following equation:

$$P(d|q) \propto P(d) \prod_{t \in q} ((1 - \lambda)P(t|M_D) + \lambda P(t|M_d)), \tag{2.1}$$

where M_d is a language model built for each document d, and M_D is the language model built for the entire document collection.

This equation combines the probability of the document with the general collection frequency of words t, which is referred to as a *linear interpolation language model* or *smooth probabilities*. The term frequency of a word in the document is estimated by using *maximum likelihood estimation (MLE)* and *the unigram language model*. The goal is to rank documents by $P(d|q)$, where the prior probability of a document $P(d)$ is interpreted as the likelihood that it is relevant to the query q. Typically, the prior probability of a document $P(d)$ is often treated as uniform across all documents. A genuine prior probability can also be implemented from other criteria, such as category, length, genre, newness, and number of previous people who have read the document. In order to make item recommendations, Equation 2.1 can be modified to generate the probability $P(b|u)$, that a concept hierarchy in item b is relevant to a given user u. Top-N items can be returned according to the likelihood of $P(b|u)$ to the user. Item recommendations can therefore be ranked based on an adaptation of the concept of relevance in LM.

All the work on item predictions discussed above is evaluated via personalised rankings. However, further exploration needs to be undertaken on how the effectiveness of item recommendations (which are based on the language model) can be improved. Specifically, in this thesis we utilise an inferential language model to recommendation-making approach. The adaptation of language model to item recommendations will be discussed in Chapter 4.

2.5 RECOMMENDATION EVALUATIONS

A variety of evaluation methods have been suggested for use in recommender systems. The choice of method depends on whether an off-line or an online experiment is being conducted. Online evaluations require interaction with real users and most existing studies focus on off-line experiments, perhaps because online experiments are expensive and time-consuming to conduct. Similarly, given the limited scope of this thesis, only off-line recommender-related experiments have been conducted. Some popular off-line evaluation metrics will be reviewed in this section.

The two types of recommendations that are used in off-line recommender-system evaluations are rating predictions and list recommendations (or Top-N recommendations) [44]. Rating prediction tasks are utilised to predict scores for missing rating values. The Top-N recommendation format is used to recommend a list of items to the target user selected from a set of candidate items [44, 67]. If the list is ordered based on expected user preferences, then it is termed a rank list recommendation. In most recommender systems, a performance evaluation of the recommendation algorithm is conducted in order to assess the accuracy of predictions and/or the relevance of item recommendations to a user's interests. Typically, recommendation approaches can be evaluated by looking at two factors: *effectiveness* and *efficiency*. In this thesis, the performance of the recommendations will be evaluated based on the effectiveness of the proposed recommendation-making and user profiling approaches. The effectiveness of the proposed approaches will be assessed in terms of Top-N recommendation task.

Most recommender systems assess the effectiveness of recommendations by looking at recommendation accuracy metrics. There are three major types of accuracy measurement metrics for recommender systems: (1) predictive accuracy metrics, which measure the similarity between true user ratings and recommender systems' predicted ratings; (2)

classification accuracy metrics, which measure a recommender system's ability to select high-quality items from the set of items for a given user; and (3) rank accuracy metrics, which measure a recommender system's ability to recommend the right order of items in a list to the user [28, 44].

In rating prediction tasks, the systems are typically evaluated using predictive accuracy measures, where the predicted ratings are compared directly with actual user ratings [76]. In other words, these evaluations measure how close the predicted ratings are to the true user ratings. The predictive accuracy error metrics commonly applied here include the *mean absolute error* (MAE), *mean squared error* (MSE), *root mean squared error* (RMSE), and *normalised mean absolute* (NMAE) metrics. MSE and RMSE use squared deviations to measure the differences between the actual ratings and predicted ratings, and thus the results emphasise large prediction errors. MAE is the average difference between the predicted and actual ratings for a given set of items.

In contrast, Top-N recommendation tasks are usually evaluated via classification accuracy measures, in which the correct and incorrect decisions that a recommender system makes are examined [44, 121]. The performance of Top-N task can be directly measured by common methodologies based on accuracy metrics (i.e., precision and recall) [23]. Precision and recall are the basic measures for information retrieval effectiveness. They are applied to evaluate the relevant and irrelevant items in recommendation tasks [97]. In the field of information retrieval, $Precision(P)$ is the fraction of retrieved documents that are relevant, defined by Equation 2.2. $Recall(R)$ is the fraction of relevant documents that are retrieved, shown by Equation 2.3 [41, 72].

$$Precision(P) = \frac{|\{relevantdocuments\} \cap \{retrieveddocuments\}|}{|\{retrieveddocuments\}|} \tag{2.2}$$

$$Recall(R) = \frac{|\{relevantdocuments\} \cap \{retrieveddocuments\}|}{|\{relevantdocuments\}|} \tag{2.3}$$

However, precision and recall are computed using unordered sets of documents. In order to evaluate the ranked retrieval results, these measures can be evaluated at a given cut-off rank in the result list and only the topmost results are considered and returned by the system. This measure is called *precision at N and recall at N* [22, 28, 97, 107]. They are thus defined as:

$$P@N = \frac{r}{n}, \tag{2.4}$$

$$Recall@N = \frac{r}{R}, \tag{2.5}$$

where the value of N is the number of documents selected based on an assumption about how many the user will view, and r is the number of relevant documents that have been retrieved at rank N. R denotes the total number of relevant documents.

Other classification accuracy metrics that relate precision and recall and can be applied for evaluating the Top-N recommendations include *Mean Average Precision* (MAP), a popular metric for search engines which is applied to calculate the mean of the average precision of all users [97, 99], and $F_1 measure$ or $F_1 score$, which tries to combine precision and recall into a single score by computing different types of means for both metrics [36, 44, 47, 97].

Since recommender systems have to deal with the growth of data and are expected to provide rapid recommendations, computational efficiency is another important factor in evaluating the recommender algorithms [36, 44]. A common approach to evaluate the efficiency of a recommender algorithm is to measure the amount of processing time or response time needed to generate a single recommendation. *Big-O notation* is widely used in computer science to describe the performance or complexity of an algorithm based on its running time [11]. It is a useful method to analyse algorithms for efficiency. Shi et al. [99] utilised Big-O notation concept to evaluate the efficiency of their algorithms. Another approach is using higher performance hardware. However, the efficiency evaluation plays a less important role than the effectiveness evaluation for a recommendation approach.

Although the accuracy of recommendations is important, there are many other factors that can affect user satisfaction. Users have different experiences and needs while using a recommender system. Even if a recommender system generates highly accurate recommendations, some users might find that the system makes unhelpful and boring suggestions. Therefore, other facets for recommender evaluation should be considered, such as coverage, confidence, trust, novelty, serendipity, diversity, risk, robustness and privacy [98].

Coverage evaluation measures the percentage of items for which a recommender system is capable of making a prediction or recommendation to users [34]. Novel and serendipitous evaluation are used to calculate the degree to which a system offers items that are both attractive and surprising to users. Confidence and trust measurements determine the extent to which a system provides reasonable recommendations and helps users to make effective decisions. Diversity evaluations examine the sum, average, minimum and/or maximum distances between item pairs and measure the value of adding new items to recommendation lists [36, 44].

There are many different evaluation metrics that can be applied to recommender systems, but these tend to be specific to the research problems being investigated in each particular study. Applying proper evaluation metrics may result in better-quality recommender systems. The evaluation of the proposed recommendation approaches will be discussed in Chapter 5.

2.6 CHAPTER SUMMARY

This chapter has reviewed existing studies related to user profiling and recommender systems. The goal of user profiling is to collect information about the subjects or topics in which a given user is interested. User profiles are the main source of information through which personalisation systems can learn about users' interests or preferences. As the literature review indicates, information about the interests of users can be obtained either by asking users direct questions (commonly called explicit feedback) or by indirect means (referred to as implicit feedback).

Most personalised recommendations have focused on achieving recommendation accuracy. A major challenge within user profiling is how user profiles can be constructed that accurately reflect users' preferences. This includes the question of how to obtain new users' preferences if the available information is limited. Besides using ratings data, additional information about users and items can be harnessed to enhance the performance of recommender systems and address problems, such as the cold-start problem and the sparsity problem. Recently, the usage of item taxonomy has been emphasised by many researchers seeking to improve recommendation performance and alleviate the cold-start problem. Taxonomy information is expensive to gather, but it is considered to be better structured and more widely applicable than standard item content information. In addition, a few studies suggest the use of a language model with the information retrieval (IR) method to solve recommendation-related problems, as proposed here.

The following chapters will extend existing knowledge by offering effective approaches to using item taxonomy information to obtain the personal preferences of a user when user rating data and profile information are limited. Furthermore, new methods based on the use to taxonomy information for improving performance in recommender systems will also be examined in this thesis.

Chapter 3

ACQUIRING USER INFORMATION NEEDS BY CONCEPT HIERARCHY

Acquiring information about user preferences is an important task because it is the initial step in building personalised recommendations. Every recommender system has to develop a user model or a user profile that contains the personal preferences of the user. The challenge of making personalised recommendations is the effective acquisition of user preferences (or needs) when there is limited personal data about users, such as when new users have only rated a small number of items [19, 102]. The system cannot generate accurate recommendations for these users, which is commonly called the cold-start problem.

A variety of techniques to solve this problem have been developed, such as discovering user preferences from user interaction with specific items [7], maintaining a user profile, building a model of user preferences to identify the needs of individual users [48], or using additional information about users, such as gender, age, and geographical location and items, such as genres, products categories, keywords, and product descriptions [32, 57, 75, 85]. In addition to using ratings data to capture users' item preferences, item taxonomic information is another popular textual source of information and provides an alternative data source for acquiring user preferences [67]. Taxonomy information is used widely on e-commerce sites. Item taxonomy contains a set of categories or topics designed by web managers or website experts based on their personal experiences. Items can be described or classified according to sets of categories or topics.

This thesis provides a unique method to solve the cold-start problem, an alternative way to deal with uncertainties when acquiring user preferences. Instead of utilising user ratings, this thesis proposes a novel *concept hierarchy model* to determine user interest in items according to taxonomic concepts that the user uses and which relate to the items in question. The user preferences and item content can be represented as a *user concept hierarchy* and *item concept hierarchy*, respectively. This chapter proposes using item taxonomic information as domain-specific knowledge.

3.1 PROBLEM DEFINITION

Most recommender systems make recommendations based on users' item preferences, which are extracted from user ratings data. The performance of recommender systems is diminished when they have only a few ratings or insufficient personal data. The recommender systems cannot surmise the relationship between users and items when there is insufficient information, such as the first time a new user visits a system or when a new item is added and has no ratings data available. The challenge in this context is how to obtain the personal preferences of new users when the systems have insufficient information with which to generate high-quality personalised recommendations. This requires the development of new techniques to alleviate the problem and produce better recommendations.

In addition, user preferences involve fuzzy information or knowledge. Fuzzy knowledge can be defined as information or concepts that are vague, imprecise, uncertain, or ambiguous in nature. Most users do not know how to describe exactly what they want or choose the right information to fits their personal needs. Consequently, the systems cannot accurately capture user preferences or needs, which leads to poor recommendations. Therefore, to profile users accurately, we must discover how to help them to accurately identify their own needs from the fuzzy information in their minds.

In addition to using ratings, other factors that can be used to learn user preferences, such as their topics of interest, the order of items in the recommended item list, or the users' taxonomic topic interests [44, 67, 111, 121]. To address the problem, additional information about the users or items must be exploited [75, 85]. This thesis demonstrates how to item taxonomic information to identify user information needs and preferences to support user profiling and to help deal with the aforementioned recommendation-making problems.

3.2 NOTATIONS AND BACKGROUND

Before delving into algorithmic details, this thesis provides the theoretical background and definitions that are relevant to this research and used in subsequent chapters. An overview of concept taxonomy, basic notation and the concept hierarchy model are outlined in the next section.

3.2.1 An Overview of Concept Taxonomy

Item taxonomic information is based on a set of categories or topics that can be used to classify and describe items in a hierarchical structure, from coarse-grained classes to fine-grained classes. Item taxonomy is often described in product descriptions, which are provided by domain experts [46, 67] and designed to help users find their preferred items

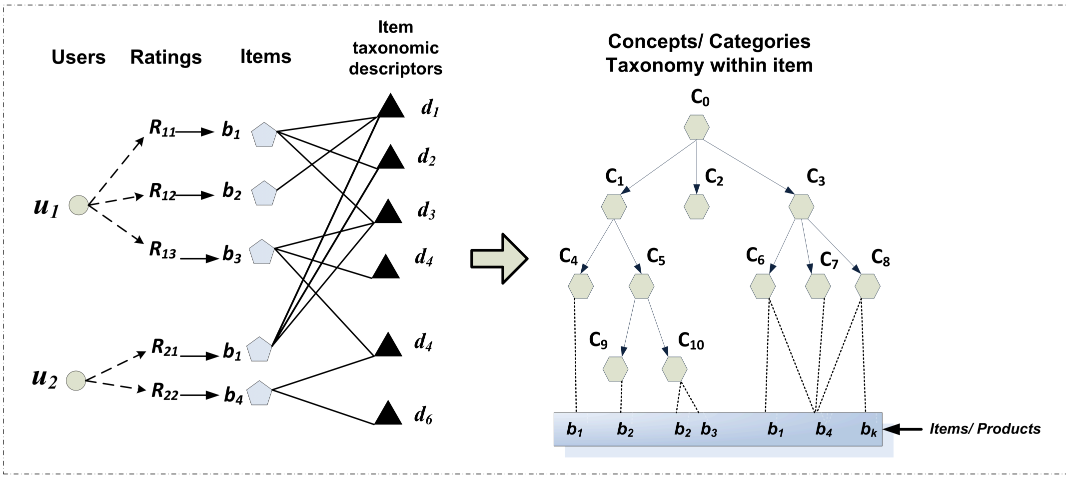

Figure 3.1: Taxonomic information for item representation concepts (or categories)

or products easily and quickly. One of the main advantages of taxonomy is that category correlations within item taxonomies represent the hierarchical relationship between categories. There are also other advantages, including implicit feedback data, standard vocabularies and the fact that it is not vague [67]. In addition, a taxonomy's hierarchical structure can also reflect users' topics of interests, from general topics of interest at the root nodes to specific topics of interest at the leaf nodes [52]. This enables user interest in items to be linked with the taxonomic information of those items. In short, information on a user and his or her preferences can be learned from an item's taxonomic information.

Figure 3.1 illustrates an example of concept taxonomy. There are two users who gave ratings to the items. Each item can be described or classified with multiple descriptors, each containing a set of categories (or concepts) that form a path in the concept taxonomy (see the right side of Figure 3.1). In general, product categories can be naturally organised into hierarchies, where the root category of a hierarchy (e.g., a tree) is the most general and the categories become more specific towards the leaves.

In a tree structure, one branch tree may have the category name *computer technology* as the concept and the child categories of *programming*, *database* and *web application* as sub-concept, and another branch may have the concept name *business* and the sub-concepts of *marketing* and *management*. To clearly show the proposed concept hierarchy approach, we assume that item represents a product and each concept represents a category. Additionally, the nodes in the concept taxonomy represent concepts (or product categories). The concepts start generally at the root node of the hierarchy and become more specific towards the leaves. Therefore, it is possible that the affinity of user preferences to one item can be linked by concept taxonomy to some of that item's connections.

3.2.2 Basic Notation

To clearly show the proposed concept hierarchy, formal definitions of some other concepts and entities relating to item taxonomy are listed below:

- **Users:** $U = \{u_1, u_2, \ldots, u_m\}$ is a set of users, where $u_i \in U$ means the user u_i who has either browsed the item or contributed ratings of the item. The aim of the proposed recommender system is to create recommendations for a user u_a, who we call an active user or a new user.
- **Items or products:** $B = \{b_1, b_2, \ldots, b_n\}$ is a set of items or products (e.g., books and music tracks) that have already been rated by users $u_i \in U$. Items $b_k \in B$ are represented by descriptors.
- **Explicit ratings:** R_{ik} denotes the users u_i who express their opinion about items b_k via ratings. R_{ik} indicates the preference by user u_i of item b_k, where high values mean stronger preferences. Users u_i can express their preferences for items in numeric form. That is, value 0 indicates a user's dissatisfaction with the item and a value of 1 or greater indicates their satisfaction with the item. In this thesis, the

explicit rating R_{ik} values between 1 and 10 are utilised to conduct the experiments.

- **Item preferences:** A user's preferred items can be classified into two groups: explicit item preferences and implicit item preferences. *Explicit item preferences* are collected from users when they directly express how much they like an item on numeric number scale (i.e. explicit item rating). *Implicit item preferences* are automatically obtained from each user's behaviour or navigation. This is usually represented by a set of binary numbers (0, 1) called implicit ratings. Implicit item preferences do not give a clear indication of a user tastes, opinions, or potential emotional involvement with items in the system. However, it is assumed that if a user clicks on an item, they have some kind of interest in it, even if they do not like it.

- **Concept taxonomy:** T is a pair $(C, `is - a')$, where $C = \{c_1, c_2, \ldots, c_w\}$ is a set of concepts (or categories), and the concept correlations within item taxonomies are organised in a tree or hierarchical structure. The 'is-a' relationship represents the hierarchical relationship between concepts. For example, c_x is-a c_y (or $c_y > c_x$), which means c_y is a super-concept of c_x, and c_x is a sub-concept of c_y. Typically, concepts can express either broad (super, or general) categories or narrow (sub or specific) categories. The root concept node is the most general concept, and the concepts become more specific towards the leaf nodes within the concept taxonomy.

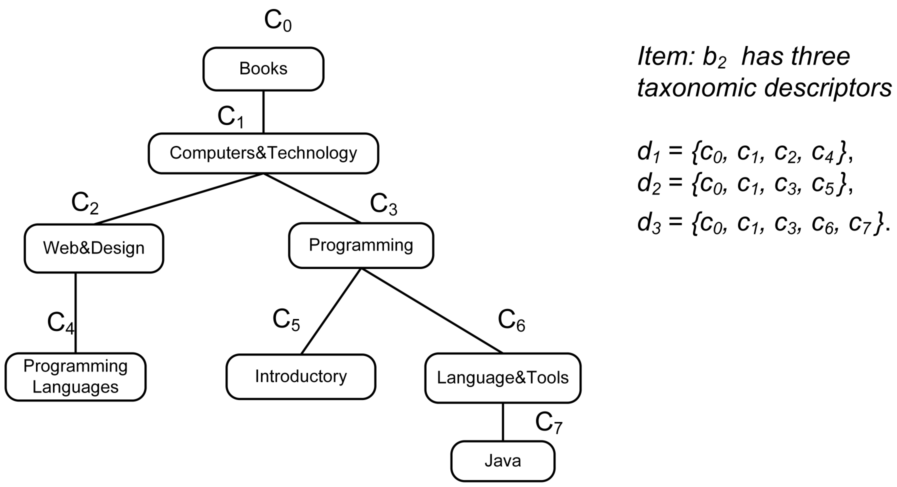

Figure 3.2: The concept correlations within the item taxonomy represent the hierarchical relationship between concepts

Figure 3.2 shows an example of concept correlations within an item taxonomy that represent the hierarchical relationship between concepts. Supposing that the book b_2 in Figure 3.2 is associated with the three item taxonomic descriptors: $d_1 = \{c_0, c_1, c_2, c_4\}$, $d_2 = \{c_0, c_1, c_3, c_5\}$, $d_3 = \{c_0, c_1, c_3, c_6, c_7\}$, the item b_2 can be described or classified by eight taxonomic concept $c_0, c_1, c_2, c_3, c_4, c_5, c_6$ and c_7. Within the item taxonomy tree, the taxonomic concepts correlations of the given item b_2, which can be described as *book*, is a root category or concept; also, *book* is a super-concept of *Computers&Technology*. The one sub-concept of the leaf nodes is *Java*, which is the most specific concept in the item taxonomy tree.

- **Item taxonomic descriptors:** $D_{b_k} = \{d_1, d_2, \ldots, d_v\}$ is a set of item descriptors where each descriptor is a sequence of concepts based on the concept taxonomy relation T. As shown in Figure 3.3, item b_k can usually be described or classified using item taxonomic descriptors. An item can be described with multiple descriptors. For example, book b_2 *Java Programming* in Figure 3.3 has the following three item taxonomic descriptors:

$d_1 =$ *Books* $>$ *Computers&Technology* $>$ *Programming* $>$ *Introductory*

$d_2 =$ *Books* $>$ *Computers&Technology* $>$ *Programming* $>$ *Languages&Tools* $>$ *Java*

$d_3 =$ *Books* $>$ *Computers&Technology* $>$ *Web&Design* $>$ *Programming Languages*

That is, $D_{b_2} = \{d_1, d_2, d_3\}$.

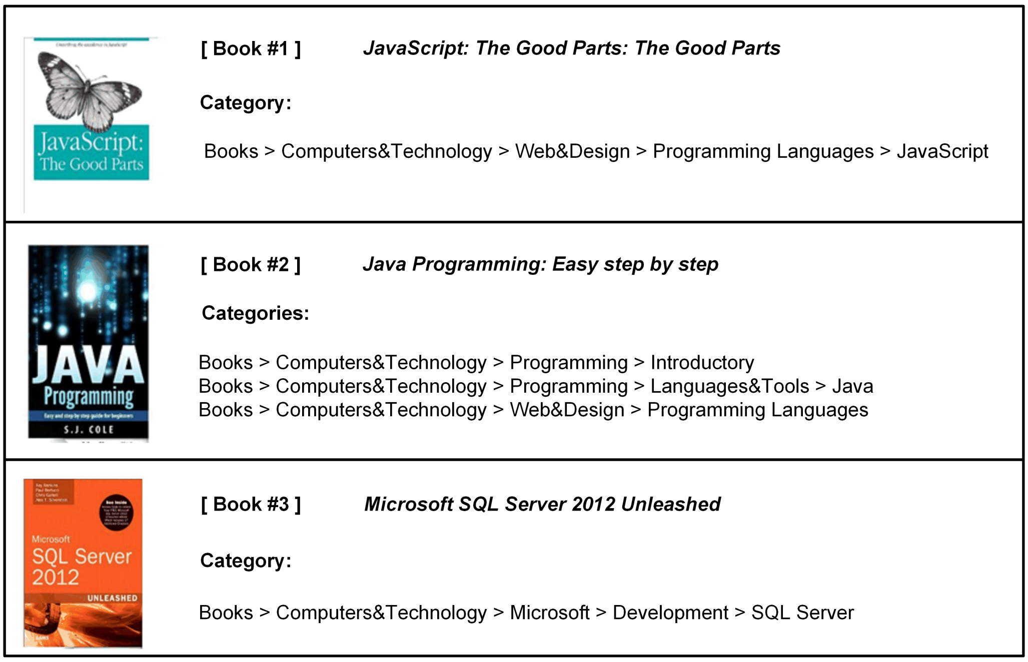

Figure 3.3: The example list of items with their taxonomic descriptors

3.3 A FRAMEWORK FOR THE CONCEPT HIERARCHY MODEL

This section describes the proposed approach of a concept hierarchy to obtain user preferences or information. We also present a method to evaluate the importance of concepts in a concept hierarchy. The main objective of this thesis is to enhance personalised recommendation approaches. Typically, most recommender systems make recommendations based on ratings that users have assigned to items. However, because the amount of user ratings data is insufficient to capture user preferences, other information about users and items can be used to learn and obtain user preferences.

Inspired by the approach of Ziegler et al. [121] and Weng et al. [111], this thesis provides an effective way to summarise useful knowledge about existing items, which significantly contributes to the cold-start problem. To formalise the problems, this thesis focuses on using item taxonomic information as domain-specific knowledge. The structural information of the taxonomy tree structure and the relationship between users and items according to the taxonomic categories utilised by users are taken into consideration when generating a new structure to determine user preferences (or needs).

This thesis proposes an effective novel framework for using *the concept hierarchy model* to gauge the interest a user u_a has in an item b_k according to a set of taxonomic concepts that the user interacts with and the item has. The rationale for this is that users may have interest in common concepts, even though they have not rated the same items. We provide a method to help systems perform the same role as human beings, in that it allows them to make assumptions and extrapolate a person's interests and abilities from limited information. Some information is more useful than others. For example, from knowing that a person has shown an interest in a toddler's book about counting, it would be a sound assumption that this person has children in his or her life and would therefore also be interested in a whole range of products and topics or categories related to children.

Instead of representing user preferences using only ratings data, the system can learn these preferences (or interests) through the concept taxonomy of items in two directions of the hierarchy: vertical and horizontal. The basic idea of a concept hierarchy is that a user has preferences that should be associated with concepts in both dimensions of the hierarchy to reflect the fact that user preferences are broader than single items. The vertical direction describes concept relations as 'is-a' (general to more specific) relationship between concepts, and the horizontal direction describes a sequence of concepts. A user's interest in the concepts on the horizontal span (on the same level) indicates his or her preferred concepts and provides a list of priorities.

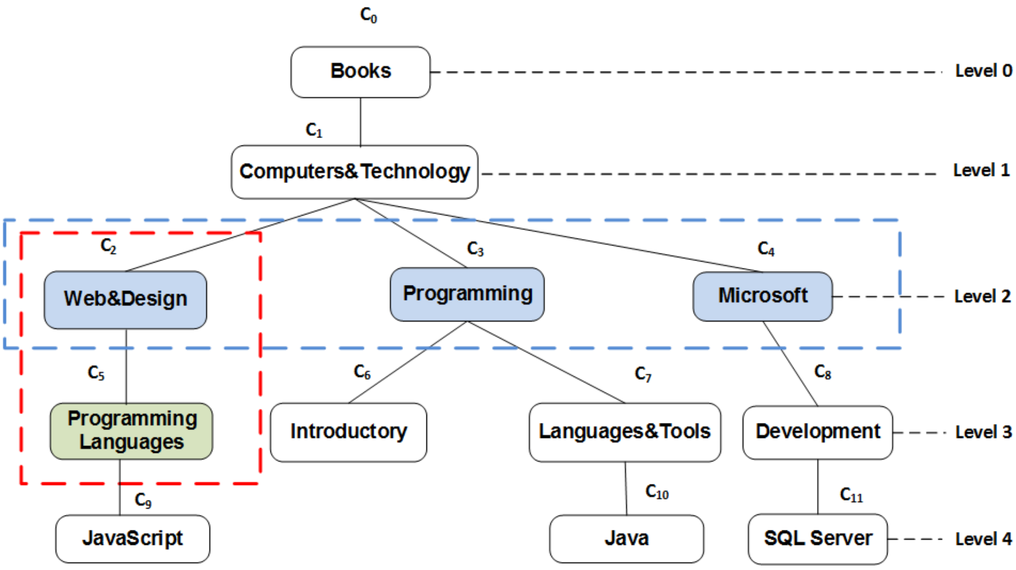

Figure 3.4: An example fragment of an item taxonomy extracted from item taxonomic descriptors

For example, in Figure 3.4, in the domain of books, we have a concept *Computers&Technology* and sub-concepts *Web&Design, Programming, Microsoft*, etc. in which *Web&Design*, *Programming* or *Microsoft* have an 'is-a' concept relationship with *Computers&Technology*. A user may like both *Computers& Technology* and *Web&Design* from a general interest to a more specific interest (vertical). As there are three sub-concepts, *Web&Design*, *Microsoft* and *Programming*, at the same horizontal concept level, the user may like both *Web&Design* and *Programming*, but might regard *Web&Design* as more important than *Programming*, based on their interest in the concept from a specific interest to a more general interest. In other words, the interests of the user in each concept are a user's horizontal list of concept priorities. It is convenient for users to describe what they want in the concept hierarchy without any value or weight. Therefore, user preferences can be represented by the user concept hierarchy. Likewise, the relevance of the item to the taxonomic concepts can be represented by the item concept hierarchy.

