Semantically Enhanced Cross-Domain Recommender Systems

Oluseyi Olarewaju

A thesis submitted in partial fulfilment of the requirements of the University of Brighton for the degree of Doctor of Philosophy

November 2018

I would like to dedicate this thesis to my parents and my wife for their unrelenting love…

Declaration

I declare that the research contained in this thesis, unless otherwise formally indicated within the text, is the original work of the author. The thesis has not been previously submitted to this or any other university for a degree, and does not incorporate any material already submitted for a degree.

Oluseyi Olarewaju

November 2018

Acknowledgements

This thesis is the result of activities carried out during my research years. During this period I received the support of many people. The following paragraphs are a token of my appreciation to everyone.

First and foremost, my sincere gratitude goes to my supervisor Dr Gulden Uchyigit for her advice and guidance through the years. Starting from my interview as a PhD applicant right till the final submission of my thesis, her contributions and knowledge on the topic have been invaluable. She encouraged me during the good times and not-so-good periods of my research work. All interactions with her gave me the assurance of being more than an academic student number. Her hard work and willingness to support me were decisive in the completion of this thesis, and I consider myself fortunate to have had her as my supervisor.

I thank Dr Andrew Fish for meeting and welcoming me at the first researcher development workshop. I learnt how to approach research problems from him and also how to clearly present ideas and thoughts in different types of paperwork. I benefited immensely from all the different tools and tips he handed to me all through my research.

My sincere appreciation goes to Dr John Taylor for agreeing to join the supervisory team and bringing on his understanding of complex mathematical problems. His simplified interpretations of difficult problems gave me a good foundation to build on and made many theories less challenging to understand.

Special thanks go to my sponsors by the persons of Prof. & Mrs V.O. Olarewaju for their financial contributions and prayers throughout my stay in the UK. Finally, I like to thank my wife Rita Pearl Ekesi-Olarewaju for her love, sacrifice, support and understanding throughout the research period. I appreciate her hard work and strength and for standing in the gap every time I got carried away by my research activities.

Abstract

Memory-based collaborative filtering use past activities of a group of similar users to recommend future preferences for a target user in the group. Recommender systems based on this type of technique are prone to errors when there are too few historic interactions (e.g. rating, likes, transaction-history, visit-frequencies) between users of the system. The sparsity in users' historic data render the memory-based algorithm less effective at finding similar users for the personalisation process.

In contrast, model-based collaborative filtering techniques such as matrix factorisation (MF) use predictive models. In single-domain recommender systems, one problem prevalent to all techniques is the cold start problem. A cold start situation happens when there is no historical information about new users or items that have just been introduced to the system.

Several recommender techniques have used semantic knowledge extracted from additional user and item information to build profiles that reflect otherwise implicit user preferences. This semantic representation of the user is then used to find other similar users and address sparsity and cold-start problems in single domain recommender systems. Recent attempts to resolve cold-start and sparsity problems are considering cross-domain collaborative filtering techniques. Cross-domain recommender systems exploit additional user and item information from domains that are unique but related to the target recommendation domain. Extending predictive models to include parameters that model the semantic similarity in user and item information across the domains constitute a genuine approach in cross-domain recommendation.

The contents of this thesis centre around the use of semantically enhanced cross-domain recommender systems as a solution to cold start and sparsity problems. The contributions to cross-domain recommender systems are in three folds. First, we investigate and analyse the performance of a cross-domain recommender model as we vary the size of intersecting user/item information from the target and auxiliary domains. Secondly, we proposed a predictive model that adds semantically related tags as additional parameters to a matrix factorisation model. Thirdly, we present a model that incorporates category similarity

into a POI ranking function as contextual information for improving the performance of multi-category POI recommenders.

In our investigation, we empirically evaluate the proposed models on datasets that are sourced from different domains, specifically movies, books and several POI categories. On the one hand, the results show that semantically enriching tags in cross-domain recommender models are possible without negatively impacting recommendation accuracy. On the other hand, cross-domain recommender models that are semantically enhanced with additional latent parameters are effective in cold start scenarios and reduce the effect of sparsity on recommendation accuracy.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Background and Motivation

A fast-paced digital era and constant exposure to numerous information about different items have made the process of choosing a few preferred ones more challenging for individuals. Uchyigit et al. (2007) defined Recommender systems as a system that guides the user in a personalised way through a set of items. Recommender systems perform the vital task of information filtering and attempt to suggest only items (movie, music, books, news, location) that the users may find interesting. In describing a generic information filtering model for recommender approaches, Oard (1997) highlighted three broad subtasks for its implementation. The first task is collecting different sources of information. The second task is filtering useful information from the sources and finally displaying the filtered information to users.

An extensive taxonomy of sources (Figure 1.1) was put together by Felfernig and Burke (2008) to show the spectrum of knowledge sources available for different recommender approaches. The taxonomy shows that there are three broad sources where the knowledge needed to generate recommendations can be derived. One source is from knowledge about the users of the system (i.e. Social knowledge). A second source is from knowledge about the specific user who is to receive recommendations. A third source broadly grouped as Content is from data about the items. Information from these knowledge sources has to be sufficient in quantity and quality for a recommendation technique to be successful at mimicking the natural choices of users. Recommendations generated for users by the system tend to be less accurate when information from these sources are scarce or unavailable.

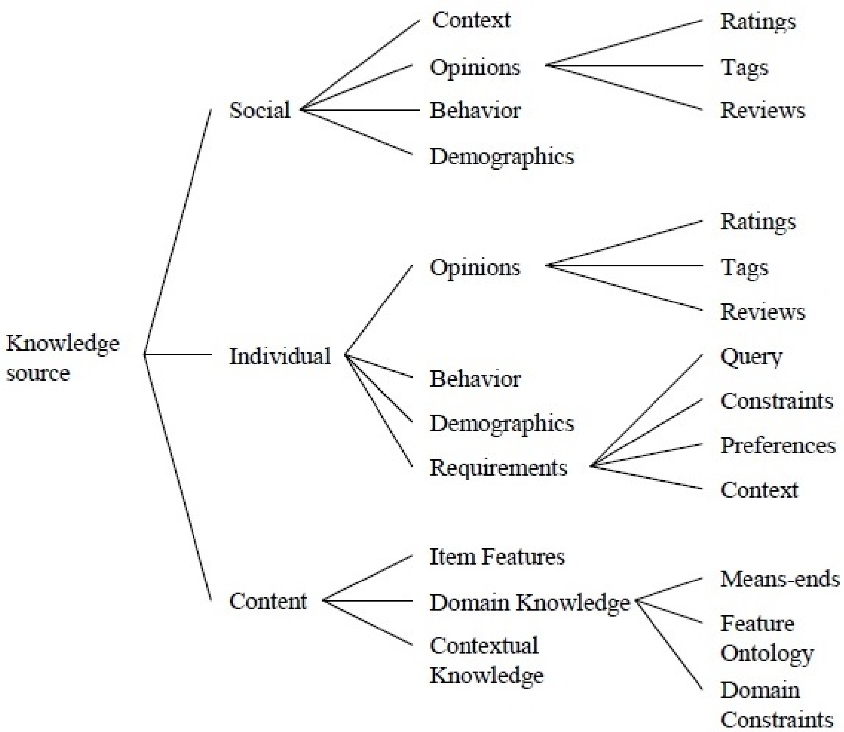


Fig. 1.1 Taxonomy of knowledge sources in recommendation (Felfernig and Burke (2008)).

The following are some of the problems and critical issues that arise as a result of insufficient knowledge sources. The effect of such shortfall in knowledge sources on the performance of the recommender system is highlighted in the corresponding recommender systems literature:

- The cold-start problem occurs when it is not possible to make recommendations due to an initial absence of individual or content knowledge sources. This problem occurs when a new user or item have just been introduced to the system. As reported in Adomavicius and Tuzhilin (2005) it becomes a challenge to find similar users/items because there is not enough information for comparison.

- The Sparsity problem occurs when the number of user feedback is small relative to the number of items. The general lack of user desire to give feedback creates a situation where there are far more items than user feedback. The resulting sparseness according to Ricci et al. (2011) makes finding users with similar interests in a conventional recommendation system challenging and collaboration difficult.

- Gray sheep are users whose opinions and interest are neither similar nor dissimilar with any group of users in the system. Claypool et al. (1999) pointed out that they rarely, if ever, receive accurate collaborative predictions, even after resolving the initial cold-start problem.

These problems have prompted researchers to raise questions on how knowledge sources within the domain of a recommender system can be extensively exploited to improve recommendation. More recently, techniques that use cross-domain recommender approaches have been emerging with similar research issues that merit investigation. Cross-domain recommendation approaches such as those proposed by Shi et al. (2011), Enrich et al. (2013) and Fernández-Tobías and Cantador (2014) can reuse data extracted from knowledge sources in an auxiliary domain for recommendation in a different one.

The rise in the growth and popularity of electronic commerce stores (e.g. Amazon, eBay and Alibaba) that offer the sale of items with diversity across numerous domains underscores the importance of inter/cross-domain recommendation systems. Recommendation technologies are progressively attracting the interest of new application domains as a valuable solution to increase system autonomy and efficiency (Felfernig et al. (2017)). The emerging Internet of Things (IoT) and IoT gateways cover such diverse domains. Gartner Inc., a leading information technology research and advisory company revealed in Heather (2017) that six billion connected things would be requesting support by 2018. According to Yao et al. (2016) the rich interactions and relations between users and things call for effective and efficient recommendation approaches to better meet users' interests and needs. The recommendation technique for things of interest can benefit from a cross-domain recommender approach that considers heterogeneous information sources.

Also, the recent surge in use of artificial intelligence in autonomous systems (e.g. driverless cars and unmanned aerial vehicles) has led to national responses from governments. A case in point is the United Kingdom's 2017 Robotics and Autonomous Systems Strategy in which the Government committed to investing an extra £2 billion a year in research and development (Robotics and Autonomous Systems 2017). At the centre of the autonomous systems are machine learning and natural language processing techniques that can learn parameters in an auxiliary domain and reuse the parameters in a target domain. Cross-domain recommendation approaches are a type of such transfer learning technique where models are learned in set-ups with different knowledge sources. According to Enrich et al. (2013) and Fernández-Tobías and Cantador (2014), they can address the cold-start and sparsity problems by utilising hidden features from the auxiliary domains.

1.2 Cross-domain Recommendation

The overwhelming number of choices that users have in selecting different brands of items have led to more competition among businesses. According to Fernández-Tobías et al. (2012), cross-selling approaches have been mainly proposed to provide recommendations in e-commerce sites, where they can increase customer satisfaction/loyalty and businesses profitability. A relationship between the domains involved in the process is generally required in order for the cross-selling system to perform the recommendations effectively. For example, a user who just purchased a movie DVD in a video and music sale/streaming site may be offered the music DVD with the soundtrack of the movie, even if the user did not show any preference for the music item. Cross-selling recommendation can be viewed as a category of cross-domain recommendation, where the type of item suggested to users is different from the type where they originally expressed their preference. According to Fernández-Tobías et al. (2012), cross-selling is one of the tasks performed by systems that provide cross-domain recommendations.

Cross-domain recommendation is a new field in recommender systems, and researchers are exploiting its techniques to improve on the challenges that prevail in conventional Collaborative Filtering systems. Enrich et al. (2013) generalised the primary objective of a cross-domain recommendation system as the search and discovery of useful relationships among items or users in different domains. For example, a popular movie rating website can take the role of a dense auxiliary domain; while a newly launched book reviewing/rating website can be viewed as having user/item data sparsity problem. In this particular case, the knowledge sources in the auxiliary domain (Movie) are likely to be adaptable for recommendation in the target domain (book). This is because movie genres and book categories can share similar item attributes, e.g. horror films and horror novel. An important question in cross-domain recommendation is how to ensure that target and auxiliary domains are adaptable for knowledge transfer. According to Fernández-Tobías and Cantador (2014), a major issue in cross-domain recommender systems is how to establish a *"bridge"* between domains in order to support the aggregation or transfer of knowledge from an auxiliary domain to the target recommendation domain.

The last decade has seen a rise in social tagging systems, where users can label contents that they and other users have utilised (e.g. watching a movie, reading a book or consuming an item) with freely chosen words known as tags. The collection of tags in the tagging systems creates a base of unstructured information that encodes the preferences of the users. As additional user information and metadata, social tags can be exploited for collaborative filtering. Although social tags are unstructured free-form text, they can expose latent features which may be common across different domains. As a result, social tags have been considered by Shi et al. (2011), Enrich et al. (2013) and Fernández-Tobías and Cantador (2014) as additional latent factors in cross-domain recommender approaches based on MF. In this thesis, we build on these approaches by introducing latent features for an MF-model based on social tags that have been grouped together according to their semantic relatedness.