This model provides a novel structure for new users to approximate their needs in a two-dimensional hierarchy. We can easily transfer a set of rated items into a concept hierarchy using item taxonomic information. As such, this model is also applicable to new users who have only rated a few items.

3.3.1 The Definition of Concept Hierarchy

To better understand how to construct the representation of the user and the item based on concept hierarchy, the following concepts must be included:

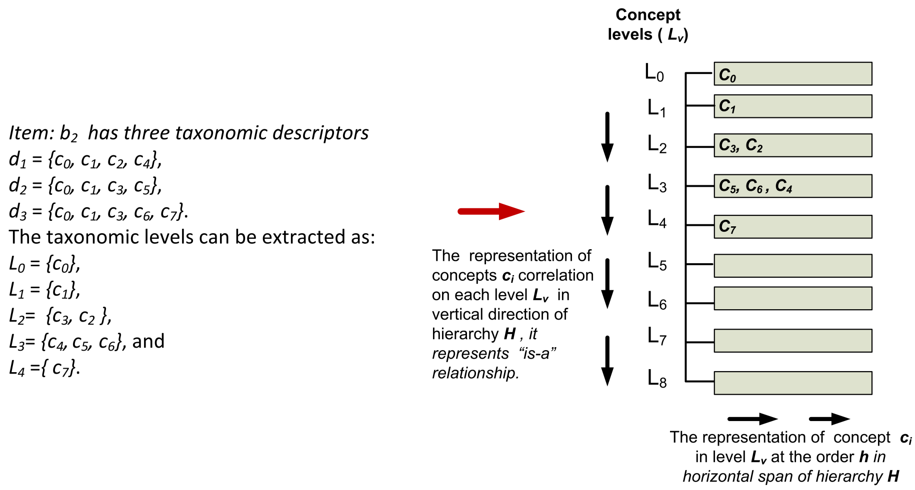

Figure 3.5: An example of item concepts taxonomy transferred into a concept hierarchy model

- **Concept preferences:** user interests or preferences for concepts or categories of items. Concept preferences record a user's preferred concepts and can be obtained explicitly or implicitly. Explicit concept preferences can be represented as a set of keywords, categories, topics or concepts provided directly by the users. The users can explicitly declare which topics they are interested in, such as through search queries or concept/category interests defined in his or her user profile. Implicit concept preferences can be represented by a set of keywords that are extracted from the content or taxonomic concepts of the items that the user clicks, buys, browses, rates, or tags. The implicit concept preferences are generated automatically based on user behaviours.

- **The concept hierarchy** $H = \{L_1, L_2, \ldots, L_p\}$ is defined as a set of concept levels where each level $L_i \in H$ includes a sequence of concepts. Level L_i can be assigned to each node in the concept taxonomy. The first level starts from the root node. The level number of root node is defined as 1, and the number of the other levels increases towards the leaf nodes by one plus the level of its parent. The user preferences in the concept taxonomy c_i can be described in two directions of the hierarchy H.

  In the vertical hierarchy, because the relationship between concepts is an 'is-a' relationship, L_1 is the top level, or parent node, that describes the most general concepts and L_p is the bottom level that describes the most specific concepts. We can indicate that the interest of the user u_a to the concept taxonomy c_i in the top level of the hierarchy is more general, and it becomes more specific towards the lower levels or leaf nodes. Given that concept $c_y > c_x$ or c_x is-a c_y, that means c_y appears in one of the upper levels and c_x appears in one of the lower levels, or the leaves. Horizontally, the interest of the user in each concept can be indicate by his or her list of concept priorities. On the same level of the hierarchy, the order of the concepts on the list is stored and sorted from left to right according to their importance. When two concepts $c_a, c_b \in C$ are on the same level, a user may be more interested in c_b than c_a.

  Figure 3.5 gives an example of how to generate a concept hierarchy from item taxonomic descriptors. Given an item taxonomic descriptor $d_x = c_1 > c_2 > ... > c_t$, if c_i is in level L_x, then c_{i+1} will be in level L_{x+1} for all $1 \leq i \leq t-1$. In Figure 3.5, there are three item taxonomic descriptors given to the item b_2:

$d_1 = < c_0, c_1, c_2, c_4 >$;

$d_2 = < c_0, c_1, c_3, c_5 >$;

$d_3 = < c_0, c_1, c_3, c_6, c_7 >$.

There are five concept levels that can be extracted. At each level L_i, the horizontal order of concepts on the list is sorted from left to right according to the frequency of the concepts occurrence in descriptors:

$L_0 = \{c_0\}$;

$L_1 = \{c_1\}$;

$L_2 = \{c_3, c_2\}$;

$L_3 = \{c_4, c_5, c_6\}$;

$L_4 = \{c_7\}$

Please note that the concept c_0, books, is the common root concept and we assign c_0 in L_0. In this paper, we ignore c_0 and start the meaningful concepts from L_1. Therefore, the concept structure starts from L_1 to L_p.

3.4 REPRESENTING ITEMS IN A CONCEPT HIERARCHY

Taxonomic information provides an important information source about items and users. Because an item $b_k \in B$ can generate the taxonomic descriptors D_{b_k} that characterise users' item preferences, the users are likely to be interested in the set of taxonomic concepts associated with those items. Typically, item taxonomy is described in product (or item) taxonomic descriptors. An item's taxonomic descriptor is a set sequence of taxonomic concepts that are organised into 'is-a' hierarchies, where the root category of the hierarchy is the most general category and the categories become more specific towards the leaves. Each item can be described according to a set of taxonomic concepts.

The hierarchical structure of a taxonomy provides the constructive knowledge for finding the relationship of an item to a taxonomic topic.

One purpose of this thesis is to be able to solve the new item cold-start problem. To do so, it proposes item representation based on a concept hierarchy model, *an item concept hierarchy*. It provides a new structure for representing the relevance of item $b_k \in B$ to the set of taxonomic concepts $c_i \in C$ in the two-dimensional hierarchy. This adjustment to the item concept hierarchy model captures user preferences faster than waiting for them to input ratings. The proposal in this thesis enhances item concept hierarchy by examining individual items' taxonomic concepts or categories. The following factors should be considered when generating the weight of a taxonomic concept $c_i \in C$ for the representation of the item $b_k \in B$ in concept hierarchy H_{b_k}:

- The structural information of an item's taxonomic concept or category hierarchy. Based on the hierarchical relationship between taxonomic concepts, when the two taxonomic concepts $c_x, c_y \in C$ are organised into 'is-a' relationships in hierarchies, c_x is-a c_y; therefore, c_x is a sub-concept of c_y and c_y is a super-concept of c_x. Also, c_y appears in one of the upper levels of the hierarchy and c_x appear in one of the lower levels. If c_x expresses a more specific concept than c_y, then c_x and other concepts at the leaves or lower levels should have a higher weight value than c_y and general concepts at the root node or the upper level within the concept taxonomic tree structure.

- The frequency of a taxonomic topic's occurrence on the same level of the hierarchy. The horizontal importance of the concepts for an item b_k can be determined using the frequency of each taxonomic concept c_i occurring in item taxonomic descriptors D_{b_k}. For example, the three concepts $c_x, c_y, c_z \in C$ are on the same level L_v (horizontal) of the hierarchy. If concept c_y appears more frequently on the same level L_v than concept c_x and

c_z, then c_y is more important than c_x, c_z and other concepts on the same level of hierarchy. c_y will then be organised as the first concept in the concepts's horizontal list of priorities. The concepts stored on the same level are ordered from left to right according to their importance. Therefore, c_y should have a higher weight value than the taxonomic concepts that occur less frequently in D_{b_k}.

3.4.1 A Framework for Item Representation based on Concept Hierarchy

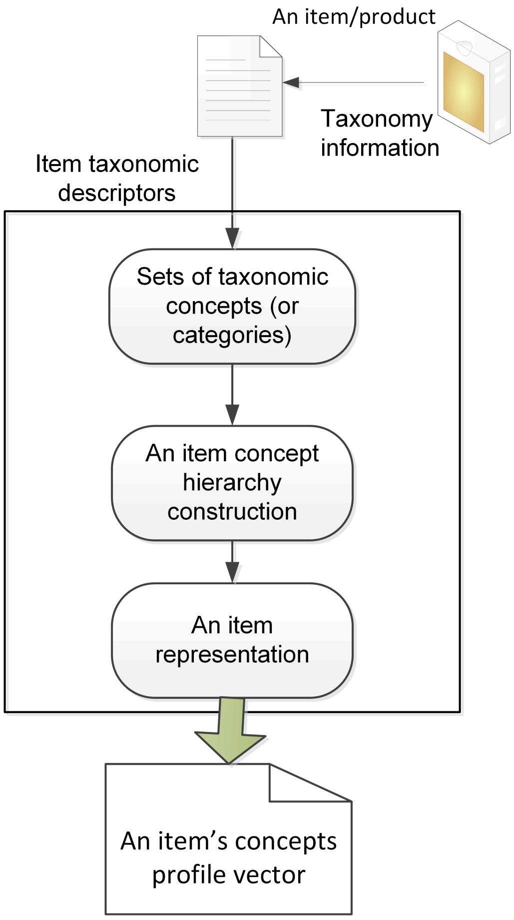

Figure 3.6: The framework for an item's representation based on concept hierarchy model

Figure 3.6 shows the framework for the item representation based on concept hierarchy. To generate an item's concept hierarchy H_{b_k}, the input is the taxonomic concept C within the item's taxonomic descriptors D_{b_k}. In this scenario, we assume that a new user has rated only a few items. A given set of rated items can be represented in an item concept hierarchy, as previously discussed. Given an item descriptor $d_x = c_1 > c_2 > ... > c_t$, if c_i is in level L_x, then c_{i+1} will be in level L_{x+1} for all $1 \leq i \leq t-1$. The horizontal order of concepts in hierarchy is sorted based on the frequency of the concept's presence in its descriptors. The vertical order of concepts on the list in the hierarchy is sorted based on the hierarchical relationships between each concept. The relationship of item b_k to taxonomic concept c_i can be measured by the weight of how much the item b_k is related to taxonomic concept c_i. The weight of concept c_i can be computed by integrating the concept's weights in the two directions of the hierarchy.

The output is the item concept hierarchy. The item concept hierarchy H_{b_k} represents the relevance of the item b_k and the taxonomic concept c_i in the two dimensions of the hierarchy H. The item concept hierarchy is characterised by a vector of dimension $|C|$. Each entry represents the relationship of the item b_k to the taxonomic concept c_i in two-dimensional hierarchy H. The profile of an item b_k is defined as a vector $\overrightarrow{b_k}$ of weights for each concept. The process of computing the concept weight for an item b_k based on concept hierarchy H is described in subsection 3.4.2.

3.4.2 The Construction of the Item Concept Hierarchy

This section describes the processes for measuring the relevance of an item b_k to a taxonomic concept c_i in the concept hierarchy model. The three main steps to construct the item concept hierarchy are shown below. Figure 3.7 shows the overview of the processes to calculate the concept weight of an item b_k based on the concept hierarchy model.

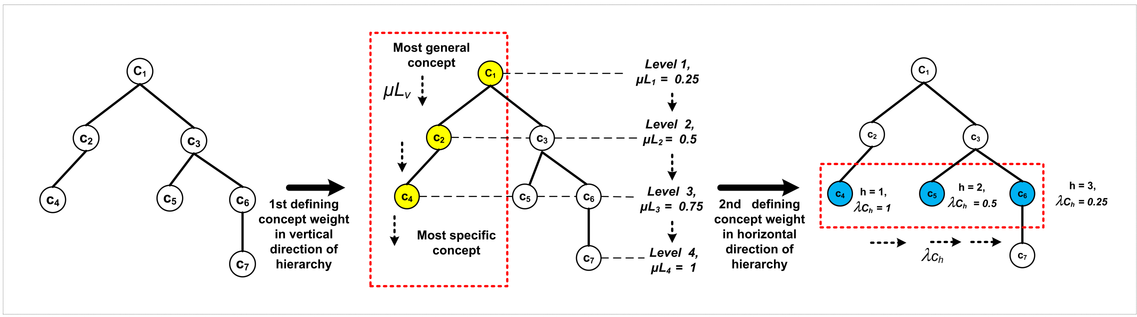

Figure 3.7: An overview of the steps required to generate the weight of taxonomic concepts for the representation of an item based on the concept hierarchy

Step 1: Defining the vertical importance of levels in the hierarchy

The structural information of the taxonomic tree structure is considered when defining the weight of the vertical taxonomic concepts of the hierarchy for an item b_k. The weight value of each vertical taxonomic concept is based on the weight of its level in the hierarchy. Given an item descriptor $d_x = c_1 > c_2 > ... > c_t$, if c_i is in level L_x, then c_{i+1} will be in level L_{x+1} for all $1 \leq i \leq t-1$. The relationship between each vertical taxonomic concept $c_v \in C$ of the hierarchy is an 'is-a' relationship, the root concept of a hierarchy (i.e. the tree) is the most general category and the concepts become more specific towards the leaves. Therefore, the set of concepts $c_v \in C$ in the leaf nodes will have a higher weight value than the taxonomic concepts on the top level.

$H_{b_k} = \{L_1, L_2, \ldots, L_p\}$ is a concept hierarchy for the item b_k, and μ_{L_v} is the most important factor of level v in the hierarchy. The weight of concept c_v in each level L_v of the hierarchy for item b_k is defined according to the values of μ_{L_v}. This thesis uses the following constraints to decide the values of μ_{L_v}:

$$\mu_{L_p} = 1;$$

$$\mu_{L_v} > \mu_{L_{v-1}}; \quad and$$

$$\mu_{L_v} = (v \times \mu_{L_p})/p$$

For all $v = 1, 2, \ldots, p$, where p is the maximum level of the hierarchy H_{b_k}, μ_{L_p} has the highest value of the concept levels.

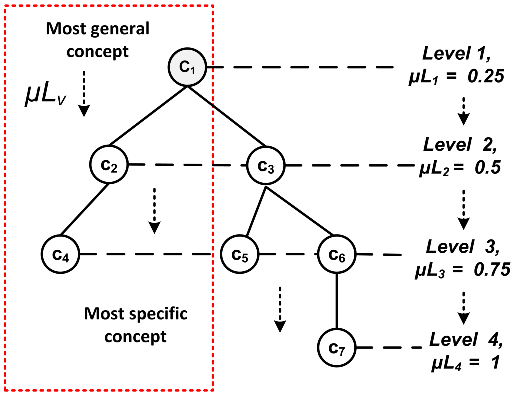

Figure 3.8: An example of taxonomic concepts organised into the hierarchical structure

For example, there are three-item taxonomic descriptors given to item b_2: $d_1 = \{c_1, c_2, c_4\}$, $d_2 = \{c_1, c_3, c_5\}$ and $d_3 = \{c_1, c_3, c_6, c_7\}$. There are four vertical levels in the hierarchy $H_{b_2} = \{L_1, L_2, L_3, L_4\}$ as follows: $L_1 = \{c_1\}$, $L_2 = \{c_3, c_3, c_2\}$, $L_3 = \{c_4, c_5, c_6\}$ and $L_4 = \{c_7\}$. Based on the concept hierarchical structure H of item b_2 in Figure 3.7, the maximum level of the hierarchy H_{b_2} is $p = 4$; thus, $\mu_{L_4} = 1$ and the weight of c_7 = 1. Based on the above constraints, the weight of c_4, c_5 and c_6 on the upper level L_3 can be computed as $\mu_{L_3} = (3 \times \mu_{L_4})/4 = 0.75$. The concept weight of c_2 and c_3 in level L_2 can be computed as $\mu_{L_2} = (2 \times \mu_{L_4})/4 = 0.5$. The concept weight of c_1 in level L_1 can be computed as $\mu_{L_1} = (1 \times \mu_{L_4})/4 = 0.25$.

Step 2: Defining the horizontal importance of concepts in the hierarchy

The concept weight c_h is defined by the horizontal order of the concept. On the same level, L_v, concepts are sorted from left to right according to their importance. The first concept (whose index number is 0) in the sequence is the most important one, and concepts on the left side are more important than those on the right side. For item b_k, the order of concepts is determined by the frequency of the concept's presence in item taxonomy descriptors. The frequency of the concepts on the same level L_v for item b_k can be computed by counting. The most frequent concept becomes the first concept in the sequence.

Likewise, if the concept c_h appears more frequently than other taxonomic concepts, c_h will have a higher weight than the taxonomic concepts that occur less frequently. This thesis defines λ_{c_h} to represent the horizontal importance of concept c_h, which is calculated as follows:

$$\lambda_{c_h} = \frac{1}{2^h}, \tag{3.1}$$

where h is the horizontal index number of concepts c_h in the concept sequence in level L_v, $h \geq 0$.

For example, the item b_2 has the three taxonomic descriptors D_{b_2}: d_1 = $\{c_1, c_2, c_4\}$, d_2 = $\{c_1, c_3, c_5\}$ and d_3 = $\{c_1, c_3, c_6, c_7\}$. There are four vertical levels in the hierarchy; therefore, $H_{b_2} = \{L_1, L_2, L_3, L_4\}$ as follows: L_1 = $\{c_1\}$, L_2 = $\{c_3, c_3, c_2\}$, L_3 = $\{c_4, c_5, c_6\}$ and L_4 = $\{c_7\}$. At each level L_v, the concepts are sorted and organised from left to right according to the frequency of their occurrence. In level L_2, if the concept c_3 appears more frequently than c_2, then c_3 will be stored and organised as the first concept and c_2 will be next in the sequence. Thus, the horizontal order of the concepts for item b_2 can be stored and organised as: L_1 = $\{c_1\}$, L_2 = $\{c_3, c_2\}$, L_3 = $\{c_4, c_5, c_6\}$, and L_4 = $\{c_7\}$. Based on the above constraints, the weight value of concept c_3 can be calculated as λ_{c_1} = 1 and c_2 can be calculated as λ_{c_2} = 0.5, based on their importance.

Step 3: Defining the vertical and horizontal importance of concepts in the hierarchy for a given item

The concept's weight in the two-dimensional hierarchy for the given item b_k can be calculated by integrating the concept's weight in the two directions of the hierarchy: vertical and horizontal. The weight value of each concept c_i in L_v at the order h for the item b_k can be calculated as:

$$cw_{H_{b_k}}(v,h) = \mu_{L_v} \times \lambda_{c_h}, \tag{3.2}$$

where $cw_{H_{b_k}}(v,h)$ is the concept hierarchy weight c_i in level v at the order h.

Step 4: Transforming an item's concept hierarchy into a profile vector of concepts for that item

For the given item b_k, its content (or profile) can be described in terms of a concept hierarchy. To measure the similarity between an active user and an item according to the two profile vectors, we represent the corresponding concept hierarchy H as a profile vector of concepts for the item b_k. We further normalise the concept hierarchy weight for each item by using the min-max normalisation technique. The following equation is used to normalise the weight of the concept at position h in level L_v for the item profile vector.

$$cw'_{H_{b_k}}(v,h) = \frac{cw_{H_{b_k}}(v,h) - min_{(x,y)\in H_{b_k}}(cw_{H_{b_k}}(x,y))}{max_{(x,y)\in H_{b_k}}(cw_{H_{b_k}}(x,y)) - min_{(x,y)\in H_{b_k}}(cw_{H_{b_k}}(x,y))}, \tag{3.3}$$

where $cw'_{H_{b_k}}(v,h)$ is the normalised weight of concept c_i in level v at the order h on hierarchy H_{b_k} for the item b_k, while $cw_{H_{b_k}}(v,h)$ is the weight of concept c_i for the item b_k in level v at the order h of the hierarchy H. If $min_{(x,y)\in H_{b_k}}(cw_{H_{b_k}}(x,y))$ is the minimum concept weight value in the item's concept hierarchy H_{b_k}, $max_{(x,y)\in H_{b_k}}(cw_{H_{b_k}}(x,y))$ is the maximum concept weight value in the item concept hierarchy H_{b_k}.

Finally, each item's concept hierarchy is characterised by a vector of dimension $|C|$. A profile vector of concepts for item b_k is represented as $\overrightarrow{b_k} = (cw'_{H_{b_k}}(1,1), ..., cw'_{H_{b_k}}(1,2), ..., cw'_{H_{b_k}}(p,1), ...)$. Each entry $cw'_{H_{b_k}}(i,j)$ in $\overrightarrow{b_k}$ represents the weight of each concept relevant to the item b_k in the concept hierarchy H_{b_k}. The interestingness of item b_k to an active user u_a based on two profile vectors can then be calculated using the cosine similarity measurement.

3.5 THE DIRECT ACQUISITION OF USER PREFERENCES

The initial step in creating personalised recommendations is acquiring user preferences. Each user profile describes user interests and preferences, such as item or topic preferences. Because ratings data are difficult to obtain and insufficient at capturing user preferences, many existing recommender system researchers are exploring new techniques to utilise the limited available user information and other information sources regarding item content. For example, existing studies have exploited the relationships between users' item preferences, the taxonomy of the given item and the user ratings data to generate user taxonomic preferences under the assumption that if users have similar taxonomic preferences, they must have similar content interests [111, 121].

Since the users' ratings on the items correspond to their preferences for items, they might reveal some aspects of the item which involve with its contents or descriptions. Therefore, how to exploit these aspects to enhance the personalised item recommendation making and represent the users' preferences in more meaningful concept. Item taxonomic information is another information source that can be used to determine user preferences.

Each user is likely to be interested in the corresponding sets of taxonomic concepts associated with an item. For example, a new user who has given a four-star rating to a children's book about Iron Man may also be interested in a toy Batman.

One purpose of this thesis is to be able to gain information about the new users faster than ratings allow. By speeding up the acquisition of user preference data, a recommender system can start cross-selling products sooner. If the system can estimate user preferences using sets of taxonomic concepts of these items that are utilised by active users, it will be able to estimate the interest of the new user in those items. The impact of the cold-start problem is therefore significantly reduced.

In contrast to the work of Ziegler *et al.'s* and Weng *et al.'s* works [111, 121], this thesis proposes a concept hierarchy model to obtain user preferences and needs, which is called *an active user concept hierarchy*. This model also provides a new structure in which new users can approximate their needs in a two-dimensional hierarchy. Instead of calculating a new or active user's preferences by using ratings data, the user's preferences are characterised by a set of taxonomic concepts that make a calculated guess at their interest in a concept (or category).

The relationships between users and items are calculated according to a set of taxonomic concepts T. Any item that the user interacts with is attached to his or her profile, and the taxonomy of that item is exploited to find the user's concept hierarchy preferences. The more items the user interacts with, the more precise the preferences become. In this way, the structural information of the taxonomy tree is utilised to calculate the weight of taxonomic concept preference in both directions of the hierarchy for a given user.

In this thesis, the user's explicit feedback is taken into consideration when obtaining new users' concept preferences in a two-dimensional hierarchy. We utilise this option to solve the problem of users being confronted with uncertain information. Many websites that have a recommender system allow users to explicitly declare the concepts they are interested in. In this scenario, the system allows users to explicitly declare which concepts (rather than items) they are interested in through the concept hierarchy. Users provide their preferred concepts level by level in the direction of the hierarchy. At each horizontal level of the hierarchy, they list the order of their preferred concepts. The model of the user's preferences or the user profile can then be represented by the user concept hierarchy. This approach takes the following two factors into consideration when designing a technique to compute the weight value of taxonomic concepts in the hierarchy for a new user:

- The user's vertical concept preferences in hierarchy. According to the interests of a user in concepts in hierarchy from general to more specific, if the relationship between concepts $c_x, c_y \in C$ is an 'is-a' relationship and $c_y > c_x$, c_x is-a c_y, so that c_y is a super-concept of c_x and c_x is a sub-concept of c_y. That means that c_x expresses a more specific concept than c_y. If a user u_a is more interested in c_x than c_y, c_x is more likely to be the users' specific concept of interest. Therefore, the weight value of concept c_x should be higher than the taxonomic concept c_y or other general (high-level) concepts.

- The user's horizontal concept preferences in hierarchy. A user's horizontal concept preferences indicate his or her preferred concepts in order of priority. When two concepts $c_x, c_y \in C$ are on the same level of the hierarchy, a user may like both c_x and c_y, but c_y might be regarded as more important than c_x. Therefore, the order of the concept in the list at the same level c_y will be considered the first priority, followed by c_x. c_y should have higher weight value than c_x.

3.5.1 A Framework for Acquiring New User Information Needs Based on Concept Hierarchy

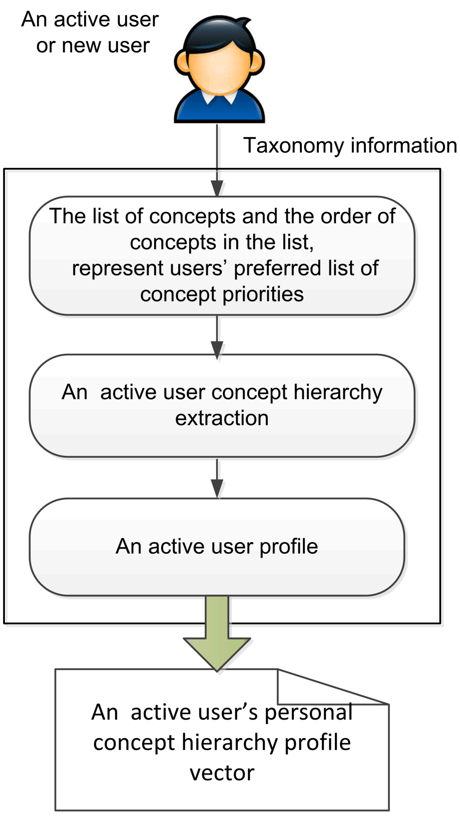

Figure 3.9: The framework for acquiring new user information needs based on concept hierarchy

Figure 3.9 shows a framework of acquiring new user information needs based on the concept hierarchy model. Given an active user represents a new user. The purpose of this framework is to obtain a user's preferences when there is limited about new user information, and to build his or her preferences more accurately by utilising the item's vertical and horizontal taxonomic hierarchy of items. This reduces the impact of the cold-start problem, which in turn increases the quality of neighbourhood forming when there are a small number of item ratings in the system.

The best way to generate a new user's profile for concept hierarchy recommender systems is to encourage the direct input of his or her interests. This list of concepts and the order of concepts in the list represent the user's preferred list of concept. They are acquired directly by asking the user to choose or declare his or her preferred concepts in two dimensions of the hierarchy. This method benefits users who are confronted by uncertain regarding their information needs. This includes enhancing the accuracy of user profiles, thereby resulting in higher quality recommendations.

In some applications of this novel approach to the concept hierarchy model, the user may have a combination of explicit and implicit preferences, or possibly only implicit ones. Regardless of whether a user provides explicit preferences, the value of implicit preferences can be enhanced by this model of utilising horizontal and vertical item taxonomic information. We provide the method to perform the same role as a human being. Instead of asking users to explicitly describe their concepts of interest, in this thesis we view the selected users in the testing set as the new user (or the active user).