1.3 Problem Identification

Predicting user ratings is generally achieved in collaborative filtering methods by first finding similarity between a target user's profile and profiles of other users in the system. The user profiles typically contain the rating history of the corresponding user. A common approach is to consider the profiles as vectors of an n-dimensional space and compute their similarity as the cosine of the angle that they form using the cosine similarity measure. The set of profiles with the closest distance can then be selected and grouped as a set in the same neighbourhood. The predictions made for users in conventional neighbourhood-based recommender system is susceptible to errors when there is little information about users' rating history. There is often a small number of items rated by users compared to the large proportion of items/products or services available. This sparsity problem is recognised as particularly challenging in memory-based collaborative filtering systems because of the difficulty in finding users with similar preferences, i.e. user neighbourhoods too small or non-existent. Latent factor models which use dimension reduction techniques such as matrix factorization (MF) (Koren et al. (2009), Shi et al. (2011), Enrich et al. (2013)) have been effectively utilized in place of neighbourhood-based models. These types of model-based collaborative filtering technique uncover latent features using a model learned from the underlying item/user data.

Researchers exploring cross-domain recommender approaches are leveraging the ability of MF models to integrate other sources of user feedback for better rating prediction. Specifically, Shi et al. (2011), Enrich et al. (2013) and Fernández-Tobías and Cantador (2014) have shown social tags can be integrated into MF techniques for cross domain recommendation. They identified the occurrence of repeated tagging patterns as the underlying criteria for cross-domain recommendation. In their proposals and implementation, they considered only tags whose character strings exactly match each other in two domains. Taking an instance of the movie and book domains cited earlier, a user who just watched a movie with a mystery plot may feel happy to tag the movie as "intriguing" and give a 4 out of 5 rating. On the other hand, a user who just enjoyed a book with a detective and adventurous story may give it a 4.5 rating and also assign the tag "intriguing" to express the feeling about the book. If the average of rating values given to items that have been tagged as "intriguing" is high on the rating scale, the authors generalised that items with tag "intriguing" are accompanied with a high rating.

The appeal of this kind of cross-domain process to recommenders systems is seen in cold start situations where users may not have rated or tagged items which have been tagged and rated highly by others. The concept of transferable rating information based on commonality of tags across different domains has motivated the exploitation of tag factors in cross-domain recommender models. Experimental results from the works of Shi et al. (2011), Enrich et al. (2013) and Fernández-Tobías and Cantador (2014) are evidence of the potential of tag-based cross-domain recommender systems.

1.3.1 Tag Semantics and Cross-domain MF Models

Current research efforts in cross-domain recommender systems do not consider the semantic relationship that may exist between social tags that are not common in the two domains e.g. "intriguing" and "fascination" may be similar concepts. To the best of our knowledge, there has not been any investigation in the literature on how the semantic relationship of user-generated tags affect the rating prediction accuracy of a cross-domain recommender system. The closest efforts in literature are Fernández-Tobías et al. (2011) and Rowe (2014) who focused on linking items in different domains using the Semantic Web technologies. Rather than reasoning in ontologies based on the semantic web for the similarity of items, our approach focuses on using subsumption hierarchies in a structured computational lexicon (WordNet) to find semantic similarities between social tags.

We show the potential of our approach by using the toy example in figure 1.2. Suppose we have four different users (Luke, Mary, John and Edna) in a movie domain and another four (Rita, Dave, Jack and Adel) in a book domain. In figure 1.2, the square brackets represent the tag and rating value (scale of $1-5$, with 5 being the highest) assigned by users to items. The broken curve lines denote the types of transfer that can happen between the domains. Our desire is to estimate rating values for users in the target domain (movie domain) from users in a denser auxiliary domain (the book domain). The missing ratings (indicated by ?) in the target domain can be inferred from the auxiliary domain tags ("magician", "award", "serenity") and the rating values that are associated with these tags. While some research work has been done in case one and three, our focus is on cases two and four where string patterns of the tags are different even though they are semantically related. This example is a

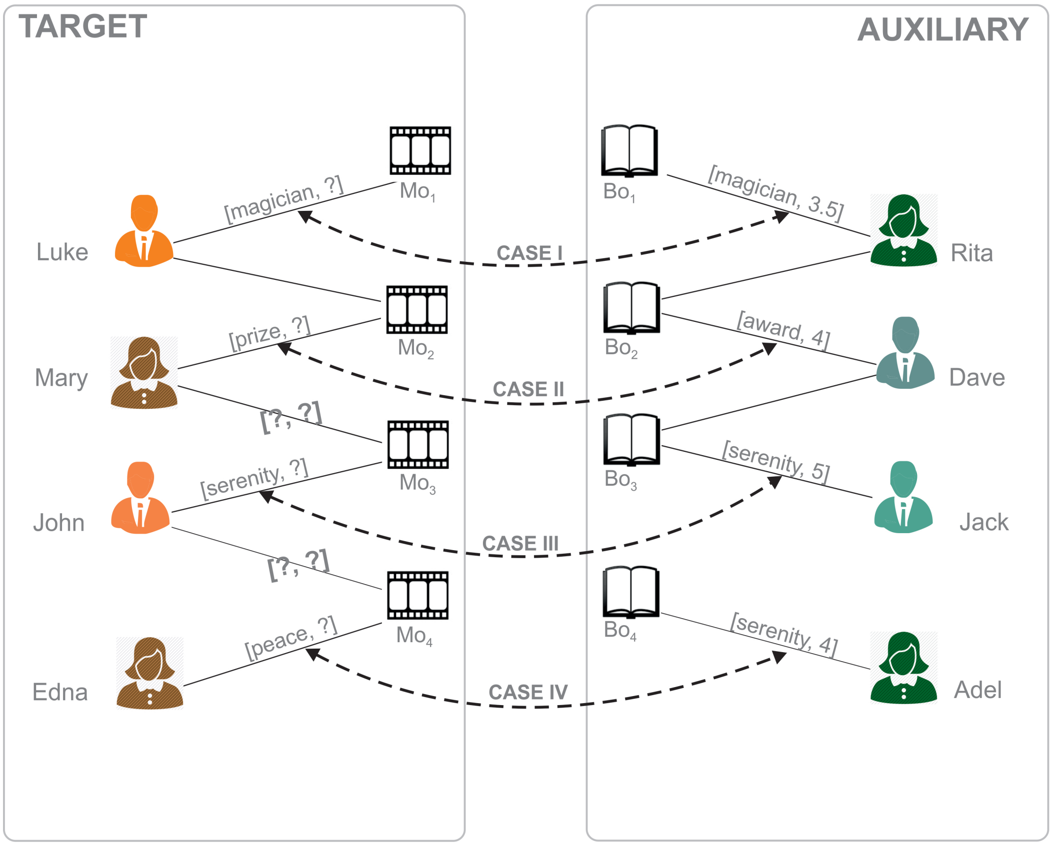


Fig. 1.2 A toy example of possible cases of rating information transfer based on tag commonality/similarity.

very high-level description but captures the idea of rating information transfer in a tag-based cross-domain recommendation. It also shows the potential of integrating semantic similarity to tag-based cross-domain recommender models.

1.3.2 Multi-category POI Recommenders

The fast-paced development of mobile devices, global position system (GPS) and Web 2.0 technologies has led to the rise of Location-Based Social Network (LBSN). LBSN are large social connection hubs where millions of users share rich information, such as experiences, reviews and tips. In addition to being a platform for social interaction, LBSNs have evolved into a system where the rich information can be exploited to infer users' preferences for yet to be visited locations that users may find interesting. The task of recommending new interesting places is referred to as point-of-interest (POI) recommendation.

POI recommender systems are a new field compared to traditional recommender systems. They belong to a group of recommender systems known as Context-aware Recommender

Systems. Several POI recommender models [Ye et al. (2010), Ye et al. (2011), Liu et al. (2013b), Li et al. (2015) and Liu et al. (2017)] that use contextual information such as geographic, temporal and social information have been studied and successfully implemented. However, there remain research issues and opportunities with modelling and combining different contextual information for improving the performance of POI recommenders.

In addition, conventional POI recommender models have used contextual information from POIs that are structured into multiple categories(domains) without considering approaches that enhance knowledge transfer between the different POI categories. There is no study on how POI recommendation accuracy can be improved by using cross-domain recommender approaches. Current researches have treated all POIs as belonging to one domain and utilise single domain collaborative approaches to recommend POIs to users. We identify this as a knowledge gap and pose a question in the next section to investigate the use of cross-domain recommender techniques for POI recommendation.

1.4 Research Questions

As our contribution to the emerging field of cross-domain recommender systems, the following questions are posed to validate and investigate the knowledge gap that we have identified:

- **Research Question 1**: Can a cross-domain recommender model perform better when the size of intersect between the set of tags in a target and auxiliary domain increases?
- **Research Question 2**: Can semantically related tags improve performance of cross-domain recommender model when they are included as additional parameters to the model?
- **Research Question 3**: Can performance of a multi-category POI recommender be improved by incorporating category similarity as context into the model?

The problem of cross-domain recommendation can be expounded by considering two sets of data from two different application domains. The task will be to enrich the sparse target domain by finding a potential collaborator represented by a user or group of users/items from an auxiliary domain. An understanding of how related the two domains under consideration are can help determine which domains will benefit the most from a cross-domain collaboration. In determining how domains can be directly related, Enrich et al.

(2013) identified four situations in which a cross-domain recommendation goal can be realized: a) no-overlap, b) user overlap, c) item overlap and d) full overlap.

In all these scenarios but the first (no-overlap) we could obtain effective recommendations with a classic memory-based collaborative filtering approach by treating all the users and items as belonging to a single domain. However, traditional memory-based models tend to be less accurate when the overlap is small or when the overlapping users or items are very dissimilar. Consequently, there is a need for the extension of conventional approaches such as matrix factorisation into techniques that can support cross-domain recommendations. Cross-domain recommender models have been shown to perform better than memory-based collaborative models in scenarios with little or no users and item overlaps.

1.5 Thesis Contributions

The research activities in this thesis have led to several contributions to the state of the art on cross-domain recommender systems and advanced the subject area in the following ways.

In Chapter 4, a new technique for **selecting optimal semantic metrics for measuring relatedness of tags** was introduced. The metrics were evaluated on how well they predicted if a concept pair was drawn from a single domain (i.e. intra-domain) or different domains (i.e. inter-domain). In later sections of Chapter 4, we **adopted a cross-domain recommender model to evaluate the effect of varying size of intersecting tags** on the recommender's performance. This enabled us to measure the response on prediction accuracy when the size of the intersect is increased by lowering the semantic threshold score for inter-domain tag pairs. In addition, we showed that Term Frequency-Inverse Document Frequency (TF-IDF) does not contribute to the performance of semantically enhanced cross-domain recommendation models when TF-IDF is used to extract relevant domain tags.

In Chapter 5, we present an extension to a matrix factorisation **model that introduces new parameters for adding the influence of semantically related tags** to a cross-domain recommender model. Corpus-based metrics were used to obtain a new set of semantically related tags which was combined with the base MF model. We used current approaches to measuring semantic relatedness by knowledge transfer from a pre-trained dataset to a test dataset enabled by Neural Networks (NN) to measure the relatedness of tags.

In Chapter 6, we present a personalised ranking based matrix factorisation model that exploit attributes of **POI categories which users have checked into in the past to find recommendations for users in multi-category recommender system**. In contrast to numerical ratings, the model utilizes the frequency of user check-ins at different POIs as positive-only feedback. Furthermore, we evaluate two methods for modelling domain/category similarity into the ranking based matrix factorisation model.

The simple framework in figure 1.3 shows the flow and connection of research activities in the thesis. The stage where each research questions (denoted with RQ) was addressed are clearly highlighted.