For each active user u_a, the user's concept hierarchy preference can be obtained from the items that were rated by the test users. The horizontal order of concepts are extracted and sorted according to their frequency in the rated items' taxonomic descriptors. The vertical concepts are sorted based on the hierarchical relationships between each taxonomic concept. The weight of how interested the active user u_a will be in the concept c_i and can be computed by integrating the concept weight in two dimensions of the hierarchy.

The output is an active user profile. Each active user profile contains the user's personal concept hierarchy preferences H_{u_a}. The active user concept hierarchy H_{u_a} represents each active user u_a's preference in each concept c_i in two dimensions of hierarchy H. Then, the active user's personal concept hierarchy preference can be characterised by a vector of dimension $|C|$, in which each entry represents the preference of the active user u_a in a concept c_i in the concept hierarchy H. The profile of an active user u_a can be defined as vector $\overrightarrow{u_a}$ of weights for each concept. The process of generating the active user concept hierarchy is provided in more detail in Section 3.5.2.

3.5.2 Representations of the Active User Concept Hierarchy

Directly acquiring a user's preferences with the proposed structure helps new users by enabling the system to learn their needs. When a system allows users to explicitly declare which concepts (rather than items) they are interested in, a concept hierarchy can be immediately built for that user. Users provide their preferred concepts vertical, or level by level in the hierarchy. At each horizontal level of the hierarchy, the order of the preferred concepts is listed. This model provides a structure for new users to approximate their needs in a two-dimensional hierarchy. The objective of this model is to solve a new user cold-start problem and reduce the problem of users being confronted with uncertain in their information needs. This section describes the steps to measure the importance of concepts in the concept hierarchy for an active user. The steps can be divided into the four steps detailed below.

Step 1: Defining the vertical importance of levels in the hierarchy

We first evaluate the vertical importance of levels in the hierarchy to define the concept weight c_v in each level L_v of the hierarchy H for each active user u_a. Typically, the vertical relationships between concepts in the hierarchy are an 'is-a' relationships. The basic idea of a user's interest hierarchy is taken into consideration to measure the concept weight of an active user's (or new user) u_a vertical reference level for each concept c_v in the hierarchy. The design obviously indicates that the concepts in the top levels are more general and those in the bottom levels are more specific. Therefore, the concepts $c_v \in C$ in leaf nodes or in the lower level of the hierarchy should have a higher weight value than those in the top level or the root of the hierarchy, as shown in Figure 3.10.

For example, someone showing interest in children books about Iron Man probably has a stronger interest in the specific category of superhero books than in the broader category of children books. For this user's preference, the superhero books concept carries greater weight than the children books concept.

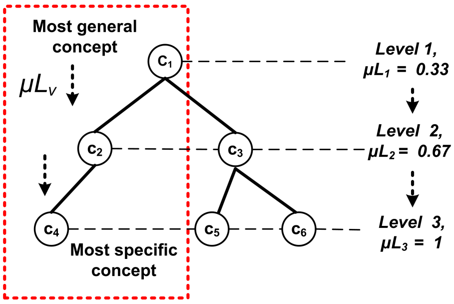

Figure 3.10: Defining a concept's vertical weight in the hierarchy

Let H = $\{L_1, L_2, \ldots, L_p\}$ the concept hierarchy and μ_{L_v} be the vertical importance of level v in the hierarchy. We can define the concept weight in each level L_v of the hierarchy for the active user u_a according to the values of μ_{L_v}. We use the following constraints to decide the values of μ_{L_v},

$$\mu_{L_p} = 1;$$

$$\mu_{L_v} > \mu_{L_{v-1}}; \quad and$$

$$\mu_{L_v} = (v \times \mu_{L_p})/p$$

For all $v = 1, 2, \ldots, p$, let p be the maximum level of the hierarchy H_{u_a}, and μ_{L_p} has the highest value for the concept levels.

For example, suppose a user has browsed two books in *Computers&Technology*, as illustrated in Figure 3.3. The user may be more interested in set of taxonomic concepts related to *Java Programming* than in the set of taxonomic concepts for *Microsoft SQL Server*

2012. Assuming that the active user u_a gives his or her vertical concept preference in the hierarchy as $Computers\&Technology > Web\&Design > Programming \quad Languages$, respectively, the vertical level of the hierarchy for the given user u_a can be defined as $H_{u_a} = \{L_1, L_2, L_3\}$. Based on the above constraints, the vertical importance of levels and concepts for the user u_a can be defined as the vertical importance factor and the weight of the concept *Programming Languages* at level v = 3 is $\mu_{L_3} = 1$.

The vertical importance factor and the weight of the concept *Web&Design* in level v = 2 and $p = 3$ is:

$$\mu_{L_2} = (2 \times \mu_{L_3})/3 = 0.67$$

The vertical importance factor and the weight of the concept *Computers&Technology* in level v = 1 is:

$$\mu_{L_2} = (1 \times \mu_{L_3})/3 = 0.33$$

Step 2: Defining the horizontal importance of concepts in hierarchy

The user's horizontal preferences in the concepts indicate his or her list of concept priorities. The order of the concepts in the list is provided by the user. We evaluate the horizontal importance of each concept to define the concept weight c_h in each level L_v at position h of the hierarchy H for each active user u_a. In the same horizontal level L_v of the hierarchy H, the concepts c_h at a level L_v are sorted from left to right according to their importance. The first concept (whose index number is 0) in the sequence is the most important, and the concept on the left side are more important than the concepts on the right side. The λ_{c_h} represents the horizontal importance of concept c_h, which is calculated as follows:

$$\lambda_{c_h} = \frac{1}{2^h}, \tag{3.4}$$

where h is the index number of concept c_h in the horizontal concept sequence in level L_v, $h \geq 0$.

For example, suppose a user browsed two books in *Computers&Technology* concept (or category), he or she might be more interested in concepts of *Microsoft SQL Server 2012* than in concepts relating to *Java Programming*, as illustrated in Figure 3.3. The user might be interested in the *Microsoft* concept first, then *Programming* and *Web&Design*, according to his or her respective concept preferences. Therefore, the importance of the *Microsoft* concept is higher than *Programming* and *Web&Design*. Based on the above constraints, the weight of the concept *Microsoft* can be defined as λ_{c_1} = 1, *Programming* can be defined as λ_{c_2} = 0.5 and *Web&Design* can be defined as λ_{c_3} = 0.25.

Step 3: Defining the vertical and horizontal importance of concepts in the hierarchy for the active user

To generate the personal preferences of the active user in the concept hierarchy, the importance value of each concept in L_v and at the position h can be calculated by incorporating the concept's vertical weight μ_{L_v} with the concept's horizontal weight λ_{c_h} as follows:

$$cw_{H_{u_a}}(v, h) = \mu_{L_v} \times \lambda_{c_h}, \tag{3.5}$$

where $cw_{H_{u_a}}(v, h)$ is the concept hierarchy weight c_h in level v on the order h.

Step 4: Transforming an active user's concept hierarchy preferences into a profile vector of concepts for the user

Based on the above discussion, both an active user's preferences and a new user's preferences can be described in a concept hierarchy. To measure the similarity between two users or between an active user and an item based on the two profile vectors, we can represent the corresponding concept hierarchy H as a profile vector of concepts for user u_a. To transfer the active user u_a's concept hierarchy preferences into the active user's personal concepts profile vector, we further normalise the concept hierarchy weight by using the min-max normalisation technique. The following equation is used to normalise the weight of the concept at position h in level L_v for the active user profile vector.

$$cw'_{H_{u_a}}(v,h) = \frac{cw_{H_{u_a}}(v,h) - min_{(x,y)\in H_{u_a}}(cw_{H_{u_a}}(x,y))}{max_{(x,y)\in H_{u_a}}(cw_{H_{u_a}}(x,y)) - min_{(x,y)\in H_{u_a}}(cw_{H_{u_a}}(x,y))}, \qquad (3.6)$$

where $cw'_{H_{u_a}}(v,h)$ is the normalised weight of concept c_i in level v at the order h on hierarchy H for the active user u_a. $cw_{H_{u_a}}(v,h)$ is the active user u_a's concept preferences weight in level v at the order h of hierarchy H, which is called the user's concept hierarchy weight. If $min_{(x,y)\in H_{u_a}}(cw_{H_{u_a}}(x,y))$ is the minimum concept weight value in user u_a's concept hierarchy H_{u_a}, $max_{(x,y)\in H_{u_a}}(cw_{H_{u_a}}(x,y))$ is the maximum concept weight value in user u_a's concept hierarchy H_{u_a} .

Finally, each active user's concept hierarchy preferences can be characterised by a vector of dimension $|C|$. A profile vector of concepts for active user u_a is represented by $\overrightarrow{u_a} = (cw'_{H_{u_a}}(1,1), ..., cw'_{H_{u_a}}(1,2), ..., cw'_{H_{u_a}}(p,1), \cdots)$. Each entry $cw'_{H_{u_a}}(i,j)$ in $\overrightarrow{u_a}$ represents the weight of a user u_a's preference for the concept c_i in level i and at the order j on the concept hierarchy H_{u_a}. Therefore, the similarity between two users or between users and items based on two profile vectors can then be identified by using the cosine similarity measure.

3.5.3 Acquiring User Preferences Using the Concept Hierarchy Algorithm

Algorithm 1 describes the procedure for acquiring user preferences from a few rated items, where the input D is the collection of all descriptors in the rated items. Step 1 initialises the levels and the number of levels, step 2 gets concepts for each level and step 4 sorts the concepts on the same level.

Algorithm 1: Acquiring user preferences using concept hierarchy

---

Input : D, a set of item taxonomic descriptors (vectors)
Output: H, the concept hierarchy of D

1 //initialisation;
let $p = max\{|d|, d \in D\} - 1$
for *i=1 to p* **do**
  let $L_i = \emptyset$
end
2 **for** *i=1 to p* **do**
  for *each* $d \in D$ **do**
    // get i-th concept of d
    if $|d| > i$ **then**
      let $L_i = L_i \cup \{d_i\}$
    end
  end
end
3 let $H = \{L_1, L_2, ..., L_p\}$
4 **for** *each* $L_i \in H$ **do**
  for *each* $c \in L_i$ **do**
    calculate its frequency in D
  end
  Sort L_i in descending order based on these frequencies
end

3.6 THE ADVANTAGES OF THE PROPOSED TAXONOMY-BASED APPROACH

To address recommendation accuracy issues related to the cold-start problem, a taxonomy-based approach is proposed. The main idea is to utilise item taxonomic information to share the relationships between users and items according to a set of taxonomic concepts (or categories) with which the user interacts with and which the item has. In this section, we explain the benefits of the proposed taxonomy-based approach, which can contribute significantly to alleviating these problems and improving the effectiveness of recommendations.

- This thesis focuses on item taxonomy to formalise the aforementioned problems. In the case of having limited ratings data in the system or if the user's personal information is difficult to obtain, item taxonomic information can be used to reflect a user's content preferences or interest in items, e.g., a user's taxonomic topic preferences. The content of an item can be represented by taxonomic categories, keywords or topics listed in different ways. The main advantages of using item taxonomy include the use of standard vocabularies and the implied hierarchical relationship between topics. In addition, a taxonomy in a hierarchical structure can also reflect user preferences in terms of topics at different levels, e.g., from a general topic (a top node) to more specific topics. Therefore, user preferences for items can be predicted by using a set of taxonomic concepts (or categories) associated with an item.

- This thesis proposes a taxonomy-based approach, namely, a concept hierarchy model, specifically designed to make recommendations to remedy the cold-start problem. The proposed taxonomy based on the concept hierarchy model approach focuses on the importance of topics rather than item ratings to obtain a new user's preferences or information needs. This method allows new users to transmit their preferences in a concept hierarchy rather than by rating items. Based on the taxonomy in a hierarchical structure, we learn about user preferences through the concept taxonomy of items in both directions of the hierarchy. The proposed taxonomy-based concept hierarchy model provides a better structure for new users to approximate their needs based on item descriptions, thereby gaining a more comprehensive user concept preference profile. Likewise, when a new item is added to the system, and therefore lacks item ratings, the hierarchical relationship between the item's taxonomic concepts can be initially used to guide recommendations for the new item. This method can be applied at the initialisation of a user profile to represent the underlying items' categories when there is insufficient ratings data in the system. Both user and item profiles can be represented as concept hierarchies in the proposed approach. These two user profiles can then be used to identify the similarities between users and items when there is a cold-start problem.

- The cold-start problem makes neighbourhood formation inaccurate, thereby resulting in poor recommendations. Basically, the popular collaborative filtering (CF) algorithms make predictions or recommendations based on the k-Nearest Neighbours (kNN) techniques. The kNN techniques focus on the similarity of ratings between users (user-based) or items (item-based). The prediction accuracy of kNN methods can be reduced when there is insufficient ratings data in the system or when faced with the cold-start problem. The proposed taxonomy-based concept hierarchy approach can provide another way of finding users or items with similar concept hierarchies for a given user, even when there is no ratings data in the system. As a result, the neighbourhood forming can be improved and becomes more accurate, thereby resulting in a higher quality of recommendations.

- Explicit acquisition is employed to obtain user preferences. Directly acquiring a user's preferences with the proposed structure helps new users by enabling the system to learn their needs. When a system allows users to explicitly declare which concepts (rather than items) they are interested in, a concept hierarchy can be immediately built for that user. Users vertically provide their preferred concept in the hierarchy. At each horizontal level, they list the order of their preferred concepts. It is convenient for users to describe what they want in the concept hierarchy by sorting sub-concepts based on their preferences without using any numerical values or weights. We also provide the explicit method to help the systems to do the same thing that human beings do-make assumptions and extrapolate a person's interests and abilities from limited information. Some information is more useful than others.

3.7 CHAPTER SUMMARY

The chapter focused on using item taxonomic information as domain-specific knowledge and exploiting the relationship between users and items according to the taxonomic categories utilised by users. Instead of using only ratings data, user preferences can be characterised by a set of taxonomic concepts. We proposed two approaches that used item taxonomic information to capture user preferences and item representations.

This chapter introduced the proposed concept hierarchy model for representing user preferences (or needs) and item descriptions in a two-dimensional hierarchy. User preferences are represented by the user concept hierarchy, and items are represented by the item concept hierarchy. This includes a novel way of exploiting users' explicit feedback to solve the cold-start problem and helping users who are confronted with uncertain regarding their information needs. This method also allows for accurate user profiling and increases the quality and speed of recommendations. User profiles and item representations based on the concept hierarchy model are used to make better personalised recommendation approaches in Chapter4.

Chapter 4

MAKING PERSONALISED ITEM RECOMMENDATIONS

One aim of this thesis is to improve the performance of personalised item recommendations. The recommendation algorithm performance evaluation explained herein determines whether a system accurately predicts demand and makes item recommendations relevant to user interests. The effectiveness of a recommendation depends on the accuracy of the users' profile and the system's ability to acquire data on the user information needs. Chapter 3 discussed using item taxonomies to obtain user preferences, generally represented in the form of user profiles. In this chapter will discussed how to utilise the two different user profiles and item representations, based on the concept hierarchy model, to make better personalised item recommendations.

This chapter focuses on developing new recommendation approaches to alleviate the recommendation problems mentioned in Chapter 1 as well as to improve the quality of item recommendations. This chapter presents two recommendation approaches that make use of item taxonomic information. The first approach, called the item popularity and concept hierarchy (PopCs), utilises item taxonomic information and item popularity to improve the recommendation quality of the standard collaborative filtering (CF) systems and alleviate the cold-start problem. The second approach, called the concept taxonomy with language model (CTLM), is designed based on the underlying language model (LM) to determine how items are likely to generate what the user wants and alleviate the cold-start problem.

4.1 PROBLEM DEFINITION

The purpose of item recommendations is to create a personalised ranking of items that a user is most likely to buy [92]. The most widely used technique to create recommendations and predictions is CF. With the traditional method of creating recommendations for the target user, the recommender retrieves items based on the past purchase information of other users who behave similarly, known as *nearest neighbours*. The other users' ratings data are then used to predict missing ratings or to create a Top-N recommendation list for target users, i.e. new users. User ratings data is primary information for the prediction tasks of a CF system. When there are insufficient ratings data or in the case of a cold-start problem, the formation of the neighbourhood becomes inaccurate, poor-quality recommendations result.

Obviously, most recommender systems cannot make recommendations without sufficient prior information about the user. The difficulty in making personalised item recommendations lies in the fact that it is nearly impossible to recommend items when information about the user is limited. Many approaches have been proposed to overcome the cold-start problem, such as using implicit ratings or hybrid techniques. The hybrid filtering approach combines content-based filtering and CF techniques to improve recommendation quality.

To help solve this problem, this thesis proposes that additional information from the items' taxonomies be used as part of a new approach to personalise recommendations. Because the problem with item recommendations is the search for items, the question is how to use item taxonomic information to develop and implement effective approaches to determine which items match or are consistent with user preferences.

4.2 THE PROPOSED RECOMMENDER SYSTEM FRAMEWORK

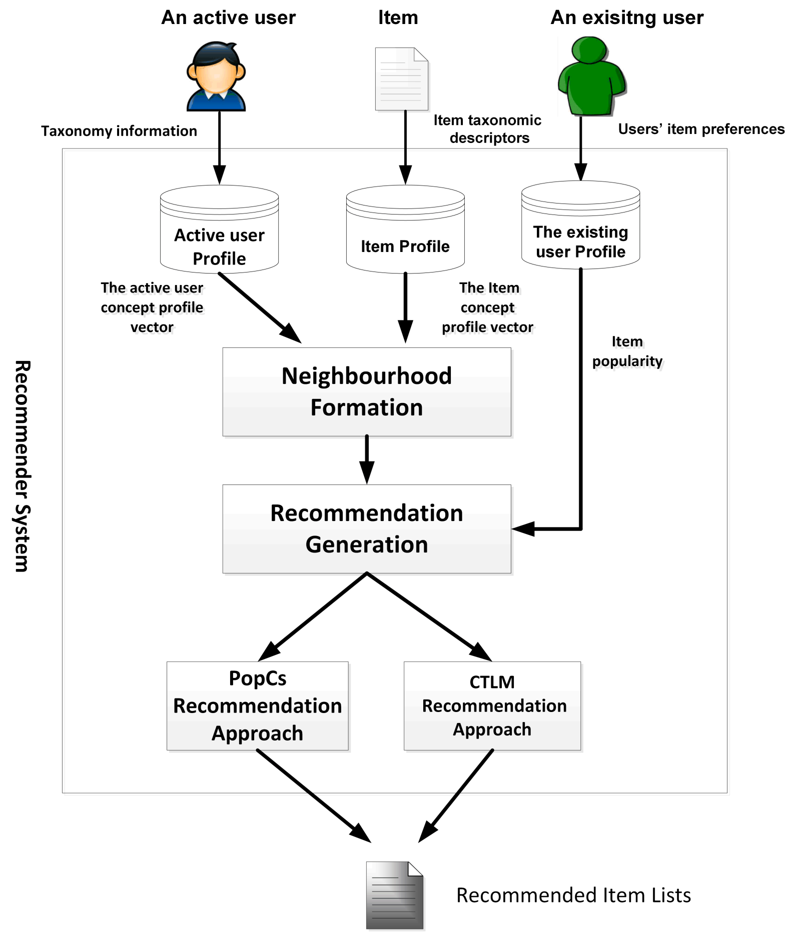

Figure 4.1: The general framework for the proposed recommender system

Figure 4.1 shows a general framework of the proposed recommender system, assuming the necessary taxonomic information is available. The input is taxonomic information (e.g., a list of concepts [or product categories]) from active users and items. The output is a list of recommended items for each active user.

The personalised recommendation process comprises three main steps. The first is to profile user preferences and represent the relevant concepts of each item. As discussed in Chapter 3, the preferences of each active user u_a and the representation of each item b_k profiled with the concept hierarchy. The concept hierarchy preferences of each user and item representation can be represented by the set of concept taxonomy (or categories) $C = \{c_1, c_2, ..., c_i\}$ in a two-dimensional hierarchy. Thus, each user's preferences are characterised by a vector of dimension $|C|$, in which each entry represents the user's preferences in a concept in the two-dimensional hierarchy. Based on concept hierarchy, the representation of item b_k is characterised by a vector of dimension $|C|$ in which each entry represents the relationship of the item to the set of taxonomic concepts in the two-dimensional hierarchy.

The users' concept preference profile vectors and the item's concept profile vector are exploited to compute the similarities of concept preferences between a user and an item based on cosine similarity. This step is to form the top *k* Nearest-Neighbours of like users and items. Finally, based on each recommendation-making approach, a given Top-N item with high prediction scores in the set of similar users and items will be recommended to each active user u_a. This step generates the item recommendations.

The following sections of this chapter discuss how to utilise the two previously proposed a user profile and item representation based on the concept hierarchy to form a neighbourhood of similar users and items. It will also explain how to generate personalised recommendation lists according to the three recommendation approaches detailed.

4.3 NEIGHBOURHOOD FORMATION

Neighbourhood formation is to generate a subset of most similar users $u_i \in U$ or items $b_k \in B$ for an active user u_a. It is the most important process in making recommendations. In this thesis, *k-Nearest-Neighbours* or kNN techniques are adopted to find neighbourhoods for users and items. The kNN formation process is used to select the top *k* neighbour users or items with the shortest distance to the active user u_a by computing the distances between the active user u_a and all other users or all other items b_k [42].

The basic idea of neighbourhood formation is to reduce the size of the neighbourhood for the active user. If every user or item is included in the neighbourhood, it negatively affects the system's performance by slowing its calculation time and the quality of its predictions [48]. The techniques for reducing the size of the neighbourhood can be divided into two techniques: defining a specific minimum threshold of user similarity and limiting neighbourhood size by defining a fixed-size neighbourhood of most similar users or items with respect to the active user and taking only the k nearest neighbours into account (or best-n-neighbours) [48].

However, the problem of finding a good number of neighbours k to use still exists. When using small neighbourhoods (less than 20), the accuracy of the system may be negatively affected. When using large neighbourhoods, it brings additional noise into the prediction. An analysis of the MovieLens dataset found that in most real-world situations, a neighbourhood of 20 to 50 neighbours is reasonable [42, 48]. In this research, we use the technique by fixing a neighbourhood size for a given user. Based on our experimental results, we found that the size of the neighbourhood at *k* = 50 provides the better results than $k = 20$ and $k = 30$. For this reason, only *k* = 50 nearest similar users or items were taken into consideration when forming the neighbourhood for each active user u_a.

In CF recommender systems, the two *k*-Nearest- Neighbours approaches (those for users and items) have proven to be top performers in CF recommender systems. The user-based *k*-Nearest-Neighbours approach is used to select the *k*-neighbour users most similar to the active user u_a by calculating the degree of similarity between the active user u_a and other users u_i. The item-based *k*-Nearest-Neighbours approach is to select the top-*k* neighbour items most similar to items through calculating the distance between item b_k and all other items. Thus, to determine an item neighbourhood for active user u_a, the similarities between two objects must be measured. This distance or similarity can be estimated using various kinds of proximity measures such as cosine similarity, the Pearson's correlation coefficient, Spearman's rank-correlation coefficient, and Jacccard's coefficient [30].

However, the insufficiency of ratings data is one important cause of the cold-start problem. It makes neighbourhood formation inaccurate, and consequently, the quality of recommendations is degraded. To tackle the problems, this thesis selects the *k*-neighbour items most similar to active user u_a by using the content matching method rather than using item ratings methods. As discussed in Chapter 3, each active user u_a's concepts hierarchy preferences are encoded by a $|C|$-sized concept hierarchy vector, which is represented by an active user profile vector $\overrightarrow{u_a}$. Each item concept hierarchy is modelled by a $|C|$-sized concept hierarchy vector, which is represented by an item profile vector $\overrightarrow{b_k}$. When user and item information is encoded in the profile vectors of concepts for users and items, the cosine similarity measure is the appropriate function to measure the interestingness of item b_k to active user u_a [88].

The *cosine similarity* measurement is popular and very efficient when used in information retrieval (IR) and text mining to capture the similarity of two documents, in which documents are represented as vectors of terms [44, 48, 102]. One example of its use is to determine term frequency, i.e. how often specific keywords or topics appear in a document. In this similarity metric, the attributes are used as a vector to find the normalised dot product of two documents. One representation method often used is the *vector-space model*. In the vector space model, user and item profiles are treated as documents by one or more profile vectors. Its metric is a measurement of the similarity between two *N*-dimensional vectors based on the angle between them. For cosine similarity resulting in a value of 0, the two documents are likely to not contain many of the same words. Formally, the degree of similarity between two profile vectors, $\overrightarrow{v_i}$ and $\overrightarrow{v_j}$, is defined as follows:

$$sim(\overrightarrow{v_i}, \overrightarrow{v_j}) = \frac{\overrightarrow{v_i} \cdot \overrightarrow{v_j}}{\|\overrightarrow{v_i}\| * \|\overrightarrow{v_j}\|} \tag{4.1}$$

Here, '**.**" is the dot-product of the two vectors. $\|\overrightarrow{v_j}\|$ is the Euclidean length of the vector, which is defined as the square root of the dot product of the vector with itself. Thus, neighbourhood formation is created by computing proximity weights $sim(\overrightarrow{v_i}, \overrightarrow{v_j})$.

The following subsections discuss how to match an active user u_a and an item b_k using the profile vectors of concept for $\overrightarrow{u_a}$ and $\overrightarrow{b_k}$.

4.3.1 Matching User-Item Concept Hierarchy Similarities

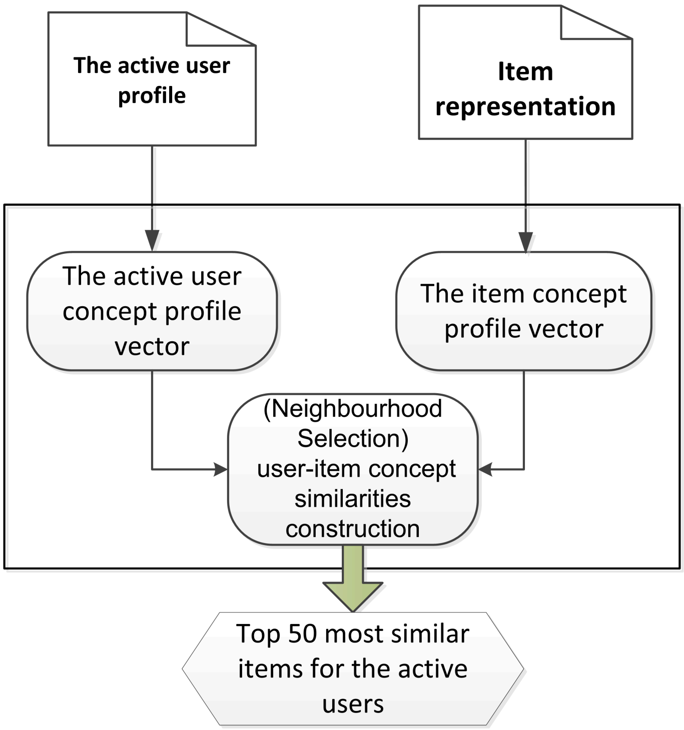

Figure 4.2: A framework for determining user-item concept similarities

Figure 4.2 shows the framework for determining user-item concept hierarchy similarities, where the inputs are the active user profile vector $\overrightarrow{u_a}$ and the item profile vector $\overrightarrow{b_k}$ and the outputs are the set of items most similar to the active user u_a. An active user u_a can provide his or her preferred concepts in a two-dimensional hierarchy H_{u_a} by using concept taxonomy. Each the active user profile contains the user's personal concept hierarchy preference H_{u_a}, which is modelled as a vector $\overrightarrow{u_a}$.

For an item b_k, its item descriptors $d_x \in D_{b_k}$ are used to create the item concept hierarchy H_{b_k}. Each item representation contains the item concept hierarchy H_{b_k}, which is defined as a vector $\overrightarrow{b_k}$. The similarities between the active user and items are measured based on these two profile vectors $(\overrightarrow{u_a}, \overrightarrow{b_k})$:

$$\overrightarrow{u_a} = (cw'_{H_{u_a}}(1,1), \cdots, cw'_{H_{u_a}}(2,1), \cdots, cw'_{H_{u_a}}(p,1), \cdots)$$

$$\overrightarrow{b_k} = (cw'_{H_{b_k}}(1,1), \cdots, cw'_{H_{b_k}}(2,1), \cdots, cw'_{H_{b_k}}(p,1), \cdots)$$

Here, each entry $cw'_{H_{u_a}}(i,j)$ in $\overrightarrow{u_a}$ represents the weight of a user u_a's preference for the concept c_i in level i and at the order j on concept hierarchy H_{u_a}. Each entry $cw'_{H_{b_k}}(i,j)$ in $\overrightarrow{b_k}$ represents the weight of each concept relevant to item b_k in the concept hierarchy H_{b_k}.

Additionally, to find the most k similar item neighbours for each active user u_a, a similarity measure must be defined. The cosine similarity is adopted to calculate the similar concept hierarchies-based content matching between the two profiles vectors of an active user u_a and item b_k. The neighbourhood for each active user u_a is identified by computing the proximity weight $sim(\overrightarrow{u_a}, \overrightarrow{b_k})$. Formally, if *W* is the *m* × *n* × *z* user-concept-level matrix, then the similarity between the active user u_a and item b_k is defined as the cosine of the N-dimensional vectors corresponding to the u_a^{th} and b_k^{th} columns of matrix *W*. Thus, for user u_a and item b_k with profile vectors $\overrightarrow{u_a}$ and $\overrightarrow{b_k}$, respectively. The vector-cosine similarity is defined as

$$sim(\overrightarrow{u_a}, \overrightarrow{b_k}) = \frac{\overrightarrow{u_a} \cdot \overrightarrow{b_k}}{\|\overrightarrow{u_a}\| \|\overrightarrow{b_k}\|}, \tag{4.2}$$

where $\overrightarrow{u_a}$ denotes the active user profile vector, and $\overrightarrow{b_k}$ denotes the item profile vector, while $\|\overrightarrow{u_a}\|$ is defined as the square root of the user u_a's concepts for the vector times with itself. Likewise, $\|\overrightarrow{b_k}\|$ is defined as the square root of the item b_k's concepts for the vector times with itself.

Therefore, the most similar items $b_k \in B$ can be selected for the active user u_a by computing proximity weights $sim(\overrightarrow{u_a}, \overrightarrow{b_k})$. User-item concept hierarchy similarities are applied when making item recommendations in the proposed *PopCs* and *CTLM* recommendation functions.

4.4 RECOMMENDATION GENERATION

A *Top-N recommendation* is a recommended set of items that are of interest to an active user. Traditionally, Top-N techniques use a user-item matrix to determine the relationships between users and items and, then compute recommendations [28]. The most popular model for recommender systems is K-nearest neighbour (kNN) collaborative filtering. Its algorithms make predictions for active users based on the preferences of like-minded peers with user-item-based CF. That being said, there are many item-recommendation techniques, including *matrix factorization* and *singular-value decomposition* (SVD) [23], which have been proposed for both implicit and explicit feedback in recommender systems. Nevertheless, the usual approach to making item recommendations is to generate a personalised score for an item based on a user's preferences. Then, the items are sorted, ranked, and recommended to the user according to this score [91].

The algorithms proposed herein are in a recommendation-list format such that the system produces a list of items matching the expected preferences of the user for the queried item. This chapter proposes three item recommendation approaches that combine collaborative and content-based filtering approaches to improve the recommendation quality of standard CF systems and address recommendation-making problems (i.e. the cold-start problem). In Section 4.4.1, item popularity is incorporated with user-item concept hierarchy similarities to construct the *item popularity and concept hierarchy* (PopCs) recommendation function. Section 4.4.2 discusses the proposed *concept taxonomy with language model recommendation* (CTLM) function that is designed based on the underlying language model (LM) and in combination with item popularity, user-item concept hierarchy similarities and concept probability.

4.4.1 The Item Popularity and Concept Hierarchy Recommendation Approach

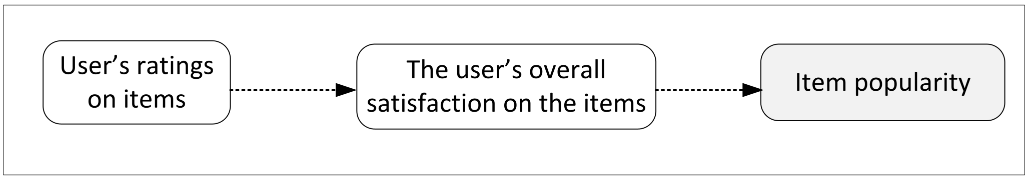

Figure 4.3: An item's popularity represents multiple users' satisfaction, based on ratings

The proposed PopCs recommendation approach has to fundamental components: (1) user ratings of items, which determine item popularity; and (2) the correlation between a user and an item based on concept hierarchies[81]. A new user problem or insufficient ratings data makes the formation of neighbourhoods in traditional CF systems inaccurate, thereby resulting in poor recommendations. This thesis formalises the problem by exploiting the relationship between users and items to obtain user preferences according to a set of taxonomic categories of the items that users interact with, represented by useritem concept hierarchy similarities, to address cold-start problem. To improve the performance of recommendations, we also utilise a user's item preferences to predict the user's opinion about an item from his or her past explicit ratings, which is represented by item popularity. The proposed PopCs generates item recommendations by integrating useritem concept hierarchy similarities with item popularity.

The following subsection discusses item popularity and the PopCs recommender algorithm in detail. The similarities between user and item concept hierarchy measurements were discussed in Section 4.3.1.

4.4.1.1 Item popularity generation

The present chapter introduces item popularity as a representation of other users' opinions about an item, as determined by ratings. Because users' ratings of items correspond to their preferences for items, they may reveal some aspect of the items that are relevant to other users; such ratings certainly speak, at least to some extent, to item quality. An item frequently viewed, with ratings that are frequently seen, is usually a popular, well-appreciated item [53]. Presenting explicit ratings data has long been effectively used to indicate user preferences. Users express their preferences to items in numeric form. $R_{ik} = 0$ indicates user u_i dislike of or disinterest in rating item b_k. Therefore, $R_{ik} > 0$ indicates that user u_i is satisfied to some degree with the item, such that 1 is the smallest positive-support value and 10 is the largest. To track item popularity, one must build an interface that extracts explicit ratings R_{ik} that users u_i give to the item b_k and must also build a means of utilising this data to generate item popularity as follows:

$$pop(b_k) = \sum_{u_i \in U} R_{ik}, \tag{4.3}$$

where $1 \leq R_{ik} \leq 10$, and 1 is the smallest positive support value and 10 being the largest positive support value. One should also normalise item popularity further using min-max normalisation as follows:

$$npop(b_k) = \frac{pop(b_k) - min_{pop}}{max_{pop} - min_{pop}}, \tag{4.4}$$

where $0 \leq npop(b_k) \leq 1$ is the normalised item popularity, $min_{pop} = min\{pop(b_k)|b_k \in B\}$, and $max_{pop} = max\{pop(b_k)|b_k \in B\}$

4.4.1.2 The PopCs recommender algorithm

The recommendation of an item to an active user u_a is determined by the prediction score computed using Equation 4.6. To recommend a set of N items to the active user u_a, we firstly form a group of items similar to the active user based on the similarity of concept hierarchies. As discussed in Chapter 3, concept correlations within item taxonomies make up the two-dimensional hierarchy structure, which can be used to describe the relevance of an item to a user, as well as to convey relevant concepts about that item. The active user's concept hierarchy preference can be represented by the user profile vector $\overrightarrow{u_a}$, and the relevance of taxonomic concepts to the item in a two-dimensional hierarchy can be represented by the item profile vector $\overrightarrow{b_k}$.

Cosine similarity is adopted to calculate the content match between an active user u_a and item b_k. The *k*-neighbour items b_k most similar to each active user u_a can be identified by computing the proximity weight $sim(\overrightarrow{u_a}, \overrightarrow{b_k})$. The degree of concept similarity between user u_a and item b_k can be defined as $cs(u_a, b_k)$:

$$cs(u_a, b_k) = sim(\overrightarrow{u_a}, \overrightarrow{b_k}) \tag{4.5}$$

For each active user u_a, the item neighbourhood of the active user u_a is denoted by $Neighbour(u_a) = \{b_k | b_k \in topK\{cs(u_a, b_k)\}\}$, where function $topK\{\}$ returns the top k most similar items to u_a. For each active user u_a, a set of items b_k are recommended according to the prediction score $p\_score(u_a, b_k)$. The prediction score is used to estimate how much interest the active user u_a will have in item b_k. In essence, then, the proposed PopCs approach is generated by linearly combining user-item concept similarities $cs(u_a, b_k)$ and item popularity $npop(b_k)$. The Top-N items with high prediction scores will be recommended to the active user u_a. Formally, the prediction score for each active user, denoted by $p\_score(u_a, b_k)$ can be calculated using the expression below:

$$p\_score(u_a, b_k) = \alpha \times npop(b_k) + (1 - \alpha) \times cs(u_a, b_k) \tag{4.6}$$

Here, parameter α is an experimental coefficient corresponding to item popularity $npop(b_k)$ and $cs(u_a, b_k)$, $0 \geq \alpha \leq 1$, which offers the ability to achieve the best performance in terms of precision and recall.

According to the algorithm 2 (in step 3), 0.5 is added to both the numerator and denominator to ensure that the denominator does not equal 0. In step 8, prediction score p_score is calculated for each item b_k to make item recommendations. The full PopCs recommender algorithm is shown below:

Table 4.1: *PopCs* Recommender Algorithm

Algorithm 2: *PopCs* recommender algorithm.

Inputs : a set of existing users $u_i \in U$, a set of items $b_k \in B$ and each item's concept hierarchy H_{b_k}, new user or active user u_a and each active user concept hierarchy H_{u_a}, the explicit ratings R_{ik} generated by users $u_i \in U$ to items b_k and parameters α.

Output: a list of N items recommended for u_a

for *each item* $b_k \in B$ **do**

// Calculate item popularity

$$pop(b_k) = \sum_{u_i \in U} R_{ik}$$

end

let $min_{pop} = min\{pop(b_k)|b_k \in B\}, max_{pop} = max\{pop(b_k)|b_k \in B\}$

for *each item* $b_k \in B$ **do**

// Normalise item popularity

$$npop(b_k) = \frac{pop(b_k) - min_{pop} + 0.5}{max_{pop} - min_{pop} + 0.5}$$

end

let $\overrightarrow{u_a}$ be the concept profile vector of u_a

let $\overrightarrow{b_k}$ be the concept profile vector of b_k

for *each item* $b_k \in B$ **do**

// Calculate the user-item concept hierarchy similarities to get the K nearest neighbours of the active user u_a

$$cs(u_a, b_k) = \frac{\overrightarrow{u_a} \cdot \overrightarrow{b_k}}{\|\overrightarrow{u_a}\| \times \|\overrightarrow{b_k}\|}$$

end

// Get the k nearest neighbour items of each active u_a with the highest similarity score.

$Neighbour(u_a) \leftarrow \{b_k|b_k \in topK\{cs(u_a, b_k)\}\}$

for *each item* $b_k \in Neighbour(u_a)$ **do**

//Recommendation generation.

//Generating a list of N items and order the items by the prediction score.

$$p\_score(u_a, b_k) = \alpha \times npop(b_k) + (1 - \alpha) \times cs(u_a, b_k)$$

end

the active user $u_a \leftarrow$ Select Top-N items with the highest scores of $p\_score(u_a, b_k)$

4.4.2 The CTLM Recommendation Approach

This thesis proposes a novel inferential language model (ILM) that combines both semantic and statistical inference to estimate the probability that a user is interested in an item for recommendation. Statistical language models (LM) were originally introduced in Ponte and Croft [90]. Basically, an LM is used in information retrieval (IR) as part of the query-likelihood model. The LM approach to IR provides different approaches to document ranking [72]. LM works by estimating a document's relevance based on its textual content with respect to query q, and documents are ranked based on the probability of the model generating the query $P(q|M_d)$.

In other words, LM intuits that the user has a document in mind, and it generates a query from words that appear in this sought-for document. Another way of using LM in IR is with recommendations, as is the case when finding user preferences for topics within an item's profile. In the case of recommendations, one might assume that the user has item preferences in mind and has generated his or her query using topics or categories that are part of the item's inherent description. LM, then, relates to three main parameters: words t, documents d and queries q. It can apply to the recommendations where item b_k represents the document, user u_a represents queries and words t is represented by a set of concept or categories c_i. This is another way of adapting the LM idea in IR to make recommendations.

CTLM is designed and developed based on the relevance of an item to a user as a kind of LM to understand how items are likely to generate what the user wants. The following two main components are taken into consideration when developing the proposed CTLM approach:

1. Users' opinions about items, which are represented by item popularity;

2. The relationship between a user's concept hierarchy and other items' concept hierarchies. This can be further divided into two sub-components:
   2.1. The similarity of the user to the item, based on concept hierarchies.
   2.2. The probability of concepts used in concept hierarchies.

The following subsections briefly review LM's function in IR, after which an LM adaptation for making recommendations is proposed and discussed in detail.

4.4.2.1 Adapting Language Model to Make Recommendations

As discussed in Chapter 3, users and items are represented as user-and-item-concept hierarchies, respectively. This section presents a novel way of estimating the likelihood that user u_a is interested in item b_k using a probabilistic language modelling approach. Such a likelihood is approximated by the probability that the user concepts of interest are generated by the items descriptors, i.e. $P(u_a|b_k)$. The basic assumption is that if the LM of an item can generate the concepts that characterise the users' item interests, then the user is likely to be interested in the corresponding item.

This novel inferential language model (ILM) approach also makes both semantic and statistical inferences to estimate the probability that a user u_a is interested in an item b_k. An LM is a probabilistic function that assigns a probability to a string t drawn from some vocabulary set T. Probabilistic LM has been applied to estimate the relevance of a document d with respect to a query q in terms of the likelihood of generation probability in the field of IR [68, 90]. Moreover, language modelling method has been successfully applied to opinion mining [58, 60]. In the present context, the usual query has been replaced by a descriptor of the user's interests, and a document is a set of terms describing the item.

$$P(d|q) \propto P(d) \prod_{t \in q} ((1-\lambda)P(t|M_D) + \lambda P(t|M_d)) \tag{4.7}$$

Here, $P(d) \prod_{t \in q} ((1-\lambda)P(t|M_D) + \lambda P(t|M_d))$ is proportional ($\propto$) to $P(d|q)$, and the probability of $P(d|q)$ of the relevance of document d to a given query q; M_d is a language model built for each document d; M_D is a language model built for the entire document collection. This equation combines the probability of the document with the general collection frequency of words t. To generate the prediction scores needed to make recommendations, Equation 4.7 has been modified as follows:

$$P(b_k|u_a) \propto P(b_k) \prod_{c_i \in H_{u_a}} ((1-\lambda)P(c_i|M_D) + \lambda P(c_i|M_{b_k})), \tag{4.8}$$

where H_{u_a} is the user's concept hierarchy and D is the set of all item-concept hierarchies (that might be rated by other users). From the Equation 4.8, we derive the following results:

$$P(b_k|u_a) \propto \ln\{P(b_k) \prod_{c_i \in H_{u_a}} ((1-\lambda)P(c_i|M_D) + \lambda P(c_i|M_{b_k}))\}$$

$$P(b_k|u_a) \propto \ln(P(b_k)) + \sum_{c_i \in H_{u_a}} ln((1-\lambda)P(c_i|M_D) + \lambda P(c_i|M_{b_k})) \tag{4.9}$$

Normally, a concept describing a user's interest absent in a b_k does not necessarily mean that the item is not relevant to the user's interests, because the document indexing scheme is not perfect and sometimes synonymous concepts are used in concept hierarchies. For instance, if the user's interest is described by the descriptor *data mining*, an item b_k (e.g., a book) about *knowledge discovery from databases* is very relevant even though the concept (or term set) *data mining* does not appear in the set of descriptors of the item. To reduce the effect of underestimating the probability of *unseen* concepts in an LM, various document *smoothing* methods have been proposed [68, 90]. The basic idea is to replace the zero probability of an unseen concept by a small value rather than zero. With Jelinek-Mercer smoothing [68], $P(c_i|M_{b_k})$ is updated using the following equation:

$$P(c_i|M_{b_k}) = (1-\lambda)P_{ML}(c_i|M_d) + \lambda P_{ML}(c_i|M_D) \tag{4.10}$$

$$P_{ML}(c_i|M_D) = \frac{tf(c_i, D)}{|D|}, \tag{4.11}$$

where D is the set of all item-concept hierarchies; $P_{ML}(c_i|M_D)$ is the maximum likelihood estimation of the entire item collection LM; λ is the Jelinek-Mercer smoothing parameter, which may take values in the range of [0.1, 0.7] [84, 116]; $tf(c_i, D)$ represents the occurrence frequency of c_i in the entire item collection D, i.e. all item hierarchies.

Using the item-collection model D to smooth an item language model M_{b_k} might partially solve the problem of the zero probability of an unseen user term. However, the generation probability might still be highly underestimated. For example, an item with the descriptor *knowledge discovery from databases* is actually very likely to match the interest, i.e. a high generation probability, of the user described by *data mining*. Accordingly, an ILM that accounts for both semantic and statistical term associations is proposed to address the above issue. Our inferential language is defined and updated $P(c_i|M_{b_k})$ using the following equation:

$$\begin{aligned} P(c_i|M_{b_k}) &= (1-\lambda)\Bigg((1-\gamma)P_{ML}(c_i|M_{b_k}) + \gamma P_{INF}(c_i|M_{b_k}) \Bigg) + \\ &\lambda P_{ML}(c_i|M_D) \end{aligned} \tag{4.12}$$

$$
\begin{aligned}
P_{INF}(c_i|M_{b_k}) &= \frac{\sum\limits_{c_i,c_j \in R} P(c_i|c_j)P(c_j|M_{b_k})}{|R|} \\
&= \frac{\sum\limits_{c_i,c_j \in R} P(c_j \rightarrow c_i)P(c_j|M_{b_k})}{|R|}
\end{aligned},
\tag{4.13}
$$

where $P_{INF}(c_i|M_{b_k})$ is the item inferential language model. This is an extension of the original ILM developed by Nie et al. [78], where previous ILM only considers semantic term relationships captured in WordNet. In this thesis, the rule set R contains the set of concept relations in the form of $c_j \rightarrow c_i$, e.g., *soccer* $\rightarrow$ *sport*, which might be acquired from an external source, such as WordNet. Meanwhile, statistical concept associations such as *wii* $\rightarrow$ *game*, they are dynamically discovered from the set of item descriptions via context-sensitive text mining or sequential text mining methods [59].

Based on the above discussion, it is very difficult to calculate $P(c_i|M_{b_k})$ because of the hierarchical relationships between concepts. The extended ILM provides us an indication for considering only concept associations for approximating $P(c_i|M_{b_k})$. Generally speaking, all possible associations can be described as the similarity between item b_k's concepts and user u_a's concepts. Hence, in this thesis, the user-item concept hierarchy similarities $cs(u_a, b_k)$ is used to approximate $P(c_i|M_{b_k})$. One might also simply use $P(c_i)$ to replace $P(c_i|M_D)$, where $P(c_i)$ is the probability of concept c_i in all relevant item concept hierarchies. Thus, if we let $npop(b_k) = ln(P(b_k))$ describe a given item popularity, based on the above analysis and Equation 4.9, we propose the following approximation equation for estimating $P(b_k|u_a)$:

$$
\begin{aligned}
p\_score(u_a, b_k) = \alpha \times npop(b_k) + (1-\alpha)[\beta \times cs(u_a, b_k) \\
+(1-\beta) \sum_{c \in H_{u_a} \cap H_{b_k}} P(c_i)],
\end{aligned}
\tag{4.14}
$$

where α, and β are experimental coefficients between 0 and 1.

The proposed CTLM is designed as an adaptation of the LM to account for the probability of taxonomic concepts so that relevance between users and items is utilised to enhance the efficacy of subsequent recommendation-making. As such, the proposed CTLM approach is composed of three parts: (1) item popularity $npop(b_k)$, (2) the user-item concept hierarchy similarities $cs(u_a, b_k)$ and (3) the concept probability $P(c_i)$. The details of the user-item concept hierarchy similarities and item popularity were described in Section 4.3.1 and subsection 4.4.1.1. In the following subsections, the details for each constituent part are discussed. Then, the proposed CTLM recommender algorithm is presented.

4.4.2.2 Concept Probability

The proposed the probability of concept $P(c_i)$ in item concept hierarchies H_{b_k} is relevant to the user u_a's preferences or interests. One must assume that user u_a is interested in the items b_k, as the user u_a has an affinity for the concepts c_i intrinsic to those items. $P(c_i)$ describes the appearance of c_i in all relevant item concept hierarchies H_{b_k}. It can be approximated by the term frequency of concept c_i's occurrence in items $b_k \in B$ divided by the total number of all concepts used in the entire items collection B in the training set. The probability of concept $P(c_i)$ can thus be calculated as follows:

$$P(c_i) = \frac{|\{b_k \in B | c_i \in H_{b_k}\}|}{\sum_{b_k \in B} |\{c_i | c_i \in H_{b_k}\}|} \tag{4.15}$$

Naturally, it is easy to verify that $0 \leq P(c_i) \leq 1$.

4.4.2.3 The CTLM Recommender Algorithm

This section discusses the novel item recommendation algorithm proposed above. The proposed CTLM recommendation approach is designed based on the relationships between users and items according to a set of taxonomic categories (or concepts) that users interact with and that the items are associated with. We also describe the relevance of an item to a user as a kind of LM to understand how users, items and concepts are integrated to estimate the probability that a user is interested in an item. The proposed CTLM generates item recommendations by incorporating three essential components: item popularity $npop(b_k)$, the similarities between users and items based on concept hierarchies $cs(u_a, b_k)$ and the probability of the concepts used in concept hierarchies $P(c_i)$.

To recommend a set of N items to a target user u_a, we first form the item neighbourhood for the active user u_a. Based on the proposed concept hierarchy model, both the user and item are converted into the vector profiles for concepts relevant to both the active user $\overrightarrow{u_a}$ and the item $\overrightarrow{b_k}$, which are then utilised to find a group of similar items with similar concept hierarchy to the active user u_a. The benefit of the proposed concept hierarchy model is that it provides an effective method of neighbourhood formation when there are insufficient ratings data in the system or limited information about new users.

Additionally, the cosine similarity function is again utilised to calculate the degree to which an active user u_a's and item b_k's content match based on their profile vectors; the top *k*-Nearest Neighbour items b_k for each active user u_a are identified by computing weight $sim(\overrightarrow{u_a}, \overrightarrow{b_k})$. Finally, concept similarity between user u_a and item b_k is denoted by $cs(u_a, b_k)$:

$$cs(u_a, b_k) = sim(\overrightarrow{u_a}, \overrightarrow{b_k}) \tag{4.16}$$

The k most similar items or the neighbourhood of the active user u_a is denoted by $Neighbour(u_a) = \{b_k | b_k \in topK\{cs(u_a, b_k)\}\}$. Then, the user-item concept hierarchy similarities are integrated with item popularity $npop(b_k)$ and the concept probability $P(c_i)$ under the adaptation LM to predict how much interest active user u_a will have in item b_k. Item popularity is the item commonly preferred by $Neighbour(u_a)$. The concept probability $P(c_i)$ is the probability of concept c_i in all relevant item hierarchy H_{b_k}, which is relevant to active user u_a's interests.

The recommendation of an item to the active user u_a will be determined based on the item recommendation function in Equation 4.17. Top-N item recommendation list is generated by ranking all items in descending order of the prediction score $p\_score(u_a, b_k)$. Formally, the prediction score for each active user is denoted by $p\_score(u_a, b_k)$, as calculated below:

$$\begin{aligned} p\_score(u_a, b_k) = \alpha \times npop(b_k) + (1-\alpha)[\beta \times cs(u_a, b_k) \\ +(1-\beta) \sum_{c \in H_{u_a} \cap H_{b_k}} P(c_i)], \end{aligned} \quad (4.17)$$

where parameter α is an experimental coefficient that corresponds to item popularity $npop(b_k)$, and β is an experimental coefficient that corresponds to $cs(u_a, b_k)$ and $P(c_i)$, 0 $\geq \alpha \leq$ 1; these allow Mean Average Precision (MAP) measurement to perform at its best.

This CTLM algorithm is described based on the above definitions in subsection 4.4.2.1, it is designed according to a concept hierarchy that incorporates LM principles. In step 3, 0.5 is added to both the numerator and denominator to ensure that the denominator does not equal 0. At least in step 10, the prediction score p_score is calculated for each item b_k to make item recommendations. The CLTM recommender algorithm is shown in Table 4.2.

Table 4.2: *T*he CTLM Recommender Algorithm

Algorithm 3: The *CTLM* recommender algorithm.

Inputs : a set of existing users $u_i \in U$, a set of items $b_k \in B$ and each item concept hierarchies H_{b_k}, a new user or an active user u_a with concept hierarchies H_{u_a}, the explicit ratings R_{ik} given by the existing users $u_i \in U$ to items b_k, and parameters α and β.

Output: a list of N items recommended for u_a

for *each item* $b_k \in B$ **do**
    //Calculate item popularity

$$pop(b_k) = \sum_{u_i \in U} R_{ik}$$

end

let $min_{pop} = min\{pop(b_k)|b_k \in B\}, max_{pop} = max\{pop(b_k)|b_k \in B\}$

for *each item* $b_k \in B$ **do**
    //Normalise item popularity

$$npop(b_k) = \frac{pop(b_k) - min_{pop} + 0.5}{max_{pop} - min_{pop} + 0.5}$$

end

let $\overrightarrow{u_a}$ be the concept profile vectors of u_a

let $\overrightarrow{b_k}$ be the concept profile vectors of b_k

for *each item* $b_k \in B$ **do**
    // Calculate the user-item concept hierarchy similarities to get the K nearest neighbours of the active user u_a

$$cs(u_a, b_k) = \frac{\overrightarrow{u_a} \cdot \overrightarrow{b_k}}{\|\overrightarrow{u_a}\| \times \|\overrightarrow{b_k}\|}$$

end

// Get top k nearest neighbour items of each active u_a with the highest similarity score.