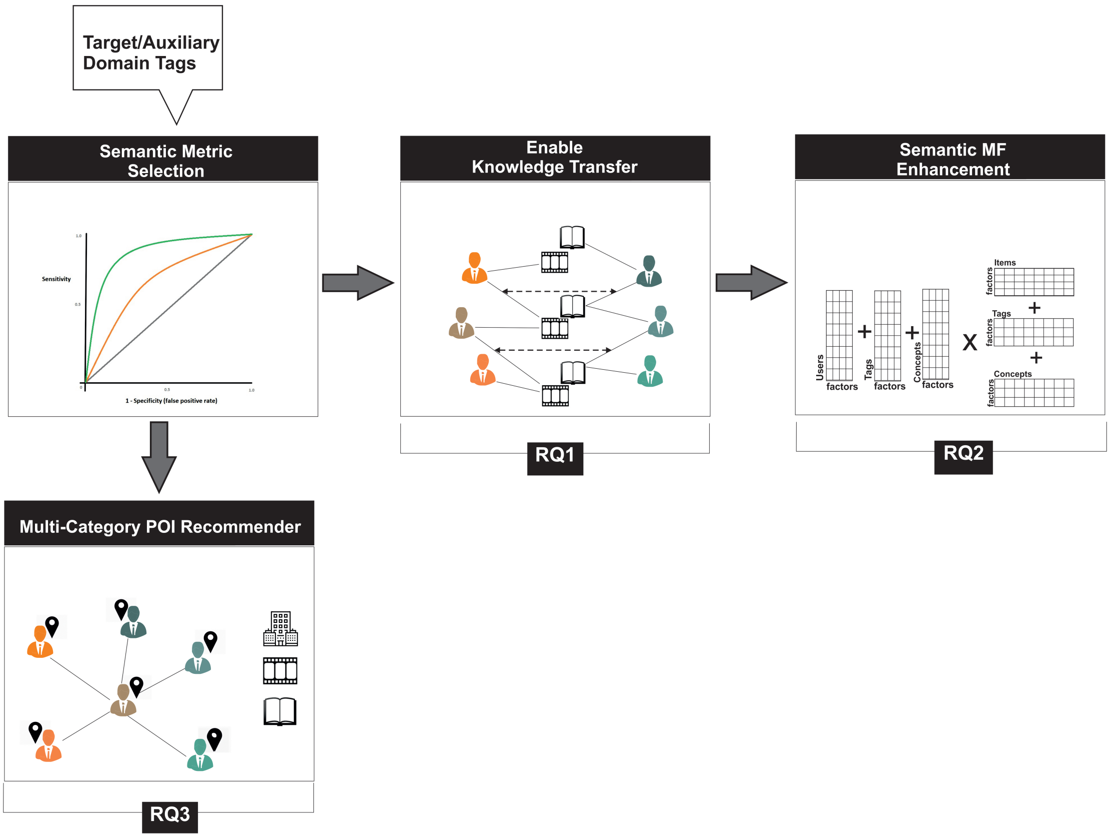


Fig. 1.3 Semantic Enhancement of Cross-domain Recommender System.

1.6 Thesis Outline

The chapters of this work have a similar structure, with sections to introduce and motivate the activities carried out, review known approaches, present the proposed approaches or models, and report and discuss the results achieved during experiments. The rest of the thesis is organized as follows:

In Chapter 2, we survey the state of the art with regards to recommender systems in general and cross-domain recommender systems in particular. The focus of our background review was set on using semantics knowledge from lexical taxonomies and distributional hypothesis in recommender systems for the purpose of enhancing domain knowledge transfer. We also review the current state of research in Point of Interest (POI) recommender systems. Location-Based Social Networks (LBSN) often span across different domains also known as categories. As a result, we reviewed several ranking algorithms and models that can benefit from cross-domain approaches.

In Chapter 3, we first present key terms and concepts that are important for the formal definition of our cross-domain recommender models. We present a general framework and test-bed upon which our experiments were based. We provide a general methodology to our experiments and the evaluation metrics for our models.

In Chapter 4, we investigated the assumption that accuracy of predicted rating depends on the number of common tags between two domains. First, we selected a standard semantic relatedness metric based on how effective the metric was at classifying domains tags. Subsequently, we adopted a cross-domain recommender model that allows us to vary the number of social tags considered as common across a target and auxiliary domain. We used the selected semantic metric to find semantically related tags across the two domains. We evaluated the performance of the cross-domain recommender model as the size of semantically related tags were increased.

In Chapter 5, we went beyond lexical taxonomies to consider more recent similarity methods that are based on word vector representations. We proposed a new cross-domain recommender model that adds an extra parameter to incorporate the influence of semantically related tag pair. We evaluated the performance of known cross-domain collaborative filtering models against our proposed model.

In Chapter 6, we turned to a different field of recommender systems known as POI recommendation. We adapted a novel personalised ranking based POI MF model to include parameters that consider the categories of POIs as extra context information. In later sections, we extend our model to cross-domain POI recommendation and observed the performance of the model in cold start situation and at different varying degrees of data sparsity.

In Chapter 7, we summarise the main contributions of this thesis and give an outlook on future work.

Chapter 2

Review of Recommender Systems

2.1 Recommender Systems

Theoretical concepts about recommendation systems and testing of prototype models were reported by Goldberg et al. (1992) to have begun in the early 1990s; while its mass commercialisation in the form of value-added services was asserted by Resnick and Varian (1997) to have happened in the mid-1990s. Details from a repository on recommendation systems by Ricci et al. (2011) suggest that they are evolving to become software tools which can offer suggestions that support users in decision making. The advances achieved by researchers in the field of information retrieval and filtering have helped in reducing the effort and time spent on locating specific information from the often large collection of structured or unstructured data. Research investigations have been driving innovations and stimulating the development of applications that use retrieval techniques in finding and ranking information based on how relevant they are to a user's query.

On the other hand, progress made in information filtering algorithms have resulted in broadening of search from being merely query based to those that disseminate information selectively by recommending the options that are relevant to users without explicit inputs from the user. A good example can be seen in the case of Google—one of the leading company in information retrieval—now offering a virtual personal assistant (Google Now), which can learn users' behavioural patterns in order to recommend actions to be taken based on the user's location and time. Another point in case as described by Belfiore (2015) is the recent announcement by Microsoft which have been providing information retrieval through its online search offering (Bing) now intending to directly integrate a personal assistant into its latest version of web browser and operating system. These cases further support the empirical findings initially reiterated by Kobsa et al. (2001) that computer systems bring benefits to users in many domains when they take individual characteristics of the users into account and adapt their behaviour accordingly.

Recommender systems generally use machine learning techniques to process stored data about user's interests, represented as user profiles. A ranked list of all resources available for recommendation is subsequently computed based on an algorithm that has been trained using the user profiles. Recommender systems have become a sought-after resource for tackling the challenges brought about by the information overload problem. According to Uchygit (2010), they are popular in application domains such as e-commerce, entertainment/news and fall into three main categories; collaborative-based, content-based and hybrid systems. The following section provides an overview of key concepts behind the evolution of Recommender Systems and the principles governing their operation. While the points presented cover novel techniques at the forefront of information filtering research, the contents do not exhaustively discuss recommendation algorithm performance nor its evaluation methods. The compilation assembled in Ricci et al. (2011) can be consulted by readers that are interested in such in-depth analysis.

2.2 Sparsity and Cold Start Problems

In building recommender systems, historic data about users are generally stored in a user-item matrix also known as the rating matrix. The historic data are past actions of users such as assigning rating values—for example in a range between 1 and 5—to an item. Let U represent the set of users registered to a system, and let I represent the set items in the recommender system. Let the rating matrix of a recommender system be denoted by R with size $|U| \times |I|$. Consider a user u and an item i in the recommender system, let k with a range from 1 to 5 represent the rating value that u gives to i. The element r_{ui} of matrix R can be obtained by the expression below:

$$r_{ui} = \begin{cases} k, & \text{if user } u \text{ rated } i \text{ with value k} \\ unknown, & \text{otherwise.} \end{cases}$$

Users of recommender systems generally provide explicit feedbacks (e.g. ratings) on a small number of items, and they often require the right kind of motivation to do so. In many large-scale applications, the size of both the items and users set are large. As such, the rating matrix will still contain lots of unknown entries even if many users gave feedbacks on items.

This problem is known as the sparsity problem, and it negatively affects the performance of collaborative filtering systems. Sparsity makes the vital task of finding similarity between two users difficulty and renders collaborative filtering approach ineffective. According to Papagelis et al. (2005), even when the evaluation of similarity is possible, it may not be very reliable, because of insufficient information processed. The process of addressing the sparsity problem in a recommender system proceeds from an initial step of estimating the level of sparsity in the system. Let the set of users that have rated at least one item be U_x and the set of items that have been rated at least once by a user be I_x, the sparsity ratio in a recommender system can be calculated as follows:

$$sparsity(\%) = \left(1 - \frac{|U| \cdot |I|}{|U_x| \cdot |I_x|}\right) \times 100.$$

Several methods have been explored in dealing with the sparsity problem in recommender systems. The approaches that are most broadly used include; dimensionality reduction of the user-item matrix (Koren (2008), Fernández-Tobías and Cantador (2014), item-based similarity instead of user-based similarity (Sarwar et al. (2001), Wang et al. (2006)), and content-boosted collaborative filtering (Forbes and Zhu (2011), Lian et al. (2017)).

Cold-start is the state in a system developed for predicting user preferences for items when a model cannot make predictions for newly added users, or predict newly added items to already registered users of the system. Cold-start situations occur due to an initial absence of information about the users and items that have just newly added to the system. Cold-start leads to the case where items cannot be recommended until users have substantially rated them. Similarly, a new user has to give feedback on a sufficient number of items before the recommender system can provide reliable and accurate recommendations to the user. Two approaches are popular in addressing cold start problem. The first approach uses active learning techniques that interactively query the new user to obtain feedback before generating recommendations. For example, some recommender applications actively select individual items or groups of items and present to the user to rate during a signup phase. According to Elahi et al. (2016), the recommender application evaluates the entire set of items and selects the items that are estimated to be the most useful ones (e.g. popular ones that user is most likely to know). The second approach considers additional information about the user during the process of recommending items. User metadata such as gender, age, area code, education and employment information were used by Pazzani (1999) and Vozalis and Margaritis (2006) to compute user-user similarities. In addition, Braunhofer et al. (2015) and Fernández-Tobías et al. (2016) showed that information about the user's personality can be more effective in some applications.

Recent evidence suggests using Cross-domain recommendation approaches that exploit user preferences in different auxiliary domains can alleviate sparsity in a target domain [Cantador and Cremonesi (2014), Shi et al. (2013) Fernández-Tobías (2016)]. Building on current efforts in Tag-based Cross-domain recommendation approaches, we investigate the use of matrix factorization models in addressing the sparsity and cold start problem by enhancing the knowledge transfer process with semantically related tags.

2.3 Content Based Recommender Systems

According to Kobsa et al. (2001), content-based recommender systems have their roots in information filtering and text mining and are typically employed in domains with large amounts of textual content. The general objective of a recommender system as posited by Oard (1997) is to automate the information filtering process such that the results of the suggestions generated by the system resemble those the user would judge as relevant and rate positively to indicate their long-term interest. The process of learning the preferences of users is integral to achieving a recommendation task. The user models that are created to represent the users' needs in the system need to correspond to the natural preferences of the users as much as possible.

Content-based recommendation is a type of supervised learning technique. According to Burke et al. (2011), one can view the problem as one of learning a set of user-specific classifiers where the classes are "useful to user X" and "not useful to user X". In a more specific account, Lops et al. (2011) identified the three components required for a content-based recommendation process. The first and second components, called Content Analyzer and Profile Learner by Lops et al. (2011) serve as the entry point into the system where key features are extracted from both unrated and rated item content. The features extracted from both sources are represented in a structured and machine-processable format. A Content Filter component then compares the two representations using a similarity measure and displays the result according to a score that indicates the most suitable match.

The quality of features extracted remains a key issue in content-based recommender systems. A baseline requirement is that objects to be recommended need to be described so that profile learning that is representative of users can take place. Theoretically, every object would be described at the same level of detail and the feature set would contain descriptors that correlate with the discriminations made by users (Lops et al. (2011)).

2.4 Collaborative Recommender Systems

In order to establish recommendations, CF systems need to compare fundamentally different objects: items against users. According to Koren et al. (2009), there are two primary approaches to facilitate such a comparison, and they constitute the two main disciplines of CF: the neighbourhood approach and latent factor models. Recommendation systems can be designed to use techniques that rely on relationships between users of the system when the item features within the information source is lacking or not sufficient enough to be utilised for discriminating the items. For instance, Lops et al. (2011) identified the limited information associated with the word frequency when trying to model user interests in jokes or poems. The strength of the user/item relationships are estimated by a comparing of all the models that encode the users' interest (i.e. the user profiles) and then grouping those whose pattern of rating items are similar. The users in each of the groups are said to be in the same neighbourhood. These users act as recommendation partners for a target user within the group, and items that occur in their profiles can be recommended to the target user. Recommendation systems based on the collaboration of users in the same neighbourhood is known as Memory-based Collaborative filtering.