$Neighbour(u_a) \leftarrow \{b_k|b_k \in topK\{(cs(u_a, b_k)\}\}$

let $C = \{c_i|c_i \in H_{b_k}, b_k \in B\}$

for *each concept* $c_i \in C$ **do**
    // Calculate the probability of concept $P(c_i)$ is for item b_k, which is relevant to active user u_a

$$P(c_i) = \frac{|\{b_k \in B|c \in H_{b_k}\}|}{\sum_{b_k \in B} |\{c_i|c_i \in H_{b_k}\}|}$$

end

for *each item* $b_k \in Neighbour(u_a)$ **do**
    //recommendation generation
    // Generating a list of N items and order the items by the prediction score.

$$p\_score(u_a, b_k) = \alpha \times npop(b_k) + (1 - \alpha)[\beta \times cs(u_a, b_k) + (1 - \beta) \times \sum_{c \in H_{u_a} \cap H_{b_k}} P(c_i)]$$

end

Return Top-N items with the highest scores of $p\_score(u_a, b_k)$ to the active user u_a.

4.5 CHAPTER SUMMARY

This chapter discussed how to apply a user profile and an item representation based on a concept hierarchies to devise several approaches to making item recommendations. Specifically, it proposed two approaches: (1) the PopCs approach, which was developed using a linear combination of item popularity and user-item concept hierarchy similarities; and (2) the CTLM approach, which was designed as a combination of item popularity, user-item concept hierarchy similarities and concept probability under an adapted LM approach. The LM was applied to the second approach to facilitate an understanding of how likely item recommendations meet user needs. The efficiency of the proposed CTLM recommender algorithm based on the Big-O concept is presented in Chapter 5. This chapter also discussed how to form the neighbourhood of similar users and similar items for an active user and utilise them to make item recommendations. The experimental results and evaluations of the two recommenders' performances are discussed in Chapter 5.

Chapter 5

EXPERIMENTS AND RESULTS

The previous chapter presented the proposed concept hierarchies and recommendation approaches. It claimed that each approach will improve the performance of recommendation accuracy and will also improve on some of the other techniques as well. This chapter will examine the evaluation of the proposed recommendation approaches, the research hypotheses, the datasets, the evaluation metrics, the baseline approaches involved in the comparative experiments, and discuss the application of the experimental results to the recommendation problems.

5.1 RESEARCH HYPOTHESES

The objective of the experiment was to show how the proposed user profiling and recommendation approaches can effectively improve the performance of recommender systems. To verify the effectiveness of the proposed approaches, the experiments were conducted and investigated in order to prove the following hypotheses:

- **Hypothesis 1:** The proposed concept hierarchy model based on taxonomy can effectively improve recommendation accuracy and solve the cold-start problem.

- **Hypothesis 2:** The Concept Taxonomy with Language Model (CTLM) recommendation approach can effectively improve recommendation accuracy.

- **Hypothesis 3:** Use of the concept taxonomy can effectively alleviate the cold-start problem.

5.2 EXPERIMENT EVALUATION

The following subsections will present the datasets, evaluation metrics, experimental setup and baseline models for the experiments.

5.2.1 Datasets

Experiments were conducted on two data collections of Amazon products: books and music. Each dataset was taken from two different applications.

1) **Dataset D1**: Book dataset. This dataset was collected from two data sources. The Book-Crossing data was collected by Cai-Nicolas Ziegler (http://www2. informatik.uni-freiburg.de/ cziegler/BX/) [121] from the Book-Crossing community with the permission of Ron Hornbaker, CTO of Humankind Systems. The Amazon-Book dataset was collected by Weng et al. [111] and Liang et al. [67]. The original data collection consisted of 82,193 users, 239,074 books and 719,471 product taxonomic descriptors. In the experiment, we selected items with explicit rating scales between 1 and 10. The final dataset consisted of 60,597 users, 130,379 books, and 411,942 product taxonomic descriptors. We used 20% of the final set as the ground truth. We selected users who had a set of rated books, and books that had a set of descriptors to describe the book categories.

2) **Dataset D2**: Music dataset. This dataset was collected by Leskovec and Krevl [62, 63] from the Amazon website and made available on the SNAP website: “Amazon

networks:Amazon product co-purchasing network metadata" (https://snap.stanford.edu/data/amazon-meta.html). The music dataset consisted of 103,144 songs, 460,260 users, and 469,100 music taxonomic descriptors. From the dataset, we selected users who had sets of rated songs. We also selected the songs that had a set of descriptors to describe their musical categories. We used 20% of the final set as the ground truth.

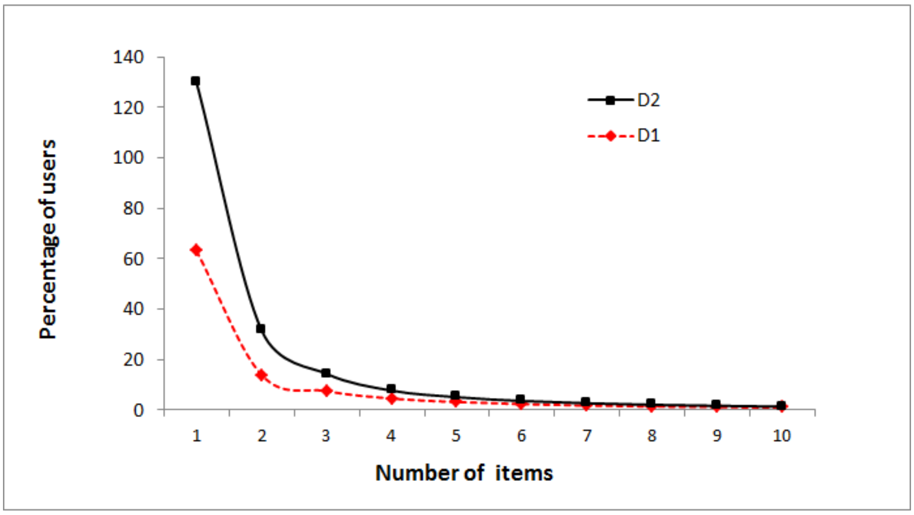

Figure 5.1: The distribution of items in Datasets D1 and D2

In order to better analyse the results of the evaluation, we studied the data's characteristics. The distribution of items in Datasets D1 and D2 is measured based on the number of items rated by users, which is shown in Figure 5.1. In Dataset D1, nearly 63% of users had rated only one book, and approximately 3% had rated at least five books. In Dataset D2, 67% of users had rated only one song, and nearly 2% had rated at least five songs. The distribution of both D1 and D2 was sparse since only a very small fraction of the users had rated a large number of items. That meant that we could only acquire a small amount information about the users. This situation affected the performance of the traditional recommender systems, including, for example, the collaborative filtering approaches that make recommendations based on nearest neighbour algorithms. In this experiment, sparse datasets were normally exploited in order to build the experimental environment use to evaluate the capability of specific approaches in the cold-start environment.

5.2.2 Baseline Models

This thesis studies the performance of the proposed recommendation approaches to understand how it can improve the accuracy of item recommendations. The four baseline approaches involved in this comparative experiment are listed below:

Baseline 1: Item Taxonomy Recommendations (ITB)

The first baseline model, item taxonomy recommendations (*ITB*), is a state-of-the-art model in this area. The approach was developed by Weng et al. [111] on the basis of Ziegler's original idea [123]. The paper that introduced this approach was also the best paper at the 10th International Conference on Enterprise Information Systems, which was held in Spain from June 13-16, 2008 [111]. The approach utilises item taxonomy in order to examine the cold-start problem in recommendation making.

This approach used the taxonomic categories of items to achieve the task of making item recommendations. It combined the users' ratings and category taxonomies into profile vectors to describe the users' taxonomic topics' preferences. The profile vectors were constructed by summarising all the taxonomy descriptors from all the rated items and by considering the structure of the taxonomy, e.g., levels and siblings. The profile vectors for active agent u_a, agent u_i and product b_k were represented as $\overrightarrow{v_a}= (v_{a_1}, v_{a_2}, ..., v_{a_{|C|}})$, $\overrightarrow{v_i}= (v_{i_1}, v_{i_2}, ..., v_{i_{|C|}})$, and $\overrightarrow{v_k}= (v_{k_1}, v_{k_2}, ..., v_{k_{|C|}})$ respectively. Here, C is the set of all concepts (or topics), and v_{ak} is the preference score of topic $c_k \in C$ for active user u_a. The profile vectors were used to measure the similarity $c(u_a, u_i)$ between two users and the similarity $c(u_a, b_k)$ between active users and items by using the Pearson correlation coefficient measure [121]. The items b_k were recommended to the active agent u_a based on the prediction score $w_a(b_k)$ that was calculated as follows:

$$w_a(b_k) = \frac{q \cdot c_b(u_a, b_k) \cdot \sum_{u_i \in A_a(b_k)} c(u_a, u_i)}{|A_a(b_k)| + Y_R}, \tag{5.1}$$

where $A_a(b_k)$ is the subset of u_a's neighbours who rated item b_k; $q = (1.0 + |f(b_k)| \cdot \Gamma_T)$,

$|f(b_k)|$ is the size of topics of product b_k that user u_a had implicitly rated; and variables Y_R and Γ_T are fine-tuning parameters that allow for customisation of the recommendation process. For the experimental analysis, Ziegler et al. [121] suggested values between 0 and 2.5, and used $\Gamma_T = 1$ and $Y_R = 1$ in their experiments. For more information about Ziegler's taxonomy vector construction algorithm, please refer to [111, 123].

Baseline 2: Item Popularity (IP)

Traditionally, rated items from other users are very useful if we know nothing about a new user. The popularity rank of an item is decided by the number of users that have rated the item, and this information can be obtained from the training set. Some researchers have used item popularity as a means of improving recommendation accuracy [2, 101]. The basic idea is that some popular items are likely to match up with what a new user wants. In this thesis, we assume that our proposed approaches can achieve better performance if we use the proposed hierarchy structure to represent the knowledge of rated items, rather than directly rely on item popularity.

Baseline 3: Matrix Factorization (MF)

The model proposed in this paper has no taxonomy over the item factors. MF techniques are a class of widely successful latent factor models that are currently the most state-of-the-art techniques used [57, 76, 103]. MF is one of the most applied techniques for solving collaborative filtering problems, such as cold-start and data sparsity. For a comparison of models that do not use taxonomy, we used MF to compare the performance of our proposed approaches to achieve better item recommendations and resolve the cold-start problem.

The basic idea behind the techniques is to find unknown ratings associated with users and items in the matrix, and then sort the ratings and select the top k items. The model maps both users and items with a latent factor space of dimensionality. MF is the general term for a broad range of algorithms that can be used to factorize a matrix into a product of matrices, usually two matrices, which can then be multiplied together to give the matrix R. It starts by initialising P and Q to matrices of random numbers, then calculates their product, compares this product to R and then tries to find the local minimum by iteratively reducing the differences between R and PQ. This follows the equation below [103, 114]:

$$R \approx P \times Q^T = \hat{R} \tag{5.2}$$

Let m be the number of users in U, n be the number of items in B, and R be a $m \times n$ matrix, where r_{ij} represents user u_i's rating value to item b_j. Assume P is an $m \times k$ matrix and Q is a $k \times n$ matrix [114]. The k value is the rank of the matrix and represents the number of underlying latent factors being explored in the data.

Accordingly, each item b_j is associated with a vector $q_j \in R^{m \times k}$ and each user u_i is associated with a vector $p_i \in R^{m \times k}$. Let p_i^T denote transposition of the $i - th$ row of P, and q_j is the $j - th$ column of Q. To predict the rating of an item b_j by user u_i, we can calculate the dot product of the two vectors corresponding to u_i and b_j:

$$\hat{r_{ij}} = p_i^T q_j \tag{5.3}$$

The error between the estimated and real rating can be calculated using the equation below:

$$E = \sum_{(u_i, b_j, r_{ij}) \in T} (r_{ij} - \hat{r}_{ij})^2, \tag{5.4}$$

where p_{ik} denotes the elements of $P \in R^{m \times k}$, and q_{kj} the elements of $Q \in R^{k \times n}$. $\hat{r_{ij}}$ is the $i - th$ user who would rate the $j - th$ item according to the model. Let T be a set of tuples, each in the form of (u_i, b_j, r_{ij}). e_{ij} is the error measured at the $(u, j) - th$ rating and E denotes the sum of the squared training errors.

In order to minimise the error, the gradient at the current values of P and Q (p_{uk} and q_{kj}) needs to be determined so that P and Q can be updated in the correct direction. The above equation is then differentiated with respect to p_{uk} and q_{kj} to get:

$$\frac{\partial}{\partial_{p_{uk}}} e_{uj}^2 = -2(r_{uj} - \hat{r_{uj}}(q_{kj}) = -2e_{uj}q_{kj} \tag{5.5}$$

$$\frac{\partial}{\partial_{q_{kj}}} e_{uj}^2 = -2(r_{uj} - \hat{r_{uj}}(p_{kj}) = -2e_{ij}p_{uk} \tag{5.6}$$

Having found the two gradients, we can now update P and Q as follows:

$$p'_{uk} = p_{uk} + 2\alpha e_{uj} q_{kj} \tag{5.7}$$

$$q'_{ik} = q_{ik} + 2\alpha e_{uj} p_{kj}, \tag{5.8}$$

where α is the value that determines the rate at which the local minimum is found. Using a larger alpha will mean a quicker convergence to the minimum, but this runs the risk of missing this minimum and oscillating around it, never to converge. α is usually kept small to reduce this risk and for this purpose α will be kept at 0.0002.

The items were sorted according to the highest rated items from the new ratings matrix $\hat{R}$ and the *N* highest rated items that had not yet been purchased by the active user were returned as recommendations.

Baseline 4: User-Based CF

The standard collaborative filtering (CF) approach is a popular benchmark baseline model. The CF approach is based on user-item relationships [95, 102]. There are two primary approaches to CF: neighbourhood methods and latent factor models [23]. Neighbourhood methods represent the most common approaches to CF. The similarity between two users is calculated based on the overlap of their item sets. The missing rating for any item b_k for a new user u_a is predicted by the average rating of all the neighbours' ratings for item b_k. In order to measure the cosine similarity between two profile vectors, the user-item preference information for both u_i and u_a were profiled in vectors $\overrightarrow{R_i}$ and $\overrightarrow{R_a}$, respectively. The following equation calculates the similarity between two users:

$$S_{a,i} = cosine(\overrightarrow{R_a}, \overrightarrow{R_i}) = \frac{\overrightarrow{R_a} \cdot \overrightarrow{R_i}}{\|\overrightarrow{R_a}\| \times \|\overrightarrow{R_i}\|}, \tag{5.9}$$

where $\overrightarrow{R_a}$ is the row vector of the ratings given to items by a new user u_a, and $\overrightarrow{R_i}$ is the row vector of the ratings given to items by the user u_i.

The item recommendation lists were computed by using a simple weighted average to predict all the missing ratings for a given user u_a. The items that had the highest average rating from a set of similar users u_i were added to the recommendation list and recommended to the active user u_a.

$$\hat{r}_{a,i} = \frac{\sum_{i \in N(a)} (S_{a,i} \times r_{ij})}{\sum_{i \in N(a)} |S_{a,i}|}, \tag{5.10}$$

where the summations are over all peer users u_i rated on items $i \in N(a)$ for the active user a, $s_{a,i}$ the similarity value between the active user u_a and the user u_i in the neighbourhood, and r_{ij} is the rating for user u_i on item j.

5.2.3 Experiment Design

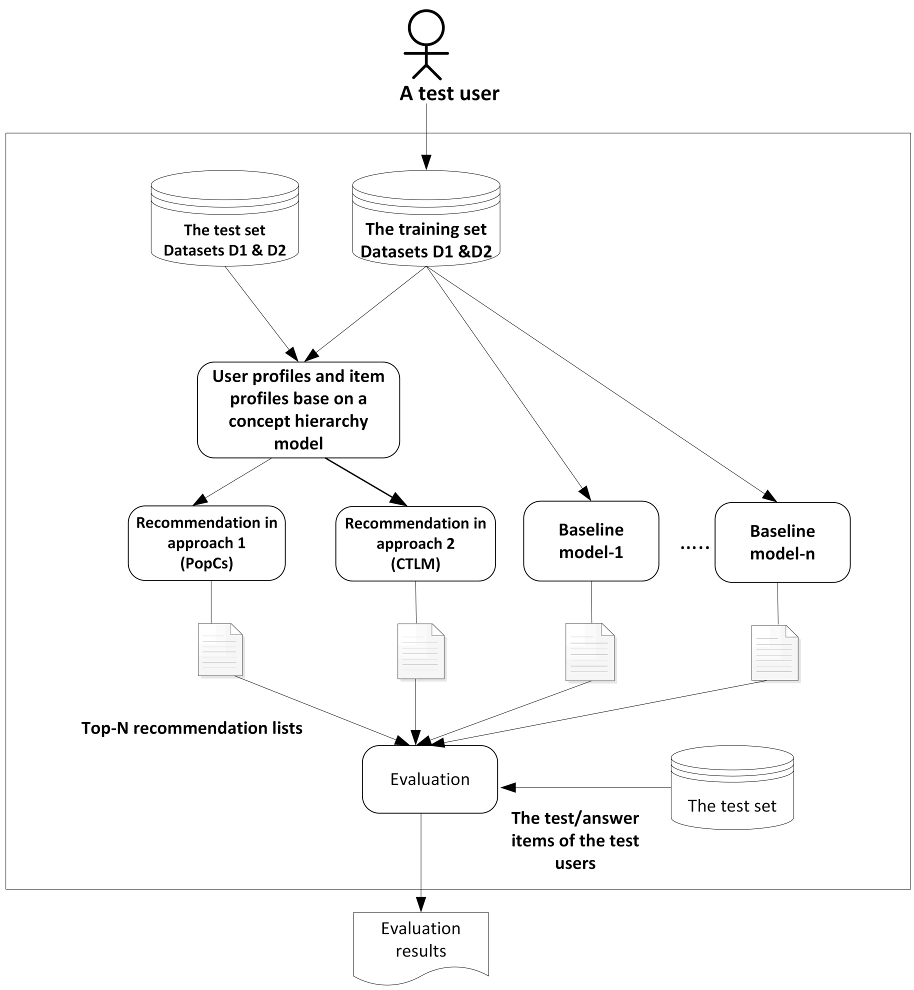

Figure 5.2: The experiments and evaluation framework.

Figure 5.2 demonstrates this thesis's experimental evaluation framework. To conduct these experiments, Datasets D1 and D2 were split into two subsets according to the number of users: the training set W consisted of the first 80% of the users and the test set X contained the remaining 20% of users, which was used as the ground truth. The details are as follows:

- The training set W for Dataset D1 contained 48,475 users, the explicit rating scales were from 1 to 10 and it had 112,413 books. The test set X for Dataset D1 contained 12,122 users and 39,584 books and the categories at leaf level were organised into a taxonomy that was nine levels deep. For the item feature, we extracted the set of categories from their taxonomic descriptors to create the item concept hierarchy. For the user feature, we extracted the set of categories from their rated items' taxonomic descriptors to create the user concept hierarchy.

- The training set W for Dataset D2 contained 368,267 users, the explicit rating scales were from 1 to 5 and it had 78,976 songs. The test set X for Dataset D2 contained 91,993 users and 57,192 songs and the categories at leaf level were organised into a taxonomy that was seven levels deep. Extractions of the item and user features for Dataset D2 were carried out in the same manner as in Dataset D1 (above).

The data in training set W was extracted to generate the item training set W_b. The item training set was utilised to generate the item profiles based on their taxonomy information. The training set was also used to learn the recommendation models, which was then used to generate Top-N items recommendation lists for each user in the test set. The test set was retained as the validation data for testing, and it was constructed to evaluate the quality of the recommender's recommendation.

Each user in the test set $u_a \in X$ was assumed to either be a new user or an active user. Each user in the training set $u_i \in W$ was either an existing or previous user in the system who had previously rated an item. In our experiment, we intuitively decided to not take all training data into account when generating recommendations. As including, the whole training set would negatively influence the performance of recommendations in relation to the required calculation time. In order to make a fair comparison, the random sub-sampling cross-validation technique was applied to evaluate the performance of the proposed approaches. Fifty data subsets (or groups) from the test set X were randomly selected from each of the two datasets and used as the ground truth for the evaluations. For each test subset X, 100 test users were randomly selected and compared with the item training set containing 10,000 items. This was viewed as a run in our experiments. Precision and recall at N, the average precision, mean average precision and the $F_1measure$ values of the 50 runs were used to measure the accuracy performance in terms of the Top-N recommendation task. The performance was also evaluated according to some Top-N recommendation lists in order to compare the proposed model with all the baseline models.

5.2.3.1 Testing Set Setup

Based on the offline evaluations, no actual users participated and the existing dataset was used instead. Typically, the existing dataset consists of items that each user has used. Therefore, in the experiments, a test user u_a in test set X was then selected and some of his or her selections were hidden before asking the recommender to predict a set of items that the test user would use based on the recommendation making approaches. For each test user in the test set, the items associated with this user were hidden as the test/answer item set.

Instead of asking the test users to explicitly describe the concepts of interest, in this thesis we view the selected 50 groups in a test set as 50 active user groups in order to simulate the real practical of the proposed approach. For each test user $u_a \in X$, the preference of a user in the concept hierarchy was acquired from the test user's rated items. A set of taxonomic concepts (or categories) within those rated items' taxonomic descriptors was extracted, and they were used to generate the test user's concept preferences in a two-dimensional hierarchy. The concepts on the list were ordered in a vertical direction (from high level to low level) and sorted based on their hierarchical relationship within the confines of the item taxonomic descriptors. The concepts on the list were ordered in a horizontal direction (from left to right) and sorted according to the frequency of the concept's presence in the rated items' taxonomic descriptors. For each test user's profile, their concept hierarchy preferences were transformed into a profile vector of concepts for the test user $\overrightarrow{u_a}$. The vector thus represents the weights for each concept that was preferred by this user.

5.2.3.2 Training Set Setup

Each item in the item training set $b_k \in W_b$, was represented by a set of taxonomic categories using in its descriptors and placed in the form of an item concept hierarchy. In this case, an item concept hierarchy represented an item profile. Sets of taxonomic concepts and item descriptors were extracted and their relations with an item in a two-dimensional hierarchy were approximated. The concepts on the list were ordered in a horizontal direction (from left to right) and sorted according to the frequency of the concept appearing in its descriptors. The concepts were ordered on the list in a vertical direction (from high level to low level) and sorted based on the hierarchical relationship between each concept within the item taxonomic descriptors.

For each training item's profile, the item's concepts hierarchy were transformed into a profile vector of concepts for the training item $\overrightarrow{b_k}$. The similarity of item b_k to test user

u_a could then be calculated using the cosine similarity measure as discussed in Section 4.3.1. Additionally, the items were previously rated by the existing user u_i, they were used to generate the item popularity. A set of taxonomic concepts within its taxonomic descriptors was utilised to generate the probability of a concept $P(c_i)$ appearing in an item b_k that was relevant to the test user's preference over all the items listed under the taxonomic concepts.

Once the user or item similarities were calculated based on profile vectors, a neighbourhood was formed of the b_k items in the item training set W_b that were most similar to the test user u_a. In this experiment, only k = 50 nearest neighbours were taken into account for each test user u_a. These neighbourhoods were then used to make a prediction for each item b_k that was rated by the test user's neighbours, but not by the test user. The prediction score for each item was generated according to the proposed recommendation approaches (i.e. PopCs and CTLM).

5.2.3.3 The Recommendations Setup

The proposed recommender algorithms worked on the training set W. Once the neighbourhood of the test user had predicted all the items for the user based on the user's u_i preference over all the items under the concept hierarchy, a Top-N recommendation list could be constructed by ranking all the items in descending order of prediction score $p\_score(b_k)$. A Top-N recommendation list for each user in test set $u_a \in X$ was formed by picking the Top-N ranked items from the list. The N items with the highest prediction scores $p\_score(b_k)$ were recommended to the test user u_a. The recommendation list was then compared to the items rated by a user in test set $u_a \in X$. If the item in the Top-N recommendation list was as also found in the test user's hidden test item list, then the item was counted as a hit. Otherwise, it was counted as a miss. The chances of finding a hit increased with N (i.e. 5, 10, 15, 20, 25). The following section will discuss how to apply classification accuracy metrics to evaluate the proposed approaches in terms of Top-N recommendations.

5.2.4 Evaluation Measures

This thesis aims to evaluate the performance of a recommender systems improvement when incorporating taxonomy information based on a concept hierarchy and a language model into the recommendation making approach. A further aim is to evaluate the effectiveness of the proposed approaches in alleviating the cold-start problem. The effectiveness of the proposed recommender's algorithms will be measured according to the recommender's ability to correctly recommend a set of items that a user may use or in which a user may be interested.

To evaluate the performance of the proposed approaches in terms of Top-N recommendations, the classification accuracy metrics (i.e. Precision at N, Recall at N, F_1 measure, Average Precision at N and Mean Average Precision (MAP) were chosen for the accuracy performance evaluation of the recommender against the users in the test set. As discussed in Section 2.5, these metrics were used to evaluate the ranked retrieval results in information retrieval (IR) systems. The evaluation measures can be modified from standard IR to recommender systems as follows.

- *Precision at N* $(Precision@N)$ is defined as the ratio of selected relevant items to the number of items selected at rank N. It is defined in Equation 5.11, where B_{rs} is the set of selected relevant items at rank N, B_s is the set of selected items at rank N and B_r is the set of relevant items. A perfect precision value of 1.0 indicates that every item recommended in the list is a good recommendation. $Precision@N$

represents the probability that a relevant item is selected at rank N.

$$Precision@N = \frac{|B_{rs}|}{|B_s|} = \frac{|B_r \cap B_s|}{|B_s|} \tag{5.11}$$

- *Recall at N* $(Recall@N)$ is defined as the ratio between the number of selected relevant items and the total number of relevant items available in a set of all items B at rank N. It is defined in Equation 5.12, where B_{rs} is the set of selected relevant items at the rank, B_r is the set of relevant items and B_s is the set of selected items that has been retrieved at rank N. $Recall@N$ represents the probability that a relevant item will be selected at rank N. A recall value of 1.0 indicates that all highly recommended items are suggested in the list.

$$Recall@N = \frac{|B_{rs}|}{|B_r|} = \frac{|B_r \cap B_s|}{|B_r|} \tag{5.12}$$

As discussed above, we aim to evaluate the accuracy of the proposed recommender algorithms in making a recommendation list to a user in a test set.To this end, for each test user the items preferred were hidden as the test or answer item set. This test items set T_{u_a} was then used to evaluate the resulting recommendation list Rec_{u_i}. In the experiments the number of selected items $|B_s|$ from Equation 5.11 was defined as the number of recommended items in the Top-N set. Therefore, $|B_s|$ = N. Let T_{u_a} represent the hidden test items set of the active user u_a in the test set X. Let Rec_{u_i} denote the set of Top-N recommended items for each user u_a. The number of selected relevant items at Top-N in Equation 5.13 can be defined as $|B_{rs}|$ = $|T_{u_a} \cap Rec_{u_i}|$. The total number of relevant items $|B_r|$ in Equation 5.12 is the number of test items of the active user u_a, which can be defined as $|B_r|$ = $|T_{u_a}|$. The precision and recall at N of the recommendations are computed as follows [67].

$$Precision@N(u_a) = \frac{|B_{rs}|}{|B_s|} = \frac{|T_{u_a} \cap Rec_{u_i}|}{N} = \frac{|T_{u_a} \cap Rec_{u_i}|}{|Rec_{u_i}|} \tag{5.13}$$

$$Recall(u_a) = \frac{|B_{rs}|}{|B_r|} = \frac{|T_{u_a} \cap Rec_{u_i}|}{|T_{u_a}|} \tag{5.14}$$

For example, if the set B_r (or T_{u_a}) of relevant items is $\{b_1, b_2, b_3, b_4, b_5, b_7, b_{10}\}$, the recommender system would recommend the set of items B_s (or Rec_{u_i}) = $\{b_2, b_1, b_4, b_7, b_{11}\}$ for the user u_a. Therefore, at Top- N = 5, the precision of this recommendation would be $|\{b_1, b_2, b_3, b_4, b_5, b_7, b_{10}\} \cap \{b_2, b_1, b_4, b_7, b_{11}\}|/|\{b_2, b_1, b_4, b_7, b_{11}\}|$ = 4/5, and the recall would be $|\{b_1, b_2, b_3, b_4, b_5, b_7, b_{10}\} \cap \{b_2, b_1, b_4, b_7, b_{11}\}|/|\{b_1, b_2, b_3, b_4, b_5, b_7, b_{10}\}|$ = 4/7.

- The $F_1 measure$ or $F_1 score$ at N ($F_1@N$) is calculated as the standard harmonic mean of precision and recall at N. It is used to present the overall performance.

$$F_1@N = \frac{2 \times Precision@N \times Recall@N}{Precision@N + Recall@N} \tag{5.15}$$

- *Average Precision at N* (AP) is the average of the precision (AP) at the rank across all the test user U in a test set X. In our context, AP can be defined as follows.

$$AP_U@N = \frac{\sum_{u_a \in U} Precision@N(u_a)}{|U|} \tag{5.16}$$

where $|U|$ is the number of active users u_a in a test set X, and $Precision(u_a)$ is the precision at N of the recommendations for an active user $u_a \in U$.

- *Mean Average Precision* (MAP) is the mean of the average precision $@N$ scores for all the users in the test sets. It is applied to calculate the average precision at N for these predictions in multiple test sets, which is computed by Equation 5.17.

$$MAP = \frac{\sum_{U \in X} AP_U@N}{|X|} \quad (5.17)$$

where $|X|$ is the number of test sets, and $AP_U@N$ is the average precision of the recommendations at Top-N of all the users in each test set $u_a \in X$.

5.2.5 Experimental Environment

To implement and evaluate the effectiveness of the proposed approaches including user profiling, item representation, recommendation making approaches and other related state-of-the-art baseline models, the experimental environment included the development environment and the system environment as follows:

1) The experiment was mainly conducted on a personal computer with an Intel (R) Core (TM) i7-2600 CPU @3.4 GHz with 8 GB memory running on a Window 7 Enterprise 64-bit Operating System. Microsoft SQL Server 2008 was installed as the database management system (DBMS). Microsoft Visual C#-2010 or 2012 was used as the programming language.

2) When it came to the implementation of the training sets and the test set, the two datasets, D1 and D2, were extracted and converted into txt files. Then, they were mapped onto a logical database structure (see an example in Appendix C). These two datasets were used to generate the recommendation applications and the four other baseline models. After the two datasets were successfully stored in the database, each was then divided into the training set and the test set. As discussed in Section 5.2.3, 20% of the users made up the test set, and they were used to create the active users' concept hierarchy. The weight of each concept in the hierarchy in the active user profile was calculated according to the steps outlined in Section 3.5.2.

For the other 80% of users that made up the training dataset, the training dataset was used to construct the item training set. All items in the item training set were extracted to build the item concept hierarchy. They were also utilised to construct the item popularity. A set of taxonomic concepts (or categories) within the item descriptors was used to construct the concept probability. The weight of each concept in the hierarchy in the item profile could then be calculated according to the steps provided in Section 3.4.2. The concept weight values of each training item and each test user as well as the item popularity weight values and the concept probability values were stored in the database.

3) Implementation of the recommendation applications and other baseline models. When the three datasets had been successfully set up, we implemented the recommendation applications and connected them with the two datasets in the database. Then we ran the files in Microsoft Visual Studio 2010.

5.3 EXPERIMENTAL RESULTS AND DISCUSSION

The results of the two proposed approaches will be examined with regard to their recommendation accuracy through a comparison with the aforementioned four baseline models. Furthermore, we will analyse the results of the proposed concept hierarchy model and the proposed language based model used to provide a better solution to the cold-start problem and improve the accuracy of the recommendations. The two proposed recommendation approaches are:

1) **PopCs:** This proposed recommendation approach is designed based on linearly combining the item popularity and the user-to-item concept hierarchy similarities. Two aspects that are taken into consideration are: (1) other users' opinions of items are considered when determining the popularity of the item; (2) a users' affinity for taxonomic

concepts attached to items (as represented by user-item concept hierarchy similarities). This contributes to solving the cold-start problem and improves the accuracy of the recommendations.

2) **CTLM:** This proposed recommendation approach is designed and developed based on the relevance of an item to a user as a kind of language model (LM) to understand how items are likely to generate what the user wants. A combination of item popularity, user-item concept hierarchy similarities and the probability of concept relevance are employed to generate the recommendation function. This contributes to solving the cold-start problem and improves the accuracy of the recommendations.

5.3.1 Results of the PopCs Recommendation Approach

The objective of this experiment is to verify the following hypothesis:

- **[Hypothesis 1:** The proposed concept hierarchy model based on taxonomy can effectively improve recommendation accuracy and solve the cold-start problem.

According to Hypothesis 1, there are two objectives of the evaluation. Firstly, to evaluate whether the accuracy of the recommendations can be improved by using the proposed concept hierarchy model based on taxonomy according to the relationship between users and items and integrating them into the recommendation making process. Secondly, the evaluation aims to evaluate whether the cold-start problem can be solved by the proposed PopCs.

As discussed in Chapter 3 and 4, this thesis proposes a taxonomy-based approach to make an important contribution to the cold-start problem and to enhance the personalised recommendation making approaches. Therefore, a user profile and item representation based on the concept hierarchy model were utilised to find the similarity of an item to the user based on the similar concept hierarchies $cs(u_a, b_k)$ which was then used to generate the proposed PopCs recommendation approach. The PopCs recommendation approach is a combination of the user-item concept hierarchy similarities $cs(u_a, b_k)$ and item popularity $npop(b_k)$.

To prove the above hypothesis, the parameterisation of the proposed PopCs approach will be discussed. Then, the MAP comparison of the recommendation accuracies of the proposed PopCs approach with the state-of-the-art user-based CF baseline model for Dataset D1 will be discussed in detail. The comparisons with existing baseline models to provide improved solutions to the cold-start problem will be presented.

5.3.1.1 Parameterisation

$$p\_score(u_a, b_k) = \alpha \times npop(b_k) + (1 - \alpha) \times cs(u_a, b_k) \tag{5.18}$$

As mentioned earlier, the purpose of this experiment to verify that the proposed concept hierarchy model can effectively improve recommendation accuracy. In Equation 5.18, $p\_score(u_a, b_k)$ stands for the prediction score obtained by linearly combining the item popularity with the user-item concept hierarchy similarities. The parameter α was defined as the experimental coefficient to estimate the item popularity $npop(b_k)$ and user-item concept hierarchy similarities $cs(u_a, b_k)$ that can achieve the best performance of MAP. We conducted the experiment by setting the value of α from 0 to 1. The results indicated that with α = 0.2, the proposed PopCs approach achieved the best performance on Dataset D1.

The following is a discussion of the best settings of the parameter α for the proposed PopCs approach.

5.3.1.2 Experimental Results

To evaluate the effectiveness of the proposed PopCs recommendation approach based on taxonomy, the state-of-the-art user-based collaborative filtering (user-based CF) baseline approach was chosen as the baseline model to compare the proposed PopCs recommendation approach. The standard collaborative filtering (CF) approach is a popular benchmark baseline model. The CF approach is based on user-item relationships [95, 102]. There are two primary approaches to CF: the neighbourhood method and the latent factor models [23]. The neighbourhood method represents the most common approach to CF in which the similarity between two users is calculated based on the overlap of their item sets. A missing rating for any item b_k for new user u_a is predicted by the average rating of all the neighbours' ratings of item b_k. Further details are provided in Section 5.2.2.

This thesis studied the performance of the proposed PopCs approach to understand how it can improve the accuracy of item recommendations and alleviate the cold-start problem. The experimental results were observed and obtained from 50 runs on Dataset D1. Each test set was constructed by randomly choosing 100 test users from the 12,122 users. We randomly selected 10,000 items from the training item set to form the top k = 50 neighbour items that were most similar to the test user u_a (or new user). The similarity between the active user and all other items was identified based on two concept profile vectors (i.e. user-item concept hierarchy similarities).

These neighbourhoods were then used to make a prediction for each item b_k that was rated by a test user's neighbours. The prediction making for each item b_k was calculated according to each recommendation method. A Top-N recommendation list was generated by ranking all the items in descending order of the computed prediction scores. To evaluate the recommendation qualities of the proposed approaches, we let each recommendation approach recommend a list of N items to each test user. The Top-N

MAP evaluation results of Dataset D1 are shown in Figure 5.3.

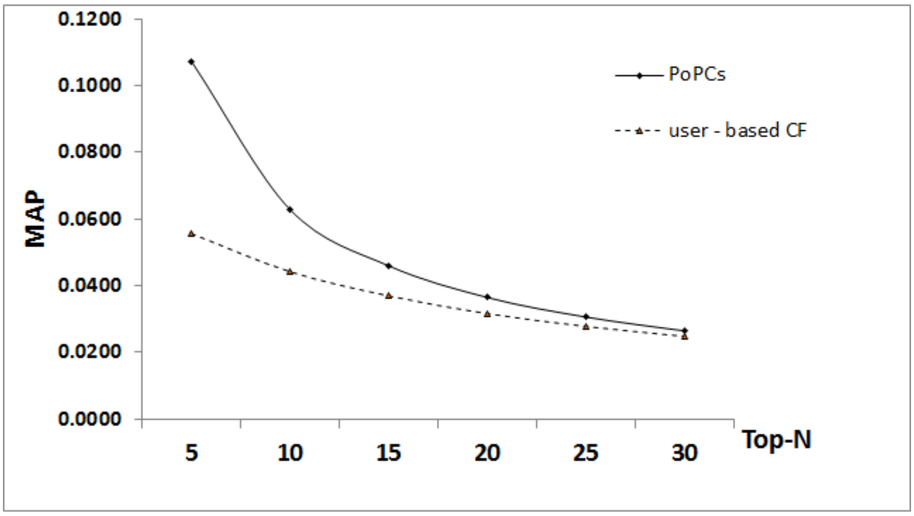

Figure 5.3: The MAP comparison of PopCs and user-based CF based on Dataset D1.

From the experimental results illustrated in Figure 5.3, the proposed PopCs approach performed better than the baseline model user-based CF. PopCs achieved the best MAP result with α = 0.2. The results indicated that after combining the proposed concept hierarchy model (as represented by the similarity of a user to an item based on concept hierarchies) with a user's opinions about the items in terms of their popularity, the accuracy of the item recommendations can be further improved upon. In contrast, the baseline model user-based CF makes recommendations based on the ratings that users have given to items. As discussed in subsection 5.2.1, the distribution of a user's rating data on Dataset D1 was sparse or less distributed. Therefore, this suggests that the results of the baseline model user-based CF might be affected by this circumstance. However, this improvement also suggests that the proposed PopCs can effectively improve the recommendation accuracy even when the amount of explicit user rating data in the system is small. Consequently, **Hypothesis 1** is valid.

5.3.1.3 Discussion and Analysis for PopCs

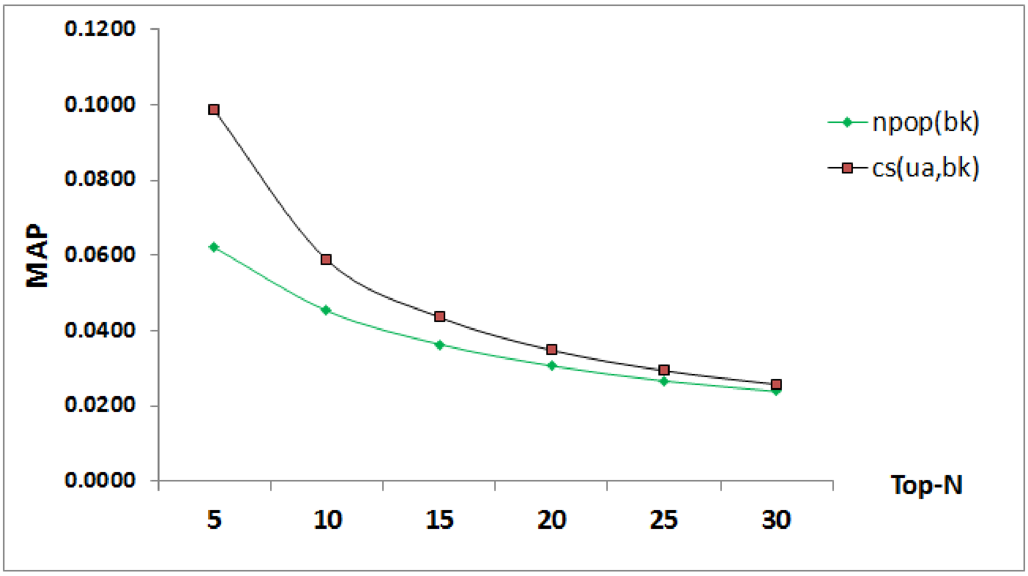

Figure 5.4: The performance comparison between item popularity and the use of the concept hierarchy model.

The goal of the proposed PopCs approach is to enhance the ability to make personalised recommendations. In turn, this would contribute to resolving the cold-start problem and improving the accuracy of recommendations. The new user problem, or when the amount of explicit rating data in the system is small, makes the formation of neighbourhoods in traditional collaborative filtering systems inaccurate, which thereby results in poor recommendations. This thesis proposes the use of the concept hierarchy model to formalise the problem. When rating data was lacking, or when faced with the new user problem, the proposed concept hierarchy model benefited from the use of the standard taxonomy vocabulary (i.e. product categories) combined with the tree structure of taxonomy information in order to obtain user preferences in a two-dimensional hierarchy. The concept hierarchy can provide a better structure to a new user in order to approximate their needs. In the proposed PopCs approach, the concept hierarchy model was represented by the user-item concept hierarchy similarities $cs(u_a, b_k)$.

In order to evaluate the performance of the proposed concept hierarchy model, the proposed $cs(u_a, b_k)$ based concept hierarchy model was compared with the item popularity $npop(b_k)$. Figure 5.4 shows the performance of the two components-item popularity $npop(b_k)$ and the user-item concept hierarchy similarities $cs(u_a, b_k)$-that are based on the concept hierarchy model. To verify that the proposed taxonomy-based concept hierarchy model can provide a solution to effectively improve the cold-start problem, we defined the parameter α as the fine-tuning parameter to estimate the performance of the item popularity $npop(b_k)$ and the user-item concept hierarchy similarities $cs(u_a, b_k)$. We conducted the experiment by setting α = 1 to verify the effectiveness of the $npop(b_k)$ approach, while setting α = 0 to verify the effectiveness of the $cs(u_a, b_k)$ approach. The results showed that the proposed $cs(u_a, b_k)$ approach based on the concept hierarchy model outperformed the proposed $npop(b_k)$ approach on Dataset D1. This improvement suggested that after considering only the proposed concept hierarchy model, the accuracy of item recommendations based on item taxonomy can be further improved.

Obviously, when rating data is lacking, when it comes to the formation of the neighbourhoods, the proposed user-item concept hierarchy similarities approach $cs(u_a, b_k)$ benefits from the proposed concept hierarchy model based on taxonomy. This concept hierarchy model is especially effective in the case of cold-start situations where users have very limited item rating data. Moreover, using item taxonomy information (i.e. category taxonomy) is also useful for finding the similarities between users and items when there is insufficient ratings data in the systems, when there is limited new user information or a new item just is added to the systems. The recommender systems can still find similar users or similar items that have similar taxonomic concepts to a target user and that new item. Therefore, the quality of the neighbourhood forming can be improved and becomes more accurate.

In contrast to other taxonomy weighting approaches, our proposed taxonomic topic (or concept) weighting approach considers the two-dimensional hierarchy into consideration when capturing a user's concept preferences. This improvement suggests that the user with a preference should be allowed to be associated with concepts in both of the two dimensions of hierarchy in order to reflect the actual user's needs. Consequently, a more accurate user profile can be gained and more accurate recommendations can be made. Therefore, the experimental results suggest that the proposed concept hierarchy model-based taxonomy can provide an effective solution to the cold-start problem, and thereby validate **Hypothesis 1**.

5.3.2 Results of the CTLM Recommendation Approach

The objective of this experiment was to verify the following hypothesis:

- **[Hypothesis 2]:** The Concept Taxonomy with Language Model recommendation approach can effectively improve recommendation accuracy.

To verify the above hypothesis, we will first introduce the parameterisation of the proposed CTLM recommendation function. Then, the experimental results of the proposed CTLM approach will be discussed. This proposed approach was evaluated on two datasets, D1 and D2.

5.3.2.1 Parameterisation

$$\begin{aligned} p\_score(u_a, b_k) = \alpha \times npop(b_k) + (1-\alpha)[\beta \times cs(u_a, b_k) \\ + (1-\beta) \times \sum_{c \in H_{u_a} \cap H_{b_k}} P(c_i)] \end{aligned} \tag{5.19}$$

The significance of the proposed CTLM function is to enhance the personalised item recommendations. It contributes to improving the accuracy of item recommendations and alleviates the cold-start problem. The proposed CTLM function was designed under the language model in order to estimate the probability of a user u_a being interested in an item according to a set of taxonomic concepts that were utilised by the user. The proposed CTLM function in Equation 5.19 was designed by combining the two primary components based on 1) other users' ratings of items in order to decide the popularity of the item, which was represented by item popularity $npop(b_k)$; and 2) the relationship between a user's concept hierarchy and other items' concept hierarchies. The latter component can be further divided into two sub-components: 2.1) the similarity of a user to an item based on concept hierarchies $cs(u_a, b_k)$; and 2.2) the probability of the concepts used in the concept hierarchies $P(c_i)$.

According to Hypothesis 2, the objective of the evaluation in this experiment is to determine whether the recommendation accuracy can be improved by incorporating the aforementioned primary components with the language model into the recommendation making process. This is predicated on the basic assumption that if the language model of an item can generate the concepts that characterise the users item interests, the user is likely to be interested in the corresponding item. To verify the proposed CTLM recommendation function in Equation 5.19, α and β were experimental coefficients used to estimate the item popularity $npop(b_k)$ and the user-item concept hierarchy similarities $cs(u_a, b_k)$, along with the probability of the concept relevance $P(c_i)$ that can achieve the best MAP and $F_1measure$ performances. We conducted the experiment by setting the value of α and β from 0 to 1. The results indicated that the proposed CTLM approach achieved the best performance in Dataset D1 with α = 0.3 and β = 1, and with α = 0.02 and β = 0.03 inn Dataset D2. In addition, the parameters α and β were designed to evaluate the performance of the CTLM component on the cold-start analysis. This will be further discussed in subsection 5.3.3.1.

5.3.2.2 Experimental Results of CTLM

To evaluate the effectiveness of the proposed CTLM approach and to understand how it can improve the accuracy of the item recommendations and alleviate the cold-start problem, the following three baselines approaches were chosen to be compared with the proposed CTLM:

- **ITB**: Item Taxonomy Recommendations, provided by Weng et al. [111] and based on Ziegler's idea [123], were used. In order to make a fair comparison with the proposed approaches, the ITB approach examined item recommendation making by using item taxonomy information. This approach used a set of taxonomic categories in item descriptors to create the task of item recommendations. It combined explicit item ratings and category taxonomies into profile vectors to generate the users' taxonomic preferences. The profile vectors were constructed by summarising all the taxonomy descriptors from all the rated items and by considering the structure of the taxonomy. This is further discussed in subsection 5.2.2.
- **IP**: The sole item popularity is an approach that does not use taxonomy. For the purposes of comparison, we assumed that if the proposed recommendation approaches could achieve better item recommendations and greater accuracy of prediction than item popularity, this could demonstrate that the proposed recommendation approaches are able to make an additional contribution in order to improve the performance of the recommender systems. The rankings of the item popularity were based on the number of relevant ratings, which was obtained from the training set. Other studies have also used item popularity in order to compare the performance of Top-N item recommendations [2, 101].

- **MF**: The matrix factorization technique is a class of widely successful latent factor models which are current state-of-the-art techniques [57, 76]. We compared the proposed approaches against the basic matrix factorization model. To undergo a comparison without using taxonomy, we used MF to compare the performance of our proposed approaches to achieve better item recommendations and a better solution to the cold-start problem. This was discussed in subsection 5.2.2.

The results were observed from 50 runs on two datasets: D1 and D2. Figures 5.5 and 5.6 illustrate the Top-N (N = 5..., 25) MAP and $F_1measure$ evaluation results of Datasets D1 and D2 for the proposed model CTLM and three other baseline models: ITB, IP and MF.

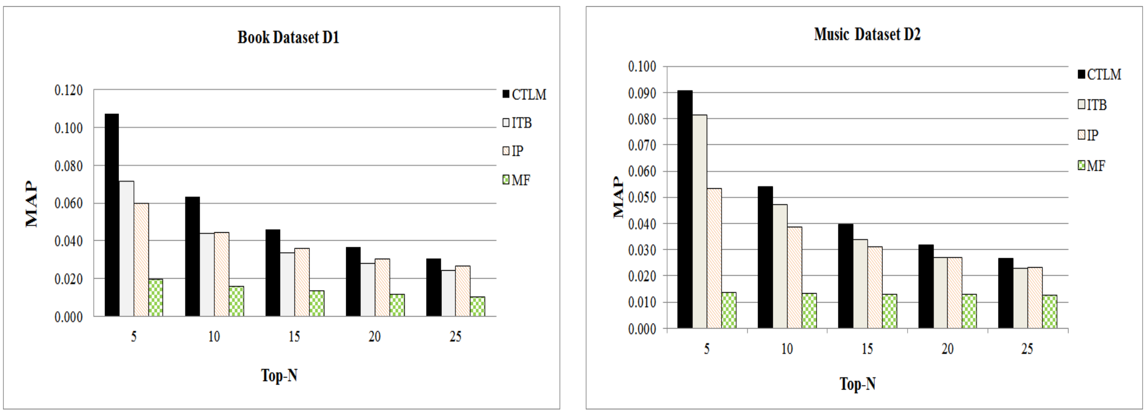

Figure 5.5: The MAP comparison of Datasets D1 and D2

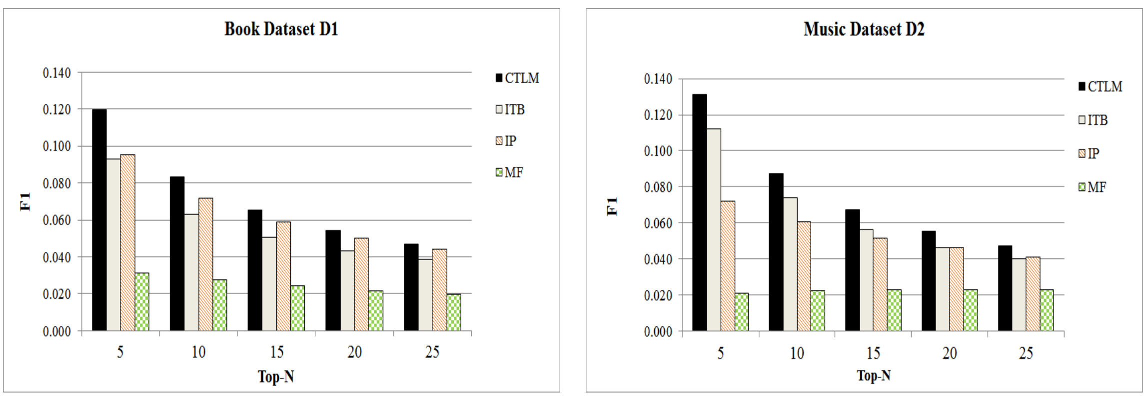

Figure 5.6: The $F_1measure$ comparison of Datasets D1 and D2

5.3.2.3 Discussion and Analysis of CTLM

As shown in Figures 5.5 and 5.6, the proposed CTLM approach significantly outperformed the other three models on both datasets. CTLM achieved the best MAP performance and $F_1 measure$ results for Dataset D1 with α = 0.3 and β = 1, and with α = 0.02 and β = 0.03 for Dataset D2. In contrast, the MF approach had the worst MAP and $F_1 measure$ results for both Datasets D1 and D2, and it failed to improve the recommendation accuracy. Moreover, it was interesting to note that when comparing the performance of CTLM with the baseline model ITB, CTLM performed better than the ITB. The difference between CTLM and ITB are that the method to compute the taxonomic concept preferences as well as the method to generate recommendations are different. This can be seen in how the ITB technique proposed by Weng et al. [111] makes item recommendation based on the taxonomic topic weighting approach proposed by Ziegler et al.[121]. The ITB's topic weighting approach integrates the users' rating data with the number of siblings a topic has in the item taxonomy tree and the length of an item's taxonomic descriptor in order to generate the users' taxonomic topic (or concept) preferences. This approach takes only the vertical direction of hierarchy into consideration when generating a user's topic preferences. The performance of the baseline model ITB might be affected by this circumstance.

In contrast, the proposed CTLM approach generates item recommendations based on the underlying language model and the relevance of an item to a user according to the popular items recommended by other users, the similarity of a user to an item based on concept hierarchies $cs(u_a, b_k)$, and frequently used concepts in item descriptors (or item concept hierarchy) of a user, which was represented by the proposed concept probability method $P(c_i)$. In information retrieval (IR), LM provides an effectiveness approach for document ranking and retrieval. LM combines both semantic and statistical inference to estimate of probability of relevance of documents to a given query and ranked document based on the probability of the language model generating the query text [25, 26].

The adaptation of a language model to estimate the probability of the users being interested in the items is shown in Equation 5.19 and the results are shown in Figures 5.5 and 5.6. The proposed CTLM approach provided a significant improvement in the overall recommendation accuracy after considering the language model approach into the item recommendation making process. The better performance of the proposed CTLM approach suggests that LM is successful applied in item recommendations. The language model approach to an item can generate the concepts that characterise the users' item interests. In addition, the proposed CTLM approach can benefit from using concept taxonomy which is represented by the proposed $cs(u_a, b_k)$ and $P(c_i)$ methods when capturing a user's concept preferences. Different from the topic weighting approach proposed by Ziegler et al. [121] and Weng et al. [111], we take the two-dimensional of hierarchy into consideration when computing the weight of a taxonomic concept for a user and an item. It is represented by the proposed $cs(u_a, b_k)$. The proposed $P(c_i)$ method uses the frequency of a concept's occurrence in an item's descriptors to estimate the probability of the user's concepts of interest. It can result in a more comprehensive user profile and lead to accurate recommendations.