Model-based approaches as reported by Breese et al. (1998) have been proposed and explored in the attempt to reduce the execution time involved in calculating the similarity between a given user and all its neighbours. According to Candillier et al. (2007), the general idea is to derive off-line a model of the data in order to predict on-line ratings as fast as possible. This is achieved typically by training some statistical or machine learning algorithm to learn a predictive model using the collection of items ratings (offline), which is then used to make rating predictions for the rest of the data stream (online). A list of the most widely used models include: Bayesian classifiers (Park et al. (2007)), neural networks (Roh et al. (2003)), fuzzy systems (Yager (2003)), latent features (Zhong and Li (2010)) and matrix factorisation (Koren et al. (2009)) among others.

2.5 Hybrid Recommender Systems

Advances in recommender systems have seen the advantages of combining different techniques, such as content-based and CF-based, in a hybrid manner to achieve more accurate performance. Generally, hybrid recommender systems utilise various ways of combining the different techniques in order to leverage on advantages from them all with fewer drawbacks of any individual technique (Burke (2002)). A detailed survey focused on the hybrid systems is compiled by Burke (2002). According to Burke (2002), the hybridisation strategy must be a function of the characteristics of the recommenders being combined. The case of combining content and collaborative recommenders is mostly a function of the quality and quantity of data available for learning. If the recommenders are uniformly unequal, it may make sense to employ a hybrid in which the inaccuracies of the weaker recommender can be contained: for example, a cascade scheme with the stronger recommender given higher priority, an augmentation hybrid in which the weaker recommender acts as a "bot" contributing a small amount of information (Burke (2002)). Hybrid recommender systems are different from cross-domain recommendation systems because they are implemented in single-domain applications.

2.6 Tag-Based Recommender Systems

Social tagging is the act of adding metadata in the form keywords to annotate and categorise items. Several works have explored recommender techniques that leverage tagging data to recommend items. For example, Tso-Sutter et al. (2008) adapted traditional collaborative filtering algorithms by incorporating tagging data into the user-item matrix. Guan et al. (2010) proposed a graph-based learning algorithm that is based on tagging data. A Bayesian model that estimates the preference of users for items based on their inferred preferences for tags was introduced by Sen et al. (2009).

A more related line of work to ours are those of Manzato (2013), Enrich et al. (2013) and Fernández-Tobías and Cantador (2014) where user and item tags are integrated into a matrix factorisation model to predict missing ratings for users. Enrich et al. (2013) and Fernández-Tobías and Cantador (2014) extended the basic framework from recommending in a single domain to cases which utilise cross-domain recommendation techniques. While these techniques showed considerable improvements in rating prediction performance, both authors acknowledged in their work that more accurate results might be achievable if the metadata (e.g. tags annotations) shared between the two domains is increased by considering those with similar semantics but that have been expressed differently in domains.

2.7 Cross-domain Recommender Systems

Traditional collaborative filtering systems are limited in making accurate recommendations when substantial ratings of items by users are not available (sparsity). This drawback according to Ricci et al. (2011) is further aggravated by the fact that users or items newly added to the system may have no ratings at all, a problem known as cold-start. Also, users may have similar preferences but may not yet have rated any item in common; therefore limiting the number of users that can collaborate and benefit from the system. While there have been several proposed approaches to resolve the cold-start and sparsity problem, the majority of the implementations are considered within a single application domain. There are however particular scenarios where providing users with cross-domain recommendations may help reduce the effects of sparsity and cold-start problems.

For example, electronic commerce sites (e.g. eBay, Amazon, Alibaba) with a vast range of items usually allow users to give feedback on items of different types—i.e. items in different domains. The preference of a user desiring to purchase items in a new domain within the commercial site can be inferred from the user's purchase history in other domains within the site. In this case, a cross-domain recommender system could utilise a multi-domain user profile model that can generate recommendations of items over several domains. Another case that justifies the importance of cross-domain recommender systems is the generation of personalised cross-selling or bundle recommendations for items from multiple domains. For instance, a movie item sold with an additional recommendation of a music album similar to the soundtrack of the movie item. In this case, the recommendation of the music album by the recommender system is guided by the user's preferences in the movie domain.

A general assumption of these cross-domain recommender use cases is that there exit inter-domain dependencies between profiles of users and items in the auxiliary and target domains. Winoto and Tang (2008) and Li et al. (2009) have demonstrated that the assumption of strong dependencies between domains is valid in marketing, behavioural, and data mining studies. According to Fernández-Tobías (2016), cross-domain recommender systems leverage these dependencies through considering, for example, overlaps between the user or item sets, correlations between user preferences, and similarities of item attributes.

2.7.1 Cross-domain Recommendation Problem and Tasks

As a relatively new field of recommender systems, there have been several formulations of the cross-domain recommendation problem without any consensus in the recommender research community. Over the years, two main approaches to describing the cross-domain recommendation problem have emerged within literature. On the one hand, researchers proposed models that provide recommendations of diverse items from a combination of several domains. On the other hand, models are developed to reduce the effects of cold-start and sparsity situations in a target domain by using information from an auxiliary domain.

However, there are general agreements on the type of recommendation tasks that cross-domain recommender systems should be able to accomplish. If we consider two domains $\mathcal{D}_x$ and $\mathcal{D}_y$ to explain cross-domain recommender tasks. Let $\mathcal{I}_x$ and $\mathcal{I}_y$ be their respective set of items. According to Fernández-Tobías et al. (2012), the following itemizes recommendation tasks that cross-domain recommender systems can perform in $\mathcal{D}_x$ and/or $\mathcal{D}_y$:

- Cross-selling: recommend items in a new domain, different to the domain where the users had shown a preference, i.e., recommend items in $\mathcal{I}_x$ to users with preferences for items in $\mathcal{I}_y$.
- Multi-domain recommendation: combining items from several domains and recommending them together as a single package to users, i.e., recommend items resulting from the combination in $\mathcal{I}_x \bigcup \mathcal{I}_y$.
- Linked domains: improve recommendations of items in a target domain to users in the target domain by utilising patterns of preferences in an auxiliary domain; i.e., recommend items in $\mathcal{I}_x$ by exploiting knowledge relating $\mathcal{D}_x$ and $\mathcal{D}_y$.

According to Fernández-Tobías et al. (2012), the three recommendation tasks should be considered altogether when formulating the problem of cross-domain recommendation.

2.7.2 Criteria for Cross-domain Recommendation

In order for cross-domain recommender systems to accomplish any of the three recommendation tasks, the active domains must directly or indirectly share some relationship with each other. To clearly describe the criteria for cross-domain recommendation, let $\mathcal{X}^U = \{\mathcal{X}_1^U, ..., \mathcal{X}_m^U\}$ and $\mathcal{X}^I = \{\mathcal{X}_1^I, ..., \mathcal{X}_n^I\}$ be the sets of characteristics used to represent profiles of users and items, with m and n as the size of the profiles respectively. Let $\mathcal{D}_A$ and $\mathcal{D}_T$ denote two domains. As a minimum criterion for reliable cross-domain recommendation, the two Domains $\mathcal{D}_A$ and $\mathcal{D}_T$ must have some relations to each other, i.e.,$\mathcal{X}_A^I \bigcap \mathcal{X}_T^I \neq \emptyset$.

According to Fernández-Tobías et al. (2012), the relationship can be by means of content-based or collaborative filtering characteristics about users and/or items, such as ratings, social tags, semantic relations, and latent factors.

- **Content-based criteria** - Contents and metadata such as keywords, demographics and categories that describe both users and items in a system can form a set of explicit features $\mathcal{F} = \{\mathcal{F}_1, ..., \mathcal{F}_m\}$ with $\mathcal{X}^U \subseteq \mathcal{F}$ and $\mathcal{X}^I \subseteq \mathcal{F}$. User and item profiles in the system are generally represented as vectors whose components are indications of how much interest a user has in a feature, or the relevance of a feature to an item. In this case, domains $\mathcal{D}_A$ and $\mathcal{D}_T$ are related when same features are distributed across both domain i.e., $\mathcal{X}_A^U \bigcap \mathcal{X}_T^U \neq \emptyset$ and $\mathcal{F}_A \bigcap \mathcal{F}_T \neq \emptyset$.

- **Collaborative filtering-based criteria** - Users' feedback about items in Collaborative filtering systems are generally represented as matrix $\mathbf{R} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}|}$ with the element R_{ui} containing the value given by user u to item i. Following the notation introduced earlier, $\mathcal{X}^U = \mathcal{I}$, where $\mathcal{I}$ is the set of rated items. Domains $\mathcal{D}_A$ and $\mathcal{D}_T$ are related on the basis of having same rated items with $\mathcal{X}_A^U \bigcap \mathcal{X}_T^U \neq \emptyset$, i.e., $\mathcal{I}_A \bigcap \mathcal{I}_T \neq \emptyset$. The same reasoning holds vice versa, where $\mathcal{X}^I = \mathcal{U}$, and $\mathcal{U}$ are users with ratings. Domains $\mathcal{D}_A$ and $\mathcal{D}_T$ are related when $\mathcal{X}_A^I \bigcap \mathcal{X}_T^I \neq \emptyset$, i.e., $\mathcal{U}_A \bigcap \mathcal{U}_T \neq \emptyset$.

In addition to content-based and collaborative filtering based criteria for cross-domain recommendations, the profile of users and/or items in a recommender system can be projected to lower dimension spaces as latent factors. In such cases, $\mathcal{X}^U$ and $\mathcal{X}^I$ will be the set of user latent factors and item latent factors respectively.

2.7.3 Domain Combinations for Cross-domain Recommendation

A growing body of literature on cross-domain recommender systems have different notions of what a domain represents. On the one hand, some have considered items like movies and books as belonging to different domains. Other authors have considered the different subgroups within a particular item type as domains. For example, the genres of movies (e.g. action movies and comedy movies) or different categories of books (e.g. romance and fiction) as different domains. Generally, domains have been classified according to the attributes and type of items recommended in the domain. Fernández-Tobías et al. (2012) conjectured that domains could be defined at the four different levels expounded below:

- *Attribute level* - at this level, items that are recommended are of the same type and the same attributes. Items are considered to be in different domains if they differ in the value of a specific attribute. For example, two movies will be considered as belonging to two distinct domains if their genre attribute is different, e.g. action movie and comedy movies.
- *Type level* - items recommended at this level belong to the same type and have some attributes in common, but differ in some other type of attributes. Items are considered as belonging to distinct domains if they have different attribute subsets. For example, console games and streamed games belong to distinct domains, since they have several attributes in common (title, genre) while they differ in regards to other attributes (e.g., the live attribute for steaming games).
- *Item level* - the distinction at this level is more recognisable because the items recommended are of the same type. The items in different domains differ in most, if not all, of their attributes. For example, at this level, movies and books belong to different domains.
- *System level* - at this level, items recommended belong to distinct systems/platforms, which are considered as different domains. For example, movies rated and reviewed in the Rotten Tomatoes (critiquing website), and movies watched and rated in the Amazon Prime (video streaming service).

There are recent research efforts on newer datasets that have been released to the research community to allow investigating different open questions on cross-domain recommender system. A detailed and up-to-date literature review was carried out in order to investigate how the different domains are combined for achieving the objective of cross-domain recommendation. Similarly to the in-depth survey in Fernández-Tobías et al. (2016), it was observed that the most frequently used domains to the least used are in the following order: movies (77%), books(56%), music(36%) and TV(15%). In addition, domains were most frequently combined together as movies-books(37%), movies-music(20%), movies-tv(8%), book-music(15%), book-tv(10%).

In table 2.1, details of the various combinations of domains in the works reviewed on cross-domain recommendation show other types of multi-domain combination. For example, a combination of books, movies, music, games and TV shows in Winoto and Tang (2008). Table 2.1 also shows the type of feedback information that is collected from users as an indicator of their preference or opinion. The feedback information is then used to establish a relation between the domains for cross-domain recommendation.

Table 2.1 Summary of domain combinations for Cross-domain Recommendation.