The performance of the proposed CTLM approach benefits greatly from incorporating the popular items recommended by other users and the probability of a user being interested in concepts of the items. The results strongly support that the argument proposed CTLM approach can effectively improve the accuracy of the item recommendations, thereby validating **Hypothesis 2**.

The following subsections will discuss the percentage change,t-test comparison and the concept taxonomy for the cold-start problem.

5.3.2.4 The Percentage Changes for CTLM

Book Dataset D1 (MAP)						
Top-N	CTLM	ITB	IP	MF	**%chg1**	**%chg2**
5	0.10724	0.07172	0.06004	0.01948	**192.88**	**49.53**
10	0.06316	0.04390	0.04424	0.01576	**129.13**	**43.87**
15	0.04608	0.03385	0.03584	0.01337	**103.08**	**36.12**
20	0.03668	0.02813	0.03030	0.01167	**88.59**	**30.39**
25	0.03066	0.02446	0.02643	0.01045	**78.30**	**25.38**
Music Dataset D2 (MAP)						
Top-N	CTLM	ITB	IP	MF	**%chg1**	**%chg2**
5	0.09068	0.08128	0.05344	0.01356	**216.66**	**11.56**
10	0.05390	0.04718	0.03866	0.01318	**120.87**	**14.24**
15	0.03957	0.03380	0.03100	0.01289	**83.89**	**17.08**
20	0.03187	0.02687	0.02692	0.01285	**61.67**	**18.61**
25	0.02672	0.02280	0.02341	0.01259	**47.85**	**17.19**

Table 5.1: MAP percentage changes in Datasets D1 and D2

The use of percentage changes is also very popular when comparing model performances. The following formula is used to compute the percentage change of a model against a baseline model.

$$\%chg = \frac{Result_{model} - Result_{baseline}}{Result_{baseline}} \times 100\% \tag{5.20}$$

Table 5.1 illustrates the MAP and Table 5.2 illustrates the values of the $F_1 measure$ for all the models and the percentage changes in both datasets. $\%chg1$ is denoted as the average percentage changes in the proposed model against the other three baseline models. From the tables, it can be seen that the baseline model ITB outperformed the other two models. We also compared it with CTLM, where $\%chg2$ is denoted as the

Book Dataset D1 ($F_1 measure$)						
Top-N	CTLM	ITB	IP	MF	**%chg1**	**%chg2**
5	0.11978	0.09292	0.09519	0.03134	**112.33**	**25.84**
10	0.08343	0.06294	0.07186	0.02766	**83.43**	**16.10**
15	0.06531	0.05067	0.05902	0.02436	**69.23**	**10.65**
20	0.05442	0.04336	0.05034	0.02162	**61.77**	**8.10**
25	0.04690	0.03848	0.04432	0.01959	**55.72**	**5.82**
Music Dataset D2 ($F_1 measure$)						
Top-N	CTLM	ITB	IP	MF	**%chg1**	**%chg2**
5	0.13118	0.11237	0.07224	0.02105	**207.19**	**81.60**
10	0.08750	0.07403	0.06083	0.02261	**116.35**	**43.84**
15	0.06725	0.05625	0.05181	0.02283	**81.31**	**29.80**
20	0.05546	0.04644	0.04635	0.02326	**59.18**	**19.67**
25	0.04721	0.04005	0.04112	0.02309	**45.73**	**14.83**

Table 5.2: $F_1 measure$ percentage changes in Datasets D1 and D2

average percentage changes in the proposed CTLM model against the best baseline model ITB. For example, the average $\%chg1$ of the $Top-5$ of the book dataset D1 can be calculated as follows:

$$\%chg1 = (\frac{0.10724 - 0.07172}{0.07172} + \frac{0.10724 - 0.06004}{0.06004} + \frac{0.10724 - 0.01948}{0.01948})/3 \times 100 = 192.88\%$$

$$\%chg2 = (\frac{0.10724 - 0.07172}{0.07172})/1 \times 100 = 49.53\%$$

The information contained in Tables 5.1 and 5.2 indicates that the proposed approach of the CTLM was very impressive with regard to its effectiveness and demonstrates that it can significantly improve the performance of recommender systems in terms of their item recommendations.

5.3.2.5 T-test Comparison for CTLM

We also used a statistical method, the paired two-tailed *t-test*, to analyse and compare the experimental results. If the associated p-value is low ($<$ 0.05), that means that the difference between the proposed model and the current examining model (a baseline model) is significant. Tables 5.3 and 5.4 show the *t-test* results which illustrate that the performance of the proposed model CTLM is significantly better when compared with the other models, item taxonomy baseline (ITB), item popularity (IP), and matrix factorization (MF), for $Top - 5$ and $Top - 25$ MAP.

Baseline Models			
Top-N	IP	ITB	MF
5	5.36021E-33	2.94912E-30	7.29678E-43
10	2.61486E-28	4.53523E-31	7.17458E-39
15	5.70535E-26	1.03157E-28	3.46491E-39
20	5.37003E-24	6.73914E-25	9.64221E-38
25	3.17916E-1	1.39E-21	1.12413E-35

Table 5.3: T-test results of CTLM against the other three baseline approaches for Dataset D1

Baseline Models			
Top-N	IP	ITB	MF
5	9.3535E-25	1.6129E-06	1.6684E-37
10	5.37643E-23	9.203E-12	5.6118E-35
15	1.3344E-20	2.369E-14	1.6525E-30
20	5.2052E-21	1.1924E-17	5.4716E-29
25	2.2002E-17	1.0124E-17	1.6796E-25

Table 5.4: T-test results of CTLM against the other three baseline approaches for Dataset D2

Additionally, we further analysed the performance of CTLM components that can contribute to improving the cold-start issue. More in-depth discussion is provided in the following subsection.

5.3.2.6 Concept Taxonomy for Cold-start Problem

One of the aims of this research was to verify the following hypothesis:

- **[Hypothesis 3]:** Use of the concept taxonomy can effectively alleviate the cold-start problem.

To verify the above hypothesis, we further investigated the performance of the proposed CTLM model in order to understand how it can alleviate the cold-start problem by enhancing the performance of the item recommendation. The performance of the three components of the proposed CTLM approach: item popularity $npop(b_k)$, the similarity of a user to an item based on concept hierarchies $cs(u_a, b_k)$ and the concept probability $P(c_i)$ will be analysed and discussed in detail.

- Parameterisation Analysis for the Cold-start Problem

In Equation 5.19, we used the three components to work out accurate predictions for relevant items: item popularity $npop(b_k)$, the user-item concept hierarchy similarities $cs(u_a, b_k)$, and the concept probability $P(c_i)$. $P(c_i)$ and $cs(u_a, b_k)$ are proposed by using the concept taxonomy, while the $npop(b_k)$ approach represents other users' opinions on popular items.

To understand how this model could alleviate the cold-start problem, we remove the item popularity $npop(b_k)$ component from Equation 5.19. We defined the parameters α and β as the fine-tuning parameters to estimate each component of the proposed CTLM approach, where $0 \geq \alpha, \beta \leq 1$. Parameter α was given to estimate the importance of using item popularity $npop(b_k)$, while parameter β estimates the importance of using $cs(u_a, b_k)$ and $P(c_i)$ components.

We assumed that if $P(c_i)$ and $cs(u_a, b_k)$ could contribute more than $npop(b_k)$, we could then determine the effectiveness of the proposed method by using concept taxonomic hierarchies and the probability of concepts would help to ameliorate the cold-start problem. In this experiment, we studied the benefits of using concept taxonomy, which was represented by the user-item concept hierarchy similarities $cs(u_a, b_k)$, and the concept probability in all relevant item hierarchies $P(c_i)$. Our main focus was to examine the effectiveness of using concept taxonomy rather than item ratings.

We compared the results generated using the item popularity $npop(b_k)$, with $\alpha = 1$ and $\beta = 0$ in Equation 5.19. Then, we compared them using only $P(c_i)$ by defining $\alpha = 0$ and $\beta = 0$; and $cs(u_a, b_k)$ with $\alpha = 0$ and $\beta = 1$. Lastly, we evaluated $npop(b_k)$ over the combination of the other two components: $cs(u_a, b_k)$ and $P(c_i)$, with $\alpha = 0$ and β = 0.1. The MAP results for the three components for Datasets D1 and D2 are shown in Table 5.5.

5.3.2.7 Discussion and Analysis of a Concept Taxonomy for the Cold-start Problem

As shown in Table 5.5, the results suggest that the proposed concept probability $P(c_i)$ and user-item concept hierarchy similarities $cs(u_a, b_k)$ methods based on concept taxonomy achieved higher performance than the proposed item popularity approach $npop(b_k)$. The two proposed $P(c_i)$ and $cs(u_a, b_k)$ methods benefited fro using the standard taxonomy vocabularies, such as taxonomic category, in the item descriptors and the relevance of an item to a user according to concepts taxonomy that user interact with and the item has to alleviate the cold-start problem.

The CTLM Components Comparison on Dataset D1 (MAP)					
Top-N	$npop(b_k)$	$P(c_i)$	$cs(u_a, b_k)$	$cs(u_a, b_k)\&P(c)$	$CTLM$
5	0.06004	0.08296	0.10052	0.09144	0.10724
10	0.04424	0.04980	0.05944	0.05434	0.06316
15	0.03584	0.03755	0.04384	0.04075	0.04608
20	0.03030	0.03078	0.03491	0.03310	0.03668
25	0.02643	0.02650	0.02938	0.02829	0.03066

The CTLM Components Comparison on Dataset D2 (MAP)					
Top-N	$npop(b_k)$	$P(c_i)$	$cs(u_a, b_k)$	$cs(u_a, b_k)\&P(c)$	$CTLM$
5	0.05344	0.09068	0.06128	0.08002	0.09068
10	0.03866	0.05384	0.04170	0.04910	0.05390
15	0.03100	0.03928	0.03264	0.03676	0.03957
20	0.02692	0.03126	0.02749	0.02994	0.03187
25	0.02341	0.02608	0.02379	0.02544	0.02672

Table 5.5: The performance of CTLM components on Datasets D1 and D2

The proposed $P(c_i)$ method benefits from using the frequently used concepts of the user u_a to estimate the probability of the user's interest. While, the proposed $cs(u_a, b_k)$ benefits from the proposed concept hierarchy model-based taxonomy. The proposed $cs(u_a, b_k)$ method takes the two-dimensional hierarchy when calculating the weight of a taxonomic concept for a user and an item. It allows users to transmit their preferences in a concept hierarchy rather than using ratings alone. It provided a better structure for new users to approximate their needs in a two-dimensional hierarchy. In the case of lacking rating data or new users first enter the systems, the two proposed $P(c_i)$ and $cs(u_a, b_k)$ methods are useful for finding users' preferences by using a set of taxonomic concepts that users interact with and the items are associated with. They also provide another way to determine the level of interest of the user in the item. As the results, a more comprehensive user preference profile and a more accurate user profile can be generated, thereby providing more accurate recommendations.

Evidently, concept taxonomy provides valuable information on the relationship between users, items, set of standard vocabularies, and the hierarchical structure of taxonomy in order to obtain the user preferences when there is a small amount of rating data available in the systems. It also implies user interests or preferences in an item. Based on the experimental results, the use of concept taxonomy in the proposed $P(c_i)$ and $cs(u_a, b_k)$ methods provided significant improvement in recommendation making in both normal and cold-start environments. It is clear that the proposed concept taxonomy is helpful for new users to obtain preferred items, therefore validating **Hypothesis 3**.

5.3.2.8 Efficiency of the CTLM Recommender Algorithm

The efficiency of an algorithm is measured based on the time it takes for the algorithm to run as a function of the input size by using the Big-O notation concept. Big-O notation is the language used to articulate how long an algorithm takes to run. There are many factors affecting this time, such as the speed of the computer's processor, memory, and so on. Consequently, it is difficult to estimate the exact runtime of an algorithm. Therefore, this thesis use Big-O notation to express how quickly the algorithm's runtime grows. The input of this algorithm is m, which is denoted as the number of active users $|U|$ in the test set and n is the number of the items $|B|$ that is used in the training set. Based on the proposed *CTLM* recommender algorithm in subsection 4.4.2.3, the efficiency of the algorithm can be described as follows:

- step 1: $\mathcal{O}(n * m)$
- step 2: $\mathcal{O}(n)$

- step 3: $\mathcal{O}(n)$
- step 4: $\mathcal{O}(1)$
- step 5: $\mathcal{O}(1)$
- step 6: using Algorithm 1 as shown in Chapter 3, section 3.5.3. For each item profile vector $\overrightarrow{b_k}$, there are three in average descriptors D to get the concept hierarchy H, where p is the size of a descriptor. The total time complexity is $\mathcal{O}(p) + \mathcal{O}(p^2) + \mathcal{O}(p) + \mathcal{O}(p^2) = \mathcal{O}(p^2)$. Therefore, the time complexity of Algorithm 1 is $\mathcal{O}(3 * n * p^2) = \mathcal{O}(np^2)$
- step 7: $\mathcal{O}(n \log n)$
- step 8: $\mathcal{O}(np^2)$
- step 9: $\mathcal{O}(m * n)$
- step 10: $\mathcal{O}(n)$
- step 11: $\mathcal{O}(n \log n)$

The total time complexity is $\mathcal{O}(n * m) + \mathcal{O}(n) + \mathcal{O}(1) + \mathcal{O}(np^2) + \mathcal{O}(n \log n) = \mathcal{O}(n * m) + \mathcal{O}(np^2) + \mathcal{O}(n \log n)$. Therefore, the time complexity of the proposed CTLM recommender algorithm is $\mathcal{O}(n * (m + p^2 + \log n))$. Based on the experimental results in subsection 5.3.2.2, we select the best baseline model ITB to compare with the proposed CTLM recommender algorithm. The proposed taxonomy-based ITB approach developed by Weng et al. and based on Ziegler's topic weight method was adapted in order to generate the baseline model ITB. We generated the ITB's algorithm according to the formulae and details provided in subsection 5.2.2. For more information on Ziegler's taxonomy vector construction algorithm, please refer to [111, 121]. The time complexity of the baseline model ITB recommender algorithm is $\mathcal{O}(mp^2) + \mathcal{O}(n * (n + p^2 + \log n))$.

- The Runtime Comparison

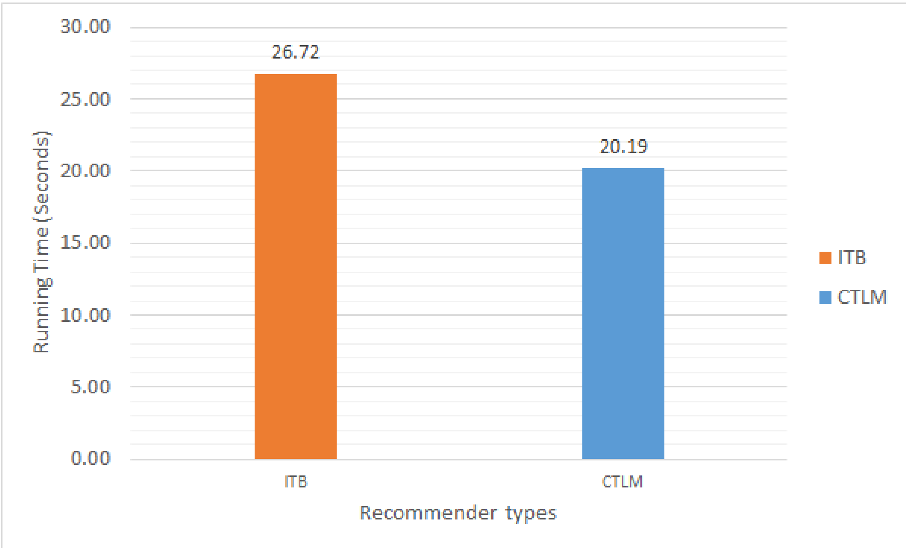

Figure 5.7: The runtime comparison of CTLM and ITB

In addition, we conduct the efficiency comparison experiments. The proposed CTLM approach and the baseline model ITB were compared by their runtime on Dataset D1. They were run on a laptop with a 4 GB memory, Core i7 CPU. The runtime comparison of the proposed CTLM approach and the baseline model ITB are shown in Figure 5.7 according to the average seconds spent per recommendation. The results suggest that the efficiency of the baseline model ITB was slightly worse than the proposed CTLM. While the proposed CTLM approach took around 20.19 seconds, the ITB took around 26.72 seconds to process the same data. This is because the ITB is computation expensive; it needs to transform all the users and items into high dimension taxonomy vectors in order to compute the similarities between a user and user and between an item and user. ITB makes the prediction or recommendation by relying on these two similarities, therefore its efficiency might be affected by this circumstance. The results suggest that the proposed CTLM approach can be used for large-scale recommender systems.

5.4 CHAPTER SUMMARY

This chapter evaluated the effectiveness of using concept taxonomy hierarchies as a part of the recommendation approach. The proposed approaches were evaluated by using two real-world datasets on books and music that were collected from the Book-Crossing community and Amazon's website. This thesis proposed three recommendation approaches to improve the performance of recommendation making and alleviate the cold-start problem. The experimental results were compared to state-of-the-art baseline approaches and other approaches involved in the experiments. The experimental results showed that the proposed PopCs and CTLM approach-based taxonomies can significantly improve the accuracy of the item recommendations. The proposed method for using the probability of concept and the user-item concept hierarchies' similarities, along with the use of concept taxonomy, has yielded a significant improvement in the recommendation making in both normal and cold-start environments.

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 SUMMARY OF THE RESEARCH

The main objective of this thesis is to enhance approaches for making personalised recommendations. This thesis also provides an alternative way to alleviate the recommendation problems with cold-starts and the issue of users having uncertain information needs. This research explores and exploits the relationship between users and items, according to a set of taxonomic categories or concepts utilised by users and relating to items. The main idea is to understand how items are likely to generate what users want, according to their relationship in concept taxonomy. This thesis proposes the user profiling and item representation approaches based on the concept hierarchy model.

This thesis has proposed a new structure to use taxonomy information to describe uncertain knowledge regarding user information needs in a two-dimensional hierarchy. The concept hierarchy model has been proposed in order to acquire new users' preferences when there is limited information about these users. The purpose was to solve the cold-start problem and users being confronted with uncertainties regarding their information needs. The new concept-weighting method, using rating data instead of considering the hierarchical relationship between each concept taxonomy and the frequency of each taxonomic concept in item taxonomic descriptors, was proposed to measure the weight of taxonomic concepts. Instead of representing user preferences by using rating data, the user preferences and the item descriptions can be represented via a user-concept hierarchy and an item-concept hierarchy, respectively. This model was utilised to generate the active user profile and item representation based on concept hierarchy.

Furthermore, we have developed effective recommendation-making approaches to improve the quality of item recommendations. The combination of both the user profiles and the item representation, based on the concept hierarchy, were applied to recommendation-making approaches. In this thesis, we use the following factors for making recommendations:

- **Item popularity:** the users' opinion about items by means of ratings. The ratings a user assigns to the item correspond to the user's overall preferences in regard to the items.
- **The user-item concept hierarchy similarities:** the similarities between user and item based on concept hierarchies.
- **The concept probability:** the probability of the concept in regard to an item that is relevant to the user's concept preferences.

The linear item popularity combined with the user-item concept hierarchy similarities made up the *PopCs* recommendation approach. In addition, the adaptation of the language model incorporating the item popularity, the user and item concept hierarchy similarities and the probability of the concept were utilised to generate the *CTLM* recommendation approach.

The experimental results were evaluated on the two real world data sets collected from the BookCrossing community and Amazon website. This thesis proposes two recommended approaches to improve the performance of the recommendation making as well as to reduce the new user problem. The experimental results were compared with the state-of-the-art baseline approaches and other experimental approaches.

It was found that the proposed PopCs and CTLM recommendation approaches can effectively improve recommendation accuracy and quality. In addition, the proposed user-item concept hierarchy similarities and the probability of concept approaches based on the use of concept taxonomy outperformed the approach based on item popularity, and they can provide significant improvement in recommendation making in both normal and cold-start environments.

6.2 CONTRIBUTIONS

This thesis makes a significant contribution toward enhancing personalised recommendations. It provides the most effective approaches to obtain user preferences and recommendation approaches based on taxonomy information. It also contributes to more accurate user profiling through the alternative solution to the cold-start problem and when users being confronted with uncertain information needs.

- **This thesis contributes to user profiling and solving recommendation problems**

A user's personal data reflects their personal interests or preferences. One of the challenging tasks in building a personalised recommendation is acquiring the user's preferences (or needs) when there is insufficient information about the user's preferences. Another issue is users who are uncertain about what they want/need or do not know how to describe exactly what they want. These issues can negatively affect the recommender system performance and make it difficult to profile users accurately and make quality recommendations.

To overcome these problems, this thesis proposes the concept hierarchy model to obtain information about user's interest in certain items, according to taxonomy information that is utilised by the user and relates to the item. Item taxonomy information is a set of categories or topics that can be used to classify and describe items in a hierarchical structure. The structural information of the concept taxonomy in its hierarchical structure can provide the best explanation of a variety of concepts about an item, from general to specific concepts. The advantage of the taxonomy tree structure is that it provides a new structure via which to obtain the user preferences in a two-dimensional hierarchy. The method can be used to generate information about new users faster than a rating method would allow. This is a new contribution to the usage of taxonomy to model users' preferences.

The model does not require the users' rating data to measure the weights of taxonomic concepts. The concept weight can be computed by using only the taxonomy tree structure and the concept frequency in the item taxonomy descriptors. This will reduce the impact of the cold-start problem. The method is applicable in cold-start situations where there are few ratings available in the system. Instead of representing the users by using ratings, the users preferences can be characterised using the concept hierarchy. It contributes to a more accurate and comprehensive user profiling database and provides an alternative way to reduce the impact of the cold-start problem.

In this thesis, a novel way of exploiting explicit feedback to obtain user information needs from users integrating with item taxonomy information is proposed, rather than using only item ratings. Users can easily use the different concepts based on the taxonomy to describe what they want. It also contributes to providing a solution to the problem when users are confronted with uncertain information needs and creates more accurate user profiling. Moreover, the proposed model can be applied to use implicit feedback information such as the content of the items or taxonomic concepts of the item that the users clicked, bought, tagged or rated. The proposed concept hierarchy model based on taxonomy can be used as a quality information source to profile users and enhance personalised recommendations. It can also be applied to other application areas such as user modelling and personalisation searching.

- **This thesis contributes to effectively using taxonomy information as a user information source in recommender systems.**

Besides using the rating data to determine the users' preferences in regards to the item, taxonomy information provides another way to determine the level of interest of the user in the item. Item taxonomy information is a set of categories or topics that can be used to classify and describe items in a hierarchical structure. These features are intrinsic to the item and as such, they do not depend on historical preferences. In addition to profiling the users' preferences using taxonomy information, this thesis also propose approaches to describe and represent items based on taxonomy information.

Since taxonomic descriptors that characterise the users' item preferences or interests can be generated from an item, and users are likely to be interested in a corresponding set of taxonomic concepts associated with the item, this thesis exploits the taxonomy information of a few items to investigate the relationship between user and item according to the set of taxonomic categories utilised by the user and relating to the item. It contributes to better utilisation of taxonomy information into useful knowledge in order to represent the user and item in a meaningful way as part of the concept hierarchy. It also provides a better understanding that, alongside ratings, reflects users' opinions of items, and features of items that can also reflect users' opinions of an item. Therefore, user preferences and item descriptions can be characterised by using the taxonomic concept, instead of representing users' preferences by using the item ratings. The proposed user profiles and item descriptions based on taxonomy information can be applied to form the neighbourhood for the target user or new user.

The recommendation approaches incorporate the opinion of users about the item by ratings and the relationship between user and item according to the use of taxonomy information to generate the item recommendations. This thesis contributes to better usage of taxonomy information to improve the accuracy of user profiling and item recommendations when there is limited information available about user. It makes significant improvements to the performance of recommender systems.

- **This thesis contributes to enhance the approaches for making item recommendations.**

In real world applications, recommender systems suffer from having insufficient personal data to generate quality personalised recommendations. Many studies have developed new techniques to make better recommendations with the information resources available. This thesis utilises, and explores how to enhance item recommendation-making based on item taxonomy information. In some studies, taxonomy information was integrated with the ratings data to increase the quality of item recommendation. The proposed recommendation approaches in this thesis can effectively improve the accuracy of item recommendation.

Recently, Information Retrieval (IR) techniques have been exploited in several studies in the field of recommender systems. In this thesis, the language model is applied to understand how items can be used to generate a concept of what a user wants. It contributes effectively to integrating the language model and the relationship between user and item according to taxonomy information to improve the accuracy of item recommendations. The users' opinions of items in terms of item popularity incorporate the relationship between a user's concept hierarchy and other items' concept hierarchies based on concept taxonomy; they are utilised as the foundation for designing the recommendation approaches.

This thesis contributes to better usage of item taxonomy information. It contributes to the new item recommendation functions. It can provide significant improvement in the quality of recommendations in both normal and cold-start environments. Therefore, taxonomy information can be used as a quality information source to assist in item recommendation. It can be applied to recommender system applications where item taxonomy is available. Moreover, the approaches proposed in this thesis can be used to further improve the accuracy of recommendations in situations where the system has few ratings available from which to base new recommendations. It can be applied to recommender system applications to improve personalised item recommendations.

6.3 LIMITATIONS AND FUTURE WORK

The following sections will discuss about limitations of this research and the possible future expansion of the study.

6.3.1 Limitations

This limitations of this study include those discussed below:

1.) The recommendation approaches proposed in this thesis require additional information to that which may readily be available; that is, item taxonomy information (e.g., categories of product). Therefore, the proposed approaches are restricted to systems or the situations where the required item taxonomy information is available. However, we believe that on a conceptual level, this limitation of the study is not critical. The issue is important in the case of real-world implementation with real datasets. This is a possible future expansion of the study.

2.) The thesis proposes approaches to improve the recommendation performance in both normal and cold-start environments, but did not analyse or take into account the sparsity situation. This is an area which could be investigated in future work.

6.3.2 Future work

During the research, there were many potentially interesting ideas and topics raised that are relevant to this research. This section briefly discusses possible directions for future study.

This research could extend in many directions. The main idea of this thesis is to make use of the relationship between users and items according to the taxonomy information that is utilised by users. It provides potential improvement in recommendation making. This relationship should be explored further in order to improve recommendation performances and user profiling. Therefore, plans have been made to extend the work of this thesis to further study this relationship together with Information Retrieval adaptation techniques to produce other algorithms for enhancing the item recommendations.

In addition, the basic idea of Knowledge Infusion (KI) could be used to further improve the performance of recommender systems. Instead of solely using taxonomy information to determine the users' preferences, the research could extend to integrate with Web 2.0 or 3.0 information such as folksonomy, blogs, and Semantic Web to generate new knowledge. It could be used to further improve user profiling accuracy and make quality recommendations.

There are also future plans to explore a new recommendation-making approach to further improve the performance of recommendations. The solutions to the cold-start will be further investigated. However, using taxonomy information with user implicit feedback to improve the solution to the cold-start problem and the quality of recommendations will require further research.