Domains	**User preferences - datasets**	**References**
books, movies	ratings - BookCrossing, MovieLens/EachMovie	Gao et al. (2013b); Li et al. (2009)
	ratings, tags - LibraryThing, MovieLens	Enrich et al. (2013); Shi et al. (2011); Zhang et al. (2012)
	ratings, transactions	Azak (2009)
	ratings - Imhonet	Sahebi and Brusilovsky (2013)
	ratings - Douban	Zhao et al. (2013)
	ratings - Douban	Zhang et al. (2016)
movies, music	thumbs up - Facebook	Shapira et al. (2013)
	thumbs up - Facebook	Fernández-Tobías et al. (2016)
	ratings - Amazon	Pagano et al. (2017)
movie genres	ratings - EachMovie	Berkovsky et al. (2007)
	ratings - MovieLens	Cao et al. (2010); Lee and Seung (2001)
books, movies, music	ratings - Amazon	Hu et al. (2013); Loni et al. (2014)
	tags - MovieLens, Last.fm, LibraryThing	Fernández-Tobías et al. (2013)
books, movies, music, TV shows	thumbs up - Facebook	Cantador et al. (2013); Tiroshi and Kuflik (2012); Tiroshi et al. (2013)
book categories	ratings - BookCrossing	Cao et al. (2010)
music, tourism	semantic concepts	Fernández-Tobías et al. (2011); Kaminskas et al. (2013)
restaurants, tourism	ratings, transactions	Chung et al. (2007)
books, games, music, movies & TV shows	ratings	Winoto and Tang (2008)
books, games, music, movies & Perfumes	ratings - Imhonet	Sahebi et al. (2017)
movies	ratings - Netflix	Cremonesi et al. (2011)
	ratings - Douban, Netflix	Zhao et al. (2013)
	ratings - MovieLens, Moviepilot, Netflix	Pan et al. (2012)
music	tags - Delicious, Last.fm	Loizou (2009)
	tags - Blogger, Last.fm	Stewart et al. (2009)

For example, tags assigned to movies in a movie domain may be used as *brigde* to a book domain if the same tag is assigned to books in a book domain. In the number of literature reviewed the most frequently collected type of feedback to the least collected were in the following order: -ratings, -tags, -thumbs up. In some cross-domain recommendation approaches, semantic concepts were used as user preferences. Firstly, a textual description of items in the domains are mapped to concepts in WordNet or Wikipedia and then used to the linking the domains.

2.8 Semantics Recommender Systems

Semantic analysis enhances representation of users and items in recommender systems by mapping the user/item data contents to concepts that are have been defined and structured in external knowledge bases such as concept diagrams (taxonomy or thesaurus) or ontologies. According to Ricci et al. (2011) the main motivation for this approach is the challenge of providing a recommender system with the cultural and linguistic background knowledge which characterises the ability to interpret natural language documents and reasoning on their content. Systems that perform recommendations based on the representations described above and supported by a combination of technologies (e.g. resource descriptions/tagging and knowledge organisation) from the semantic web are generally referred to as semantic (or semantically-enhanced) recommender systems (Peis et al. (2008)).

The benefits of adopting semantic data in the recommendation process have been investigated in the last decade with results that show the potential of overcoming the limitations of traditional recommendation techniques. Previous work such as Middleton et al. (2004), Maidel et al. (2008), Sieg et al. (2010) and Cantador et al. (2011) demonstrated that the exploitation of semantic relations can help to improve performance of traditional models. Pazzani (1999) tackled the cold start problem by using a product taxonomy from which the user profiles are defined (without users needing to provide explicit valuations). The active user profile is used to discover users with similar interests, whose valuations help the system generate recommendations. Cantador and Castells (2006) presented an approach that finds similarities among users by comparing profiles of their interests for semantic topics and specific concepts. By taking advantage of the relations between concepts, and the (weighted) preferences of users for the concepts, the system clusters the semantic space based on the correlation of concepts appearing in the preferences of individual users. This method uncovers implicit social networks that may help to define both content-based and collaborative-based recommender systems. Codina and Ceccaroni (2010) considered recommendations for a movie domain by incorporating semantics to enhance user modelling through the application of a domain-based inference method and providing a more accurate recommendation by applying a semantic-similarity method. As an added contribution to their work, Codina and Ceccaroni (2012) presented new methods for measuring the semantic relatedness between attribute values of items based on their co-occurrence in similar contexts. Our work stands out from all the approaches mentioned above by exploring beyond processes that generate recommendations in a single domain to implementing recommendation techniques that focus on knowledge transfer from an external/axillary domains.

2.9 WordNet Sense Disambiguation

In Natural Language Processing and its application in information retrieval/filtering, the ambiguity of words often arises due to polysemy and synonymy of the natural language unit. While humans have little difficulty in differentiating between various meanings of words, more effort is typically required to replicate such results computationally. According to Jiang and Conrath (1997), when a word level semantic relation requires exploration, many potential types of relations can be considered: hierarchical (e.g. IS-A or hypernym-hyponym, part-whole), associative (e.g. cause-effect) and equivalence (synonymy). To measure the semantic similarity/distance between words and concepts, authors have come up with consistent computational models and measures that combine lexical taxonomies (e.g. WordNet) and statistical contents of corpora (e.g. Brown Corpus) to assess these types of relations.

WordNet is a broad lexical network of English words. Nouns, verbs, adjectives, and adverbs which are each organised into networks of synonym sets (synsets) that each represents one underlying lexical concept and are interlinked with a variety of relations (Budanistsky and Hirst (2005)). A polysemous word will appear in one synset for each of its senses. An illustration of how synonyms and polysemy are represented in WordNet can be seen in the matrix of table 2.2. The Wordnet lexical matrix as described by Degemmis et al. (2007) shows the mapping between word forms and their meanings. Word forms are imagined to be listed as headings for the columns, word meanings as heading for the rows. An entry in a cell of the matrix implies that the form in that column can be used (in an appropriate context) to express the meaning in that row. Thus, entry $E_{1,1}$ implies that word form F1 can be used to express word meaning M_1. If there are two entries in the same column, the word form is polysemous; if there are two entries in the same row, the two word forms are synonyms (relative to a context).

Table 2.2 WordNet lexical matrix showing synonyms ($E_{1,1}$,$E_{1,2}$) and polysemy ($E_{1,2}$,$E_{2,2}$)

Word Meaning	Word forms			
	F_1	F_2	...	F_n
M_1	$E_{1,1}$	$E_{1,2}$		
M_2		$E_{2,2}$		
...				
M_m				$E_{n,m}$

2.10 Semantic Terminologies and Definitions

An account of semantic terminologies used in the rest of the chapter is provided as follows, with the formal definitions of all key terms that were utilized in the measure of semantic similarity/relatedness of tags.

2.10.1 Definition 1: Concepts

A concept can broadly mean a collection of things with a common interest. In many lexicons, it is generally accepted that synonyms are words that have similar meanings. In lexical databases (e.g. WordNet), a set of synonyms, or synset, is a group of synonyms. Correspondingly, a synset refers to an abstract concept.

2.10.2 Definition 2: Relations

Taxonomic relationships are the most frequent in the network of concepts that are in WordNet. As a subsumption hierarchy, they are the backbone of the network and account for close to 80% of the relations (Budanistsky and Hirst (2005)). According to Fellbaum (2005), they form a super-subordinate and transitive relation (also called hyponymy or IS-A relation) that link general concepts to increasingly specific ones. Subsumption is the containment of one concept by another. Generally, concepts on a lower lever are proper subsets of the concepts on a higher level. An is-a hierarchy describes the relationship between each level. The addition of is-a hierarchies creates a taxonomy.

There are other non-taxonomic relations in Wordnet which link synsets that denote parts, components, or members to synsets denoting the whole. This type of relation is known as Meronymy (Fellbaum (2005)), and the synsets are not transitive, i.e. the parts are not inherited "upward" as they are characteristic only of specific kinds of things rather than the class as a whole.

2.10.3 Definition 3: Lowest Common Subsumer

In the hierarchies of Wordnet taxonomy, each node represents a concepts while the edges that link the concepts refer to the relationships between the concept. Least common subsumer (LCS) of two concepts is the most specific concept they share as an ancestor (Budanistsky and Hirst (2005)). Formally, if we consider the WordNet network as a directed acyclic graph (DAG) G having two nodes (concepts) c_1 and c_2 among others, the LSC of the two concept is the lowest (i.e farthest from the root) node that is a superordinate to both concept c_1 and c_2 as illustrated in figure 2.1 The LSC can be located in the graph by tracing of paths from each concept upward towards the root node. For later computation of semantic similarity/relatedness, we denote the LCS of two concepts c_1 and c_2 as $lcs(c_1, c_2)$.

2.10.4 Definition 4: Information Content

The degree of abstractness[1] of a superordinate concept in an IS-A taxonomy reflects the extent to which the subordinate concepts it subsumes share information with each other in the hierarchy. According to Resnik et al. (1999), it is an indication of the degree of similarity between the concepts being subsumed. When tracing two different concepts to their superordinates in a hierarchy with IS-A/hyponymy relation, paths with long lengths generally tend to link to more abstract concepts high in the taxonomy. Simply counting the paths to superordinate concepts for evaluating semantic similarity has been known have flaws. The most notable as reported by Resnik (1995) are the inconsistencies that result when accounting for differences in the distance of the edges from both concepts to their superordinate. According to Resnik et al. (1999), there are wide variabilities in the distance covered by a single taxonomic link, particularly when certain sub-taxonomies are denser

[1]or specificity as it is termed by authors such as Resnik et al. (1999) and Harispe et al. (2013).

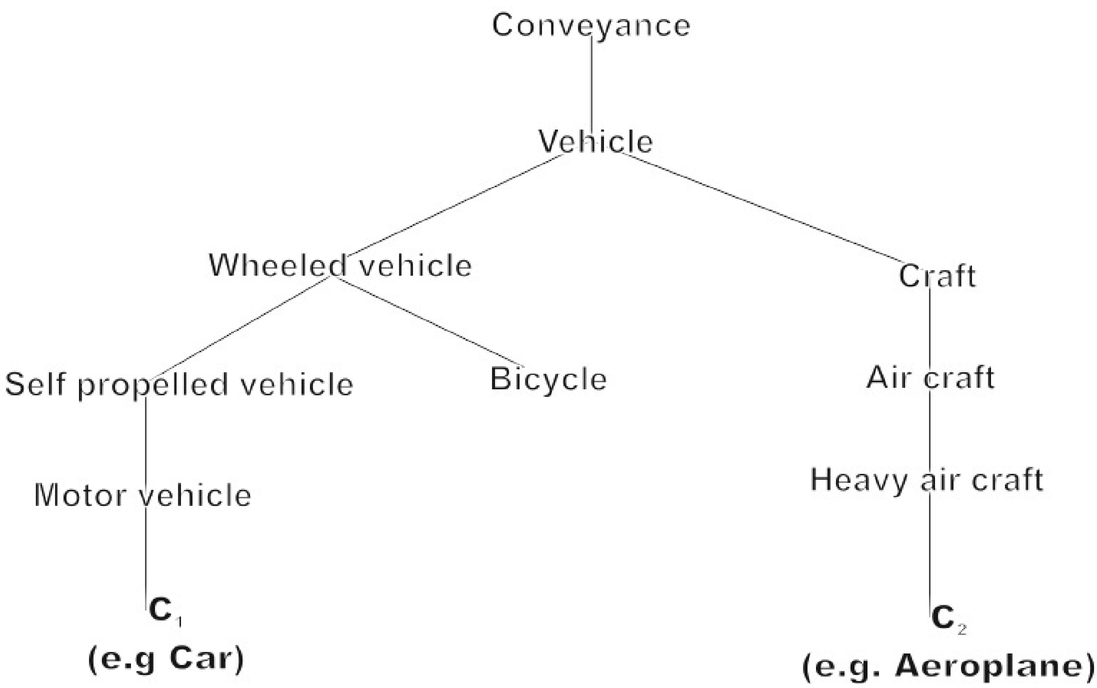


Fig. 2.1 Fragment of WordNet taxonomy

than others. The importance of considering the quality of the information shared by the subordinate concepts—level of "abstractness/specificity"—of the superordinate concept, therefore, becomes apparent. Consequently, information content was introduced by Resnik et al. (1999) to the measure of semantic similarity as opposed to merely finding the length of shortest paths.

The extent to which a superordinate concept appears as an abstract entity is computed based on frequency counts of the concepts as found in a corpus of text (Pedersen (2010)). The information content of the node representing a concept c in the taxonomy is determined by the probability $p(c)$ of encountering an instance of concept c from a corpus . Following the formal definition by Resnik (1995), information content of concept c denoted hereafter as $IC(c)$, is the negative log of the probability of that concept (based on its observed frequency counts in a corpus); and calculated as:

$$IC(c) = -logp(c) \tag{2.1}$$

In order to quantify the information contained in the outcome of an observation, suppose there is a discrete set of possible outcomes $x_1, x_2, x_3, ...$ of some variable X. The basic idea is that the information contained in X taking on a particular value x_i, that is $X = x_i$, is the degree of "surprise" in X taking on this value. For example, if it is almost certain that X takes on the value x_i then this outcome provides very little information; by contrast, if $X = x_i$ is a rare occurrence then this provides more information. For example, if we are considering information regarding the weather on a Caribbean island, then the statement 'it is warm today' contains very little information.

Let $I(x_i)$ denote the information content of X taking on the value x_i. If p_i denotes the probability that $X = x_i$, then the information content function I satisfies the following conditions. Firstly, I depends on the probability p_i rather than the actual value x_i itself. Secondly, I should be a continuous function of probability so that a small change in the probability of x_i results in a small change in its information content $I(x_i)$. Thirdly, as indicated above, I is a decreasing function of the probability p_i so that events that have small probability carry more information than those with larger probability. The final condition is that the information content in two independent events is the sum of the information content of each individual event. This is simply reflecting that, for independent events, the outcome of one event does not influence the outcome of the other so the information contained in both events occurring is the sum of the information associated with each event. Since the probability of two independent events x_i and x_j occurring together is the product of their probabilities, $p_i * p_j$, this last condition becomes $I(p_i * p_j) = I(p_i) + I(p_j)$.

The logarithm of the reciprocal of the probability satisfies these four conditions; furthermore, it can be shown these are the only functions that satisfy these conditions. Hence the information content in X taking on the value x_i is:

$$I(x_i) = log\left(\frac{1}{p_i}\right) = -log(p_i).$$

Expressing information content by associating probabilities with concepts in a taxonomy, convey the same idea as using edge distance in the taxonomy but avoids the earlier stated problems of edge distances. It is observable from equation 2.1 that as probability increases, informativeness decreases, so the more abstract a concept, the lower its information content. Furthermore, if there is a concept at the root/top of the taxonomy, its information content will be zero following equation 2.1.

2.10.5 Definition 5: Semantic Similarity

Semantic similarity can be generalized as an instance of semantic relatedness (defined in the following subsection) because in measuring semantic likeness it only takes into account the "classic" IS-A relation (hyponymy). It specialises the notion of semantic relatedness, by utilizing only taxonomical relationships in the evaluation of the semantic strength between two elements (Harispe et al. (2013)). Consider the example by Resnik (1995), “Car” and “bicycle” are semantically similar, not because they both have wheels and means of steering and propulsion, but because they are both instances of “vehicle”.

2.10.6 Definition 6: Semantic Relatedness

Semantic Relatedness is used in this work to refer to a measure of the association between concepts based on all the possible kind of relationships that can link concepts together. Harispe et al. (2013) defines it as the semantic interactions between two elements without restriction regarding the types of semantic links considered. Following the example by Resnik (1995) as above, “Car” and “gasoline” may be closely related to each other because gasoline is the fuel most often used by cars, typifying a recognisable "functional-association" between the two. Besides the "classic" IS-A relations, the other relationships that the notion of relatedness encompasses are meronymy (part-off), antonymy (opposite-of), and other “non-classical relations” (Morris and Hirst 2004).

2.10.7 Definition 7: Semantic Distance

Generally considered as the inverse of the semantic relatedness, all semantic interactions between the compared elements are considered. According to Budanistsky and Hirst (2005), two concepts are “close” to one another if their similarity or their relatedness is high, and if otherwise, they are “distant”.

2.11 Metrics for Semantic Measure

There have been an extensive volume of work [Resnik (1995), Jian and Conrath (1997), Leacock and Chodorow (1998), Lin (1998), Budanistsky and Hirst (2005)] that have investigated semantic measures between pairs of words/concepts from theories to empirical validations. Two broad categories have emerged based on grouping measures according to the elements of the graph/network that underpins their measuring technique:

- **Edge-Based Approach** - Measures that use this type of approach focus on the analysis of the relationship between pairs of concepts. The distance between concepts in a multi-dimensional concept space can be measured by the geometric distance between the nodes representing the concepts. The shorter the path from one node to the other, the more similar the concept they represent are. Differences in the distance of edges between adjacent nodes and their superordinate often necessitate assigning weights to the edges. Weights are assigned based on features typical to structural characteristics of the hierarchical network. Some conceivable features are local network density, depth of a node, type of link, and strength of an edge link (Jian and Conrath (1997)).

- **Node-Based Approach** - This approach evaluates similarity based on the analysis of nodes in the taxonomy. The measures that use this approach utilise information concept *IC* as defined above in computing the similarity score between concepts. Specifically, the overall semantic similarity score is dependent on the information content value of the most specific super-ordinate node/concept that subsumes the pair of concepts whose similarity is being measured. The value of the information content of a concept is obtained by estimating the probability of occurrence of the concept in a large text corpus (Resnik (1995)).

We expound[2] on the standard semantic metrics that are grouped under these two categories in the following subsections without entirely reintroducing them or elucidating on them in full details. We also restrict the discussion to semantic metrics that were selected for evaluating the similarity of tags in our dataset. Table 2.3 shows the different types of metrics, the category they belong to, the respective type of semantic relationship they measure and the range of score returned when computed using Wordnet taxonomy.

2.11.1 Knowledge-Based Semantic Measures

Knowledge-based measures are generally used to compare terms structured through unambiguous semantic relationships or concepts defined in taxonomies and knowledge organisation systems Harispe et al. (2015). It also encompasses measures commonly used to compare terms or senses defined into lexical databases such as WordNet (Miller, 1998; Fellbaum, 2010). This section focuses on knowledge-based measures and in particular concentrates on measures which rely on ontologies processed as semantic graphs or semantic networks.

[2]The reader interested in more comprehensive details may consult the corresponding references in the subsection.

Table 2.3 Comparison of Different Semantic Similarity Metrics

Metric	Category	semantic measure	Score Range
Leacock-Chodorow	Edge-Based	Similarity	0 - infinity
Resnick	Node-based	Similarity	0 - infinity
Lin	Node-based	Similarity	0 - 1
Jiang-conrath	Node-based	Distance	0 - infinity
Hirst-St-Onge	Edge-based	Relatedness	0 - 16

Leacock-Chodorow Measure

Leacock-Chodorow Measure (LCH) as proposed by Leacock and Chodorow (1998) is based on the edge-counting approach. According to Pedersen (2010), LCH measure finds the shortest path between two concepts, and scales that value by the maximum path length in the is–a hierarchy in which they occur. The similarity between two concepts c_1 and c_2 equals the number of nodes along the shortest path between them, divided by double the maximum depth (from the lowest node to the top) in the taxonomy in which c_1 and c_2. For example, the number of nodes between two siblings, i.e. two nodes with the same parent node is three. Tne equation 2.3 below shows how LCH is computed for two concepts c_1 and c_2:

$$Sim_{LCH}(c_1, c_2) = -log\left[\frac{len(c_1, c_2)}{2\ \mathbf{x}\ Max(D)}\right], \tag{2.2}$$

$len(c_1, c_2)$ is the length between concept c_1 and c_2.
$Max(D)$ is the maximum depth when tracing from the root node to the lowest node on the path containing concept c_1 and c_2.

Resnick Measure

Resnik (1995) initiated the node-based approach to semantic similarity evaluation. The main contribution was the introduction of Information Content (defined in the preceding subsection) to the measure of semantic similarity measure. Resnik (1995) stated that the similarity score of two concepts in an IS-A taxonomy equals the information content value of their lowest common subsumer dominating them both). By considering the hierarchy of concepts in a multidimensional space, Resnik (1995) identified the LCS as the specific concept node that subsumes the pair of concept nodes whose similarity is being evaluated. More precisely, this superordinate concept should be the first upward in this hierarchy that subsumes both concepts (Jian and Conrath (1997)). The value of the information content of a superordinate concept is then obtained by estimating the probability of occurrence of this concept in a large text corpus. Following the definition of information content as given priorly in equation 2.1, the similarity of two concepts according to Resnik (1995) can be formally defined in equation 2.3 below:

$$Sim_R(c_1, c_2) = -logp(lcs(c_1, c_2)) \tag{2.3}$$

Jiang-Conrath Measure

Jian and Conrath (1997) proposed a method that combines both edge-counting and node-based approach. The result of experiments by Jian and Conrath (1997) confirmed that information content approach proposed by Resnik (1995) provides a significant improvement over the traditional edge counting method. Jian and Conrath (1997) also that the proposed combined approach outperforms the approaches using information content only when both the experiment output are correlated with similarity results based on human-judgement. The Jiang-Conrath measure gives semantic scores in terms of distance between two concept nodes. Jian and Conrath (1997) defined such distance metric by considering a particular multidimensional semantic space where every node (concept) in the space lies on a specific axis and has a mass (based on its information content or informativeness). The semantic distance between any such two nodes is the difference of their semantic mass if they are on the same axis or the addition of the two distances calculated from each node to a common node where two axes meet if the two original nodes are on different axes (Jian and Conrath (1997)). Formally as shown in equation 2.4, the similarity by iang-Conrath measure is defined by the weight of the shortest path which links the concepts being compared and contains their LSC.

$$Dist_{JC}(c_1, c_2) = IC(c_1) + IC(c_2) - 2 \textbf{ x } IC(lcs(c_1, c_2)) \tag{2.4}$$

Lin Measure

The Lin semantic similarity measure also augment the information content of the LCS of two concepts with the sum of the information content of the individual concepts. Lin (1998) attempted to define a measure of similarity that would be both universal (applicable to arbitrary objects) and theoretically justified (derived from a set of assumptions); rather than being dependent on a particular application, domain, or resource, as was the case priorly. Based on a set of assumptions that were directed towards the concerns above, Lin (1998) proposed a metric for estimating similarity of two concepts A and B:
The similarity between A and B is measured by the ratio between the amount of information needed to state their commonality and the information needed to fully describe what they are (Budanistsky and Hirst (2005)). Lin's measure of similarity between two concepts in a taxonomy is an outcome of this theorem and is given in equation 2.5 below:

$$Sim_L(c_1,c_2) = \left[\frac{2\ X\ logp(lsc(c_1,c_2))}{logp(c_1)X\ logp(c_1)}\right] \tag{2.5}$$

The probabilities $p(c)$ are determined in a manner similar to Resnik's $p(c)$ in equation 2.4.

Hirst-St-Onge Measure

The Hirst-St-Onge measure is a path based measure that computes semantic relatedness as opposing to semantic similarity i.e. it considers not only the IS-A relation but all other types of relationships between concepts in the taxonomy. Hirst and St-Onge (1998) classifies relations in a taxonomy as having direction. The edges corresponding to an IS-A relation are upwards, while those with HAS-PART relations are considered to be horizontal. The semantic relatedness between two concepts is obtained by tracing a path between them, that satisfies the criteria of neither being too long nor changes direction too often. For two concepts c_1 and c_2 in a taxonomy, Hirst and St-Onge (1998) proposed a measure whose approach may be summarised by the equation 2.6:

$$Rel_{HS}(c_1,c_2) = C\ - len(c_1,c_2)\ - K\ x\ turns(c_1,c_2). \tag{2.6}$$

Where C and k according to Budanistsky and Hirst (2005) are constants (in practice, they used $C = 8$ and $K = 1$), $turns(c_1,c_2)$ is the number of times the path between c_1 and c_2 changes direction and $len(c_1,c_2)$ is the length of the shortest path from concept c_1 to c_2.

2.11.2 Corpus-Based Semantic Metrics

Corpus-based semantic measures use statistics on the natural distribution of words in large volume of texts that maybe unstructured or semi-structured. According to Harispe et al. (2013), they are based on NLP techniques which often rely on statistical analysis of word usage in test, e.g. based on the analysis of word (co-)occurrences and the linguistic contexts in which they occur. They are often referred to as distributional measures in literature to highlight the dependence of the measures on distribution hypothesis. Distributional hypothesis as proposed by states that words occurring in similar contexts convey similar meaning. According to Harispe et al. (2013), studies of distributional measures are tightly related to spatial representations of the semantic space which characterises a corpus and the words to compare.

2.12 Evaluating Semantic Measures

Several works [Rubenstein and Goodenough (1965), Miller and Charles (1991), Pakhomov et al. (2010)] have shown that there is high consistency and collective agreement on the semantic similarity of pairs concept/words when the method for measurement is dependent on human judgement. While studying the relationship between similarity of context and similarity of meaning (synonymy), Rubenstein and Goodenough (1965) collected "synonymy judgements" from 51 human subjects on 65 pairs of words. The pairs ranged from "highly synonymous" to "semantically unrelated", and the subjects were asked to rate them, on the scale of 0.0 to 4.0, according to their "similarity of meaning". A study by Miller and Charles (1991) chose 30 pairs from the original 65, taking 10 from the "high level (between 3 and 4), 10 from the intermediate level (between 1 and 3), and 10 from the low level (0 to 1) of semantic similarity", and then obtained similarity judgments from 38 subjects, given the same instructions as Rubenstein and Goodenough (1965) on the 30 chosen pairs.

The following are more recent sets of word pairs that are correlated with human judgement for measuring semantic similarity.

- WS353-Rel by Finkelstein et al. (2001) is made up of 353 pairs of words and 13 to 16 human subjects were asked to assign a numerical similarity score between 0.0 to 10.0 (0=totally unrelated and 10=very closely related). This dataset was collated to measure general relatedness rather than similarity because as it takes into consideration kind of semantic relations (e.g., antonyms are considered as similar).