Most recommender systems focused on improving the effectiveness of the system. However, since the information overload and the number of features in recommender systems is large, it is essential that the algorithms be efficient. Therefore, the high performance of the new algorithms and the solutions to the scalability problem will require further research.

Appendix A

EXAMPLE CATEGORIES TAXONOMY DATA SOURCE

- Categories Taxonomy Information of the book

Data format

```
Id:   15
ASIN: 1559362022
  title: Wake Up and Smell the Coffee
  group: Book
  salesrank: 518927
  similar: 5  1559360968  1559361247  1559360828  1559361018  0743214552
  categories: 3
   |Books[283155]|Subjects[1000]|Literature & Fiction[17]|Drama[2159]|United States[2160
   |Books[283155]|Subjects[1000]|Arts & Photography[1]|Performing Arts[521000]|Theater[2
   |Books[283155]|Subjects[1000]|Literature & Fiction[17]|Authors, A-Z[70021]|( B )[7002
  reviews: total: 8  downloaded: 8  avg rating: 4
    2002-5-13  cutomer: A2IGOA66Y6O8TQ  rating: 5  votes:   3  helpful:   2
    2002-6-17  cutomer: A2OIN4AUH84KNE  rating: 5  votes:   2  helpful:   1
    2003-1-2  cutomer: A2HN382JNT1CIU  rating: 1  votes:   6  helpful:   1
    2003-6-7  cutomer: A2FDJ79LDU4O18  rating: 4  votes:   1  helpful:   1
    2003-6-27  cutomer: A39QMV9ZKRJXO5  rating: 4  votes:   1  helpful:   1
    2004-2-17  cutomer:  AUUVMSTQ1TXDI  rating: 1  votes:   2  helpful:   0
    2004-2-24  cutomer: A2C5K0QTLL9UAT  rating: 5  votes:   2  helpful:   2
    2004-10-13  cutomer:  A5XYF0Z3UH4HB  rating: 5  votes:   1  helpful:   1
```

Data format:

- **Id:** Product id (number 0, ..., 548551)
- **ASIN:** Amazon Standard Identification Number
- **title:** Name/title of the product
- **group:** Product group (Book, DVD, Video or Music)
- **salesrank:** Amazon Salesrank
- **similar:** ASINs of co-purchased products (people who buy X also buy Y)
- **categories:** Location in product category hierarchy to which the product belongs (separated by |, category id in [])
- **reviews:** Product review information: time, user id, rating, total number of votes on the review, total number of helpfulness votes (how many people found the review to be helpful)

Figure A.1: The example data format of categories taxonomy information

Appendix B

THE GENERAL STRUCTURE AND COMPONENTS OF THE SYSTEM

Figure B.1 presents the structure and components used for modelling the user profiles for the recommender systems in this study. The recommender systems are made up of three main components: 1) training flow; 2) testing flow; and 3) recommendation flow. The training flow starts with the item data and user purchase data. The item taxonomy information for both items and user purchase data were retrieved and transformed to profile the users and items in the concept hierarchy model. The testing flow starts with the test users' data, and the active users' concepts of interest were transformed to generate the active user concept hierarchy. This was then used to identify the similarity between user and item and between two users based on the two concept profile vectors. The user-to-item concept hierarchy similarities and the user-to-user concept hierarchy similarities were exploited to generate the three recommendation approaches. To enhance recommendation making, item popularity and probability of the concept are incorporated in recommendation approaches. The recommendation flow starts with the three recommendation approaches, and each recommendation approach proposed specific items to the target users according to the prediction score.

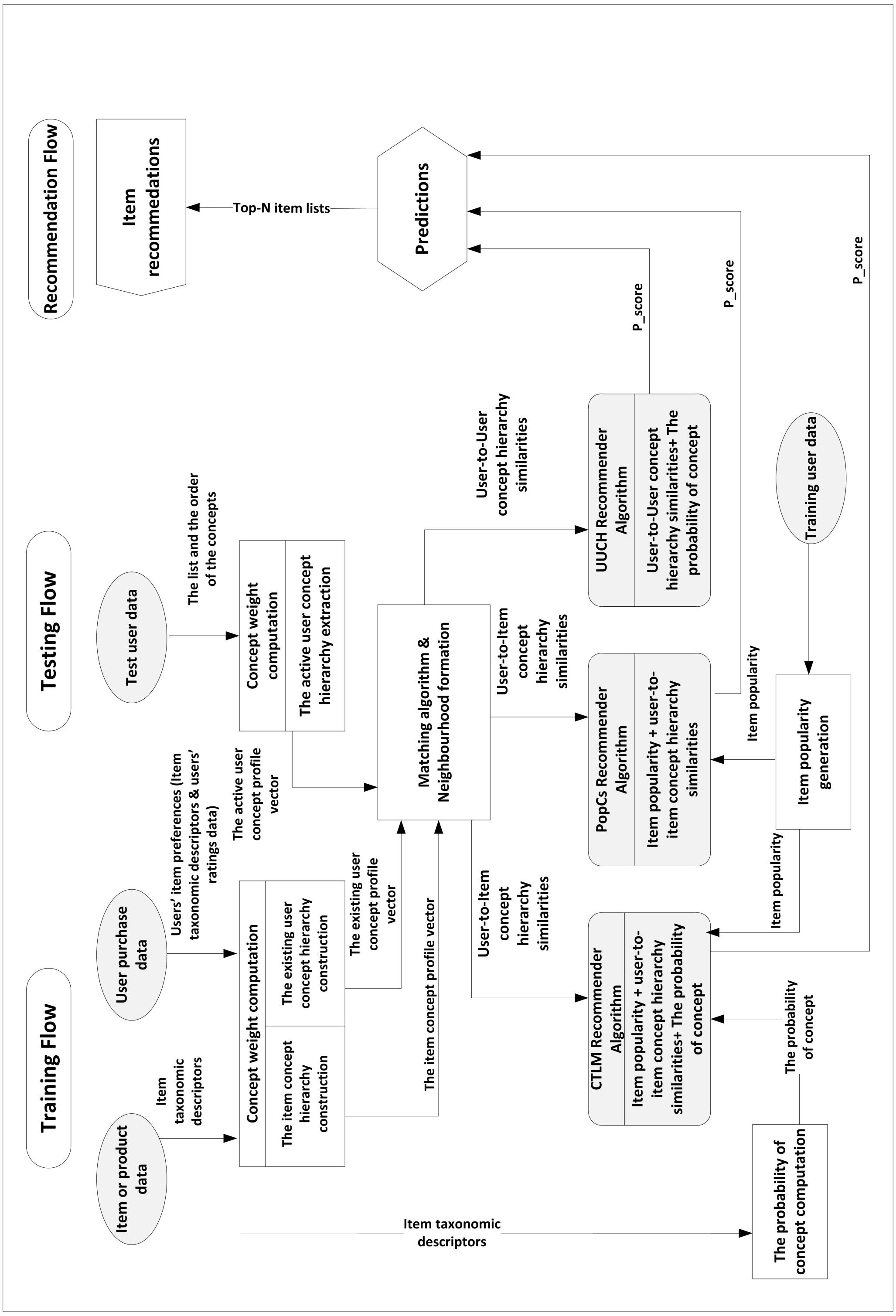

Figure B.1: The structure and components of modelling user profiles for recommender systems of this thesis

Appendix C

EXAMPLE OF BOOK DATABASE STRUCTURE

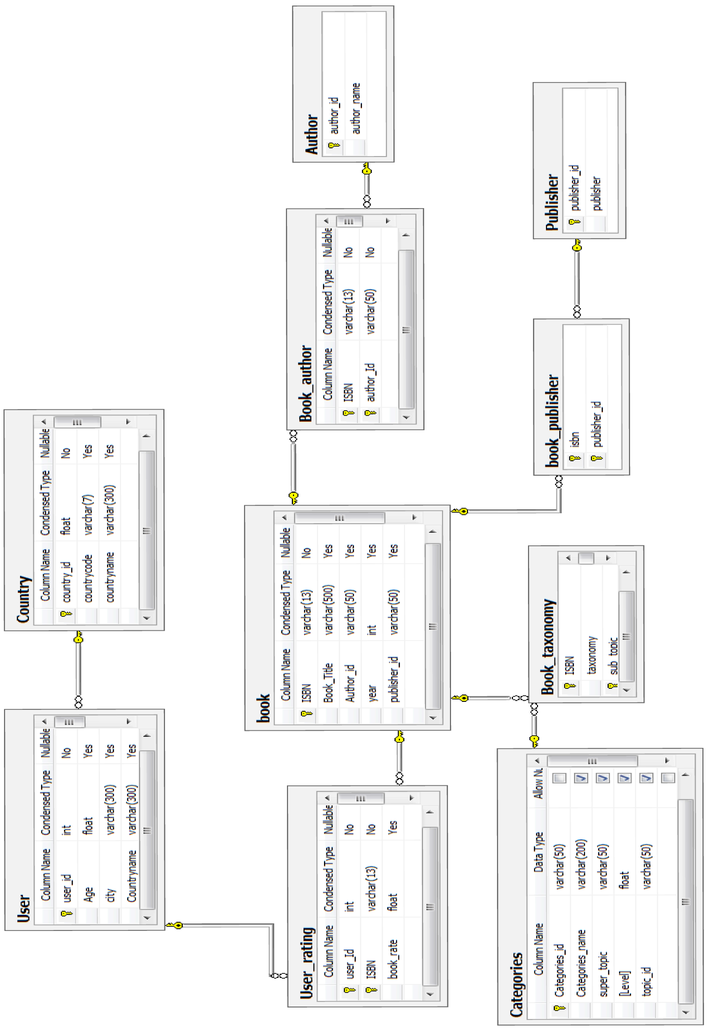

Figure C.1: The example of book database structure

REFERENCES

[1] Z. Abbassi, S. Amer-Yahia, L. V. Lakshmanan, S. Vassilvitskii, and C. Yu. Getting recommender systems to think outside the box. In *Proceedings of the third ACM conference on Recommender systems*, pages 285–288. ACM, 2009.

[2] G. Adomavicius and Y. Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *Knowledge and Data Engineering, IEEE Transactions on*, 24(5):896–911, May 2012.

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

[4] A. Ahmed, B. Kanagal, S. Pandey, V. Josifovski, L. G. Pueyo, and J. Yuan. Latent factor models with additive and hierarchically-smoothed user preferences. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 385–394, New York, NY, USA, 2013.

[5] B. Amini, R. Ibrahim, and M. S. Othman. Discovering the impact of knowledge in recommender systems: A comparative study. *arXiv preprint arXiv:1109.0166*, 2011.

[6] A. B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-López, F. A. Mikic-Fonte, and A. Peleteiro. A hybrid content-based and item-based collaborative filtering approach to recommend tv programs enhanced with singular value decomposition. *Information Sciences*, 180(22):4290–4311, 2010.

[7] R. M. Bell and Y. Koren. Improved neighborhood-based collaborative filtering. In *KDD Cup and Workshop at the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. sn, 2007.

[8] R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 43–52. IEEE, 2007.

[9] A. Bellogín, J. Wang, and P. Castells. Structured collaborative filtering. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2257–2260. ACM, 2011.

[10] P. K. Bhowmick, S. Sarkar, and A. Basu. Ontology based user modeling for personalized information access. *IJCSA*, 7(1):1–22, 2010.

[11] P. E. Black. big-o notation. *Dictionary of Algorithms and Data Structures*, 2007, 2007.

[12] D. Bollen, B. P. Knijnenburg, M. C. Willemsen, and M. Graus. Understanding choice overload in recommender systems. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 63–70, New York, NY, USA, 2010.

[13] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, UAI'98, pages 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[14] R. Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[15] R. Burke. Hybrid web recommender systems. In *The adaptive web*, pages 377–408. Springer, 2007.

[16] F. Cacheda, V. Carneiro, D. Fernández, and V. Formoso. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Trans. Web*, 5(1):2:1–2:33, Feb. 2011.

[17] Y. Cao and Y. Li. An intelligent fuzzy-based recommendation system for consumer electronic products. *Expert Systems with Applications*, 33(1):230–240, 2007.

[18] J. Chen, J. Huang, and H. Min. Easy recommendation based on probability model. In *Semantics, Knowledge and Grid, 2008. SKG '08. Fourth International Conference on*, pages 441–444, Dec 2008.

[19] Y. Chen, C. Wu, M. Xie, and X. Guo. Solving the sparsity problem in recommender systems using association retrieval. *Journal of computers*, 6(9):1896–1902, 2011.

[20] Y. H. Cho and J. K. Kim. Application of web usage mining and product taxonomy to collaborative recommendations in e-commerce. *Expert systems with Applications*, 26(2):233–246, 2004.

[21] E. Cox, M. O'Hagan, R. Taber, and M. O'Hagen. *The Fuzzy Systems Handkbook with Cdrom*. Academic Press, Inc., 1998.

[22] N. Craswell. Precision at n. In L. LIU and M. ZSU, editors, *Encyclopedia of Database Systems*, pages 2127–2128. Springer US, 2009.

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

[24] P. Cremonesi, R. Turrin, and F. Airoldi. Hybrid algorithms for recommending new items. In *Proceedings of the 2nd international workshop on information heterogeneity and fusion in recommender systems*, pages 33–40. ACM, 2011.

[25] W. B. Croft, S. Cronen-Townsend, and V. Lavrenko. Relevance feedback and personalization: A language modeling perspective. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, volume 3, page 13, 2001.

[26] W. B. Croft and J. Lafferty. *Language modeling for information retrieval*, volume 13. Springer Science & Business Media, 2003.

[27] M. Daoud, L. Tamine, M. Boughanem, and B. Chebaro. Learning implicit user interests using ontology and search history for personalization. In *Proceedings of the 2007 international conference on Web information systems engineering*, WISE'07, pages 325–336, Berlin, Heidelberg, 2007. Springer-Verlag.

[28] J. J. de Wit. Evaluating recommender systems – an evaluation framework to predict user satisfaction for recommender systems in an electronic programme guide context. Master's thesis, University of Twente, May 2008.

[29] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, Jan. 2004.

[30] C. Desrosiers and G. Karypis. A novel approach to compute similarities and its application to item recommendation. In *PRICAI 2010: Trends in Artificial Intelligence*, pages 39–51. Springer, 2010.

[31] E. Djuana, Y. Xu, Y. Li, and A. Josang. Ontology learning from user tagging for tag recommendation making. In *Proceedings of the 2011 IEEE/WIC/ACM*

International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03, pages 310–313. IEEE Computer Society, 2011.

[32] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 176–185, Dec 2010.

[33] Z. Gantner, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme. Bayesian personalized ranking for non-uniformly sampled items. *JMLR*, 18:231–247, 2012.

[34] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli. User profiles for personalized information access. In *The adaptive web*, pages 54–89. Springer, 2007.

[35] A. Gunawardana and C. Meek. Tied boltzmann machines for cold start recommendations. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 19–26. ACM, 2008.

[36] A. Gunawardana and G. Shani. A survey of accuracy evaluation metrics of recommendation tasks. *The Journal of Machine Learning Research*, 10:2935–2962, 2009.

[37] G. Guo. Resolving data sparsity and cold start in recommender systems. In *Proceedings of the 20th International Conference on User Modeling, Adaptation, and Personalization*, UMAP'12, pages 361–364, Berlin, Heidelberg, 2012. Springer-Verlag.

[38] G. Guo. Improving the performance of recommender systems by alleviating the data sparsity and cold start problems. In *Proceedings of the Twenty-Third*

international joint conference on Artificial Intelligence, pages 3217–3218. AAAI Press, 2013.

[39] M. Hahsler. Developing and testing top-n recommendation algorithms for 0-1 data using recommenderlab. *NSF Industry University Cooperative Research Center for Net-Centric Software and System*, 2011.

[40] Y. Hang, C. Guiran, and W. Xingwei. A cold-start recommendation algorithm based on new user's implicit information and multi-attribute rating matrix. In *Hybrid Intelligent Systems, 2009. HIS'09. Ninth International Conference on*, volume 2, pages 353–358. IEEE, 2009.

[41] D. Harman. *Information Retrieval Evaluation*. Morgan & Claypool Publishers, 1st edition, 2011.

[42] J. Herlocker, J. A. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information retrieval*, 5(4):287–310, 2002.

[43] J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, CSCW '00, pages 241–250, New York, NY, USA, 2000. ACM.

[44] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[45] W. Hong, L. Li, and T. Li. Product recommendation with temporal dynamics. *Expert Systems with Applications*, 39(16):12398–12406, 2012.

[46] L.-p. Hung. A personalized recommendation system based on product taxonomy for one-to-one marketing online. *Expert Syst. Appl.*, 29(2):383–392, Aug. 2005.

[47] N. Hurley and M. Zhang. Novelty and diversity in top-n recommendation – analysis and evaluation. *ACM Trans. Internet Technol.*, 10(4):14:1–14:30, Mar. 2011.

[48] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010.

[49] G. Jawaheer, P. Weller, and P. Kostkova. Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Trans. Interact. Intell. Syst.*, 4(2):8:1–8:26, June 2014.

[50] B. Kanagal, A. Ahmed, S. Pandey, V. Josifovski, J. Yuan, and L. G. Pueyo. Supercharging recommender systems using taxonomies for learning user purchase behavior. *Proceedings of the VLDB Endowment*, 5(10):956–967, 2012.

[51] G. Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the tenth international conference on Information and knowledge management*, CIKM '01, pages 247–254, New York, NY, USA, 2001. ACM.

[52] H.-R. Kim and P. K. Chan. Learning implicit user interest hierarchy for context in personalization. *Applied Intelligence*, 28(2):153–166, 2008.

[53] Y. S. Kim. Recommender system based on product taxonomy in e-commerce sites. *Journal of Information Science and Engineering*, 29(1):63–78, 2013.

[54] N. Koenigstein, G. Dror, and Y. Koren. Yahoo! music recommendations: Modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 165–172, New York, NY, USA, 2011.

[55] Y. Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 426–434, New York, NY, USA, 2008.

[56] Y. Koren. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(1):1, 2010.

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

[58] R. Lau, C. Lai, and Y. Li. Leveraging the web context for context-sensitive opinion mining. In *Proceedings of the 2009 IEEE International Conference on Computer Science and Information Technology*, pages 467–471. IEEE, 2009.

[59] R. Lau, Y. Li, S. T. Wu, and X. Zhou. Sequential Pattern Mining and Nonmonotonic Reasoning for Intelligent Information Agents. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(4):773–790, 2007.

[60] R. Y. Lau, C. Li, and S. S. Liao. Social analytics: Learning fuzzy product ontologies for aspect-oriented sentiment analysis. *Decision Support Systems*, 2014. Forthcoming.

[61] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 120–127, New York, NY, USA, 2001. ACM.

[62] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1), May 2007.

[63] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[64] S. Li, G. Wu, and X. Hu. Hierarchical user interest modeling for chinese web pages. In *Proceedings of the Third International Conference on Internet Multimedia Computing and Service*, ICIMCS '11, pages 164–169, New York, NY, USA, 2011. ACM.

[65] X. Li and S.-K. Chang. A personalized e-learning system based on user profile constructed using information fusion. In *DMS*, pages 109–114. Citeseer, 2005.

[66] X. Li and W. Croft. Improving the robustness of relevance-based language models. Technical report, Technical Report IR-401, Center for Intelligent Information Retrieval, University of Massachusetts, 2005.

[67] H. Liang, Y. Xu, Y. Li, and R. Nayak. Personalized recommender system based on item taxonomy and folksonomy. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM'10, pages 1641–1644. ACM, 2010.

[68] X. Liu and B. Croft. Cluster-based retrieval using language models. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 186–193, 2004.

[69] P. Lops, M. de Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, pages 73–105. Springer, 2011.

[70] P. Lops, M. de Gemmis, G. Semeraro, C. Musto, F. Narducci, and M. Bux. A semantic content-based recommender system integrating folksonomies for

personalized access. In *Web Personalization in Intelligent Environments*, pages 27–47. Springer, 2009.

[71] V. Maidel, P. Shoval, B. Shapira, and M. Taieb-Maimon. Evaluation of an ontology-content based filtering method for a personalized newspaper. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 91–98. ACM, 2008.

[72] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[73] O. Matos-Junior, N. Ziviani, F. C. Botelho, M. Cristo, A. Lacerda, and A. S. da Silva. Using taxonomies for product recommendation. *JIDM*, 3(2):85–100, 2012.

[74] M. R. McLaughlin and J. L. Herlocker. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 329–336. ACM, 2004.

[75] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Eighteenth national conference on Artificial intelligence*, pages 187–192, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[76] P. Melville and V. Sindhwani. Recommender systems. In *Encyclopedia of machine learning*, pages 829–838. Springer, 2010.

[77] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, 22(1):54–88, Jan. 2004.

[78] G. A. Miller, B. R., C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database. *Journal of Lexicography*, 3(4):234–244, 1990.

[79] A. Mnih. Taxonomy-informed latent factor models for implicit feedback. *In KDD Cup, KDD 2011*, 2011.

[80] M. Montaner, B. Lpez, and J. de la Rosa. A taxonomy of recommender agents on the internet. *Artificial Intelligence Review*, 19(4):285–330, 2003.

[81] W. Nadee, Y. Li, and Y. Xu. Acquiring user information needs for recommender systems. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences*, volume 3, pages 5–8, Nov 2013.

[82] N. Nanas, A. De Roeck, and M. Vavalis. *What happened to content-based information filtering?* Springer, 2009.

[83] N. Nanas, V. Uren, and A. de Roeck. Building and applying a concept hierarchy representation of a user profile. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 198–204. ACM, 2003.

[84] J.-Y. Nie, G. Cao, and J. Bai. Inferential language models for information retrieval. *ACM Transactions on Asian Language Information Processing*, 5(4):296–322, Dec. 2006.

[85] M. Papagelis, D. Plexousakis, and T. Kutsuras. Alleviating the sparsity problem of collaborative filtering using trust inferences. In *Trust management*, pages 224–239. Springer, 2005.

[86] J. Parapar, A. Bellogín, P. Castells, and Á. Barreiro. Relevance-based language modelling for recommender systems. *Information Processing & Management*, 49(4):966–980, 2013.

[87] D. Pathak, S. Matharia, and C. Murthy. Nova: Hybrid book recommendation engine. In *Advance Computing Conference (IACC), 2013 IEEE 3rd International*, pages 977–982, Feb 2013.

[88] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[89] I. Pilászy and D. Tikk. Recommending new movies: even a few ratings are more valuable than metadata. In *Proceedings of the third ACM conference on Recommender systems*, pages 93–100. ACM, 2009.

[90] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, Melbourne, Australia, 1998.

[91] S. Rendle and C. Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 273–282, New York, NY, USA, 2014. ACM.

[92] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press, 2009.

[93] F. Ricci, L. Rokach, and B. Shapira. *Recommender Systems Handbook*, chapter Introduction to Recommender Systems Handbook, pages 1–35. Springer US, Boston, MA, 2011.

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

[95] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.

[96] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.

[97] G. Schröder, M. Thiele, and W. Lehner. Setting goals and choosing metrics for recommender system evaluations. In *UCERSTI2 Workshop at the 5th ACM Conference on Recommender Systems, Chicago, USA*, volume 23, 2011.

[98] J. Sean, M. McNee, and J. Konstan. Accurate is not always good: How accuracy metrics have hurt recommender systems. *extended abstracts on Human factors in computing systems (CHI06) p*, pages 1097–1101, 2006.

[99] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver. Tfmap: Optimizing map for top-n context-aware recommendation. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 155–164. ACM, 2012.

[100] S. Singh, M. Shepherd, J. Duffy, C. Watters, et al. An adaptive user profile for

filtering news based on a user interest hierarchy. *Proceedings of the American Society for Information Science and Technology*, 43(1):1–21, 2006.

[101] H. Steck. Item popularity and recommendation accuracy. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 125–132, New York, NY, USA, 2011.

[102] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:4, 2009.

[103] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Matrix factorization and neighbor based algorithms for the netflix prize problem. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 267–274, New York, NY, USA, 2008.

[104] J. Tang, L. Yao, D. Zhang, and J. Zhang. A combination approach to web user profiling. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(1):2, 2010.

[105] X. Tao and Y. Li. A user profiles acquiring approach using pseudo-relevance feedback. In *Rough Sets and Knowledge Technology*, pages 658–665. Springer, 2009.

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

[107] S. Vargas. Novelty and diversity enhancement and evaluation in recommender systems and information retrieval. In *Proceedings of the 37th International ACM SIGIR Conference on Research; Development in Information Retrieval*, SIGIR '14, pages 1281–1281, New York, NY, USA, 2014.

[108] J. Wang. Language models of collaborative filtering. In *Information Retrieval Technology*, pages 218–229. Springer, 2009.

[109] J. Wang, A. P. de Vries, and M. J. Reinders. A user-item relevance model for log-based collaborative filtering. In *Advances in Information Retrieval*, pages 37–48. Springer, 2006.

[110] Y. Z. Wei, L. Moreau, and N. R. Jennings. A market-based approach to recommender systems. *ACM Transactions on Information Systems*, 23:2005, 2005.

[111] L. T. Weng, Y. Xu, Y. Li, and R. Nayak. Exploiting item taxonomy for solving cold-start problem in recommendation making. In *Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on*, volume 2, pages 113–120. IEEE, 2008.

[112] W. Woerndl, C. Schueller, and R. Wojtech. A hybrid recommender system for context-aware recommendations of mobile applications. In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, pages 871–878, April 2007.

[113] B. Xu, J. Bu, C. Chen, and D. Cai. An exploration of improving collaborative recommender systems via user-item subgroups. In *Proceedings of the 21st international conference on World Wide Web*, pages 21–30. ACM, 2012.

[114] A. A. Yeung. Matrix factorization: A simple tutorial and implementation in python. http://www.quuxlabs.com/blog/2010/09/matrix-factorization-a-simple-tutorial-and-implementation-in-python/, 16 September 2010.

[115] J. Yu, F. Liu, and H. Zhao. Building user profile based on concept and relation for web personalized services. In *International Conference on Innovation and Information Management*. Citeseer, 2012.

[116] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22:179–214, 2004.

[117] Y. Zhang. Exploiting latent information in recommender systems: A thesis submitted to the victoria university of wellington in fulfilment of the requirements for the degree of doctor of philosophy in computer science. page 230, 2012.

[118] Y. Zhang, A. Ahmed, V. Josifovski, and A. Smola. Taxonomy discovery for personalized recommendation. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 243–252, New York, NY, USA, 2014.

[119] Y. Zhang and J. Koren. Efficient bayesian hierarchical user modeling for recommendation system. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 47–54, New York, NY, USA, 2007.

[120] C.-N. Ziegler, G. Lausen, and J. A. Konstan. On exploiting classification taxonomies in recommender systems. *AI Commun.*, 21(2-3):97–125, 2008.

[121] C.-N. Ziegler, G. Lausen, and L. Schmidt-Thieme. Taxonomy-driven computation of product recommendations. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM'04, pages 406–415, 2004.

[122] C.-N. Ziegler, S. M. McNee, J. o. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th ACM international conference on World Wide Web*, pages 22–32. ACM, 2005.

[123] C.-N. Ziegler, L. Schmidt-Thieme, and G. Lausen. Exploiting semantic product descriptions for recommender systems. In *Proceedings of the 2nd ACM SIGIR Semantic Web and Information Retrieval Workshop*, Sheffield, UK, July 2004.