- WS353-Sim was created by Agirre et al. (2009), the dataset consists of 203 pairs of words and is a subset of WS353. It is most notably used for evaluating semantic similarity in literature.
- SimLex was created by Hill et al. (2015) and is the most recent of the set, consisting of 999 word pairs for evaluating semantic similarity. The dataset contains 111 adjective pairs (A), 666 noun pairs (N), and 222 verb pairs (V). Each pair of words was rated by at least 36 subjects (native English speakers) with similarity scores on a scale from 0.0 (no similarity) to 10.0 (exactly mean the same thing) and the average score was assigned as final human judgement score.

The datasets are a list of triples with each comprising of two words and a similarity score assigned by the human subjects. The human ratings on those word pairs have been proven to be highly replicable. According to Resnik (1995), the correlation obtained from M&C with respect to R&Gs experiment was 0.97. They replicated the M&C's experiment again in 1995 using 10 computer science graduate students and post-doc researchers to assess similarity. The correlation with respect to the M&C's results was 0.96. This shows high consistency with human assessment of semantic similarity between words. More recently, Schwartz and Gomez (2011) used three different datasets based on human judgement to experimentally show that there is between 73% to 89% inter-human agreement between scores of semantic similarity associated to pairs of words.

2.13 Point of Interest Recommenders

The convergence of global position system (GPS) and Web 2.0 technologies has increased the popularity of Location-Based Social Network LBSN. In LBSNs, users can start social connections with other users, upload contents, and share their locations by check-in to different points of interest (e.g. hotels, airports, restaurants, bars). Point-of-Interest (POI) recommender systems have become essential to both users and providers of LBSN. Users benefit from POI recommendations that help them explore attractive locations and providers of LBSN services benefit from increased revenue by providing targeted services such as location-aware advertisements.

POI recommender systems are a new type of recommender systems and several factors such as shown in figure 2.2 influence the recommended POIs. POI recommender approaches differ from traditional recommender systems techniques based on the three influencing factors described below.

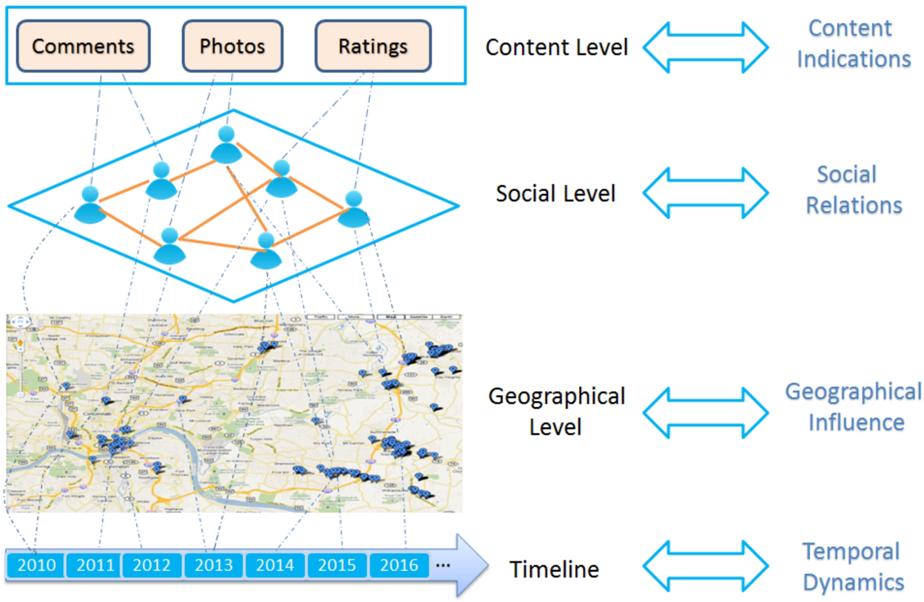


Fig. 2.2 Factors Influencing POI Recommender System (Source: Zhao et al. (2016)).

2.13.1 Geographical Influence

According to Tobler (1970), everything is related to everything else, but near things are more related than distant things. This statement is widely known as Tobler's first law and in LBSN it implies that most users will prefer to visit nearby locations rather than distant ones. This kind of proximity influence is vital in POI recommender systems, and it distinguishes it from traditional POI. Also, users are more interested in the POIs that are around the location of the POI they have shown a preference towards. Several types of model to represent the effects of proximity as it affects users' visiting behaviours in POI recommenders are shown in Zhang et al. (2012). One model by Ye et al. (2011) utilised power law distribution to model geographical influence. Power law distribution pattern has been observed in human mobility such as withdrawal activities in Automatic Teller Machines and travels in different cities [Brockmann et al. (2006), Gonzalez et al. (2008)]. Ye et al. (2011) leveraged the power law distribution to model the geographical influence and combine it with conventional collaborative filtering techniques to recommend POIs. Other models considered in order to incorporate geographical influence to improve POI recommendation are the Gaussian distribution model, and kernel density estimation model and detailed in the survey by Zhao et al. (2016).

2.13.2 Social Influence

In traditional recommender systems, the assumptions that friends tend to have similar interests have led to models that combine social relations with rating information to improve the quality of item recommendation. The work of O'Donovan and Smyth (2005) and Jamali and Ester (2010) are examples of memory-based models that reported benefits of including social relationships to recommender systems. On the other hand, works by Ma et al. (2008) and Guo et al. (2015) are model-based examples that have shown how beneficial social relations are when utilised in recommender systems.

The success of the aforementioned efforts and others alike have prompted several researchers to attempt to adopt the concepts of combining social relations with location check-ins in POI recommender systems. However, results of work by Ye et al. (2010), Cheng et al. (2012) and Gao et al. (2012) who proposed POI recommender models with social influence only showed limited improvements. According to Zhao et al. (2016), this can be explained by the ease at which users in LBSNs make friends online without any limitation. Studies by Ye et al. (2010) have shown that a large number of friends in LBSN do not have any POI that they have visited in common. Specifically, around 96% of users share less than 10% common visited interest. Ye et al. (2010) and Zhao et al. (2016) concluded that social influence contributes limited effects on users' check-in behaviours.

2.13.3 Temporal Influence

Previous researches have demonstrated that temporal influence in traditional recommender system can effectively model preferences users per time and improve recommendations accordingly. Matrix factorisation approaches such as Koren et al. (2009) and random walk based approach such and Xiang et al. (2010) is an example of successful implementation of "time-aware" recommender models. In real life situations, there are physical constraints on check-in activities of users at different POIs (e.g. opening or closing times). These time constraints result into specific patterns of user behaviour and underscore the importance of modelling temporal influence into POI recommender systems.

Temporal influence in LBSNs recommendation system are of three different types: periodicity, consecutiveness, and non-uniformness(Zhao et al. (2016)). Periodic patterns are observable at specific time periods. For example, customers often visit Restaurants at noon while Nightclubs have higher visitations at night time. There are also weekly time trends such as workplaces with high check-in during weekdays and shopping malls during weekends.

Seasonal variation in the number of visits to certain POI are also recorded during national holidays e.g. airports during bank holidays. This type of check-in activity where users visit the same POI at certain time period has motivated researchers [Cho et al. (2011), Gao et al. (2013a), Yuan et al. (2014), Yuan et al. (2013), Li et al. (2015)] to try to model such behaviour for POI recommendation.

Consecutiveness is the type of check-in pattern that follows specific sequences in time and succession in location. For example, a user may decide to do some exercise in the local gym after long work hours. This check-in pattern according to Zhao et al. (2016) implies that local gym and workplace are geographically adjacent in terms of venue function. Two POIs with short check-in intervals are considered to be highly correlated and in work of Cheng et al. (2013), Feng et al. (2015) they used the factorised personalized Markov Chain (FPMC) model to recommend successive POIs.

Non-uniform temporal influences are check-in activities that show the variance in users preference at different hours of the day, at different days of the week and at different months of the year. A study by Gao et al. (2013a) using an example of a random user's aggregated check-in activities on the user's top five most visited POIs showed that user's check-in preference changes at different hours of a day. Similar temporal characteristics also appear at different months of a year and different days of a week. According to Zhao et al. (2016), a user's life custom may explain the non-uniform nature of his check-in: (1) At morning hours, check-in at POIs close to the user's home. Visit locations around the office during the day, and have fun in the club at night time. (2) In a week, the users may check-in more around home and office at weekdays and more at touristy POI and shopping malls at weekends. (3) Users may prefer different food and entertainment during different months. A user may visit ice cream shops and swimming pools and local beaches during summer and visit indoor attracts during winter periods.

This type of temporal influence has been modelled to improve POI recommendations in the work of Cheng et al. (2013), Gao et al. (2013a), Yuan et al. (2013) and Zhao et al. (2016)

2.13.4 Frequency Data and Sparsity

Conventional recommender systems depend on historic user-generated contents as a source of user preferences that should be closely matched by items suggested by the system. In contrast to conventional recommender systems where user preferences are indicated as a rating on a scale, POI recommenders infer user preferences from the frequency of check-in at different POIs. The frequency of a user's visit to a POI cover a larger range compared to ratings (e.g. 1-5 or like/dislike). For example, a user may check-in to his/her a favourite restaurant 100 times in a month, while checking into a less preferred location only a 10 times in the same month. Furthermore, the sparsity in utility ($Users \times Items$) matrix of a conventional recommender system is significantly smaller than the sparsity in the utility ($Users \times POI$) matrix in a POI recommender system.

In addition to the frequency of visit, other user-generated contents such as comments can be used to enhance the POI recommendation. POI system users can provide additional textual information beyond the check-in behaviour. Compared with the check-in activity, the comments usually provide explicit preference information, which is a kind of complementary explanations for the check-in behaviour (Zhao et al. (2016)). Contents in the form of user comments have been harnessed for better recommendation by authors such as Yang et al. (2013) and Gao et al. (2015).

2.14 Explaining Recommendations

The process of selecting a set of items to be recommended to users is an important step in building an effective recommender system. In order for a recommender system to be of value to users, the items recommended must be presented through an interface where explanations about the recommendations can also be displayed to the users. According to Vig et al. (2009), recommender systems tell users what items they might like while explanations of recommendations reveal why they might like them. Explanations add several values to the recommender system. Studies and surveys such as McSherry (2005) and Tintarev and Masthoff (2007) have shown that trust, user satisfaction, and transparency are a broader set of goals that contribute to the value that users get when they interact with a recommender system. The work of Bilgic and Mooney (2005) showed that explanation assisted users in making more accurate decisions. Herlocker et al. (2000) found that explanations improved user satisfaction and acceptance of recommendations. Trust and loyalty were also found to increase in Sinha and Swearingen (2002) when explanations accompany the set of items recommended to users.

In the last two decades, recommender systems have evolved into useful tools that perform well at guiding the user on how to manage the overwhelming information that they see daily. In recommender systems literature, more research efforts have been directed towards recommender models' accuracy than advancing the way the recommendations are brought to users. Studies, however, suggests that users want explanations of their recommendations. According to Vig et al. (2009), a survey of users of a movie recommender site showed that 86% of those surveyed wanted an explanation feature added to the site. Generally, users want explanations to reveal how items recommended to them are related to their personal preferences. A conventional way of presenting the relationship between users and the recommended item is to use an *"intermediary entity"*. As illustrated in figure 2.3, an intermediary entity is used to infer the relationship between a user and a recommended item based on the relationship the user has with other users or items in the system. Explanations of recommendations fall into one of three categories: item-based, user-based, and feature-based, depending on the type of intermediary entity used to relate the user to the recommended item (Vig et al. (2009)).

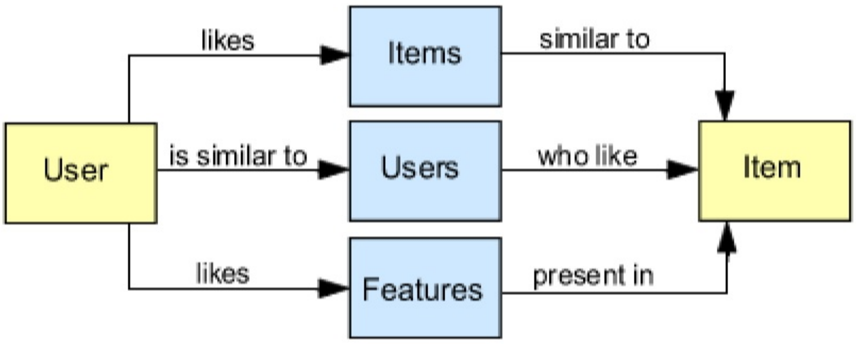


Fig. 2.3 Intermediary entities relating users to recommended items (Source: Vig et al. (2009)).

In explanations that are item-based, a set of items that a user has liked or ranked are used as the intermediary entity. User-based explanations compare historic feedback of a target user with other to establish similarity in taste and then utilise the users as intermediary entities. Feature-based explanations use attributes of the recommended item as intermediary entities. For example, book recommender systems use book attributes like author, genre, and publisher to justify relationships between users and items recommended.

2.14.1 Tag-based Explanations

Tags quickly become abundant in systems that allow organisation of user-generated contents into categories because they enhance navigation of content. Tags may describe what an item is, what it is about or they may suggest the item have certain features (Golder and Huberman (2006)). Tags have been demonstrated to provide both factual and subjective descriptions (Sen et al. (2006)). The information tags can reveal about items make them useful as a feature-based intermediary entity for explaining a recommendation. However, tag quality varies in different social tagging systems and usefulness of each tag need to be assessed before they are exploited for an explanation.

Vig et al. (2009) defined two ways—tag relevance and tag preference—for assessing tag quality. Tag relevance represents the degree to which a tag describes a given item while tag preference measures the user's sentiment to the given tag, for example, the number of likes or dislikes a user assigns to a particular genre, e.g. comedy-action. Tag-based explanations such as presented in Guy et al. (2010) show how a tag relates to an item and how the user relates to the tag. An example of a text description for a tag-based explanation given to a movie (Predator) is shown below:

*"You have been recommended the movie Predator because it is tagged with **explosion** and you have enjoyed other movies tagged with **explosion**".*

Tag-based cross-domain recommender systems that utilise memory-based models can use tags as intermediary entities in a similar approach to conventional collaborative recommender systems. The textual description can be extended in cases where the semantic relatedness of tags has been used to recommend more items. An example of the explanation text for a tag-based cross-domain model is stated below:

*"You have been recommended the movie Predator because it is tagged with **explosion** and you have enjoyed other movies tagged with **war** which has a 74% similarity to **explosion**".*

2.14.2 Explainable Recommendation for Latent Factors Models

Latent factor models such as matrix factorisation are more challenging to explain because the representations for users and items are projected to a latent space. The user and item feature vectors are defined in a low dimensional latent space where each dimension represents a particular factor that influences user decisions. While the factors are represented in the latent space with reduced dimensions, the meanings of these factors are not explicitly known, and therefore the recommendations provided by latent factor models are difficult to explain.

Research efforts by Zhang et al. (2014) and Chen et al. (2016) show more recent approaches that extract explicit product features from textual user reviews and align each latent dimension in matrix factorisation with a particular explicit feature. The proposed approach utilises the explicit features to give personalised explanations to users for items recommended. A generic example of such explanation is as follows:

"The product is recommended because you are interested in a particular feature, and this product performs well on the feature".

Tags like reviews are user-generated texts and can, therefore, be processed to extract explicit features from items in a similar manner described by Zhang et al. (2014). In tag-based cross-domain recommender systems that are matrix factorisation models, the labels of the processed tags can be used directly to name/identify the latent features. As a result, the explanation for a recommended movie item (e.g. Predator) with the identified latent feature "explosion" can follow the textual sample below:

"The movie Predator is recommended because you are interested in the movie feature ***explosion*** *which is highly relevant to Predator".*

Chapter 3

Methodology for semantically enhancing Cross-domain Recommender Systems

There are several reasons, both in the information technology industry and in academia for mining latent features from different domains. A relevant use case to our work involves mining features from a dense domain (i.e. one with low sparsity) in order to transfer the patterns learned to a different but similar domain whose sparsity is high. In the case of classification, Pan and Yang (2010) states that it is desirable to use available structure/knowledge of an auxiliary application domain to help build better classifiers/clusters for a target domain. Specifically, recent approaches considered for cross-domain collaborative filtering have been implemented in Shi et al. (2011), Enrich et al. (2013) and Fernández-Tobías and Cantador (2014) to extend conventional single-domain recommendation techniques by integrating additional resources/metadata (e.g. tags) from auxiliary domains. These extra resources enrich those traditionally required (e.g. rating values) for the recommendation process. Cantador and Cremonesi (2014) Recognised that utilising cross-domain techniques in recommender systems can be an opportunity or a problem. The auxiliary domain can be a potential source of bias if it is substantially richer in resources than the target domain. Recommender algorithms in such cases learn how to recommend items to users in the auxiliary domain while falling short for users in the target domain. The auxiliary domain can also be a potential source of noise if the user models in the two domains differ significantly. This can lead the recommender system to treat features from the auxiliary domain as noise during the learning process due to the dissimilarity in user profile representation. Consequently, choosing a methodology that can enhance the use of latent features from auxiliary domains in cross-domain recommender model is of high importance.

In this chapter, we first provide formal definition of the terms that are key to constructing our semantically enhanced cross-domain recommender model. In section 3.2, we review previous works that are closely related to our approach. A general experimental set-up based on collective matrix factorization is presented in section 3.3 as our methodology. A framework and the test-bed for our experiments is also presented in section 3.3. We describe the metrics we used to evaluate the performance of our model in section 3.4.

3.1 Introduction

In this chapter, we first provide a formal definition of the terms that are key to constructing our semantically enhanced cross-domain recommender model. In section 3.2, we review previous works that are closely related to our approach. A general experimental set-up based on collective matrix factorisation is presented in section 3.3 as our methodology. A framework and the test-bed for our experiments is also presented in section 3.3. We describe the metrics we used to evaluate the performance of our model in section 3.4.

Definition 1: Items and Item Factors

Items are objects which have relative utility/value to the different users evaluating them. In order to formally define item-factors, we shall consider a type of collaborative filtering technique known as Model-based[1] technique. According to Koren (2008), the latent factor in model-based collaborative filtering tries to explain the rating value users give to items by users in a way that characterises both items and users on 20 to 100 factors (Koren et al. (2009). These factors are inferred from the history of rating values already recorded in the system.

In a latent factor space of dimensionality k, let I be the set of all items. The item factors are components of the vector function $q(i)$ representing the profile of an item $i \in I$. The components quantify the extent to which the item possesses those factors, i.e. in small or large amounts. The function q below maps an item i to the specific set of n factors/features that differentiates it from other items within the latent space.

$$\begin{aligned} q\colon I &\to \mathrm{I\!R}^{\mathrm{k}} \\ i &\mapsto q(i) = \{f_1, f_2, f_3, \ldots, f_n\} \\ & where 1 \leq n \leq k. \end{aligned}$$

[1]Model-based techniques are described in detail with references to literature in section 2.3 of chapter two

The parameter n is the total number of specific factors that characterise an item, and k is the dimension of the space and the maximum number of factors that can define items in the space. For items such as movies, Koren et al. (2009) states that the discovered factors might measure obvious dimensions such as comedy versus drama, amount of action, or orientation to children; less well-defined dimensions such as depth of character development or quirkiness; or completely uninterpretable dimensions.

Example 1. In the case of a movie recommender system, let us consider a latent space with a dimensionality $k = 2$. These dimensions are indicated by the 2 perpendicular axes of the graph in figure 3.1. Let us take orientation towards children (i.e. PG rating) and comedy as two hypothetical factors. Assuming the intervals (as shown by grids of figure 3.1) for the factors are in increasing order[2] of strength; G, PG, PG-13, R, NC-17 for PG-rating and comedy-adventure, comedy-fantasy, comedy-drama, comedy-romance, comedy-horror for comedy factor. A movie such as Toy Story will have a location in the space on the higher

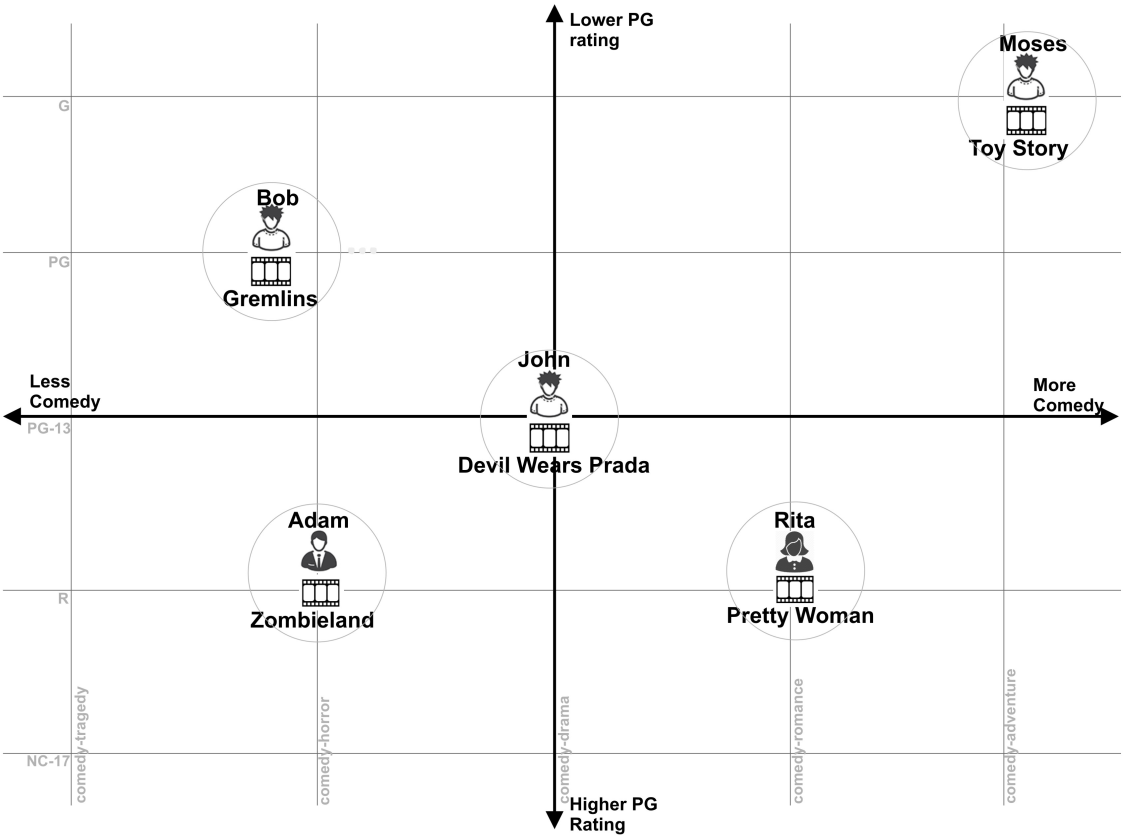


Fig. 3.1 A simplified illustration of item and users factors using two components: comedy vs orientation towards children.

[2] We note that the order of hypothetical factors for the latent space is subjective and open to personal interpretations. The order used for the explanation in this section are those of the researcher.

side of comedy and lower side of PG rating while the location of the rest will follow the order of the list of the factors as in table 3.1.

Table 3.1 Hypothetical Scale of item (movie) factors in a latent space of dimension $k = 2$

	PG-Rating Factor	Comedy-Level Factor
Toy Story	G	Comedy adventure
Gremlins	PG	Comedy fantasy
Devil Wears Prada	PG-13	Comedy drama
Prettty Woman	R	Comedy romance
Zombieland	R	Comedy horror

Definition 2: Users and User Factors

Users of a recommender system are the human agents that have the choice to register the level of satisfaction received from consuming an item and its resources. A user is modelled in the system by a user profile. The user profile is a way of representing the set of item resources that the user has indicated a preference for by giving explicit or implicit feedback. In terms of representations that are key to dimensionality reduction, users can be described by factors that model the users according to the degree of influence the factors have on the user. Formally, let U be set of all users, if we consider a latent factor space of dimensionality k, the user factors are components of the vector function $p(u)$ that represent the taste of user u. The function p below projects the profile of a user u in the latent space as a set of n factors/features that distinguishes him/her from other users in overall set of users U within the space.

$$\begin{aligned} p \colon\ & U \to \mathbb{R}^{\mathrm{k}} \\ & u \mapsto p(u) = \{f_1, f_2, f_3, \dots, f_n\} \\ & where 1 \leq n \leq k. \end{aligned}$$

The parameter n is the total number of specific factors that inform a user's preferential tendencies, and k is the dimension of the space and the maximum number of factors that can influence users' choices in the space.

Example 2. If users and item factors are assumed to be in the joint latent space of figure 3.1, then the following scenarios can be observed. A teenage user "*Bob*" who is strongly influenced by comedy movies that contain horror scenes will be in the position similar to where the movie "*Gremlin*" is in the space. A user "*Rita*", who is captivated by comedies with a lot of romance in it will be around the location of the movie "*Pretty Woman*". A more neutral user "*John*" who is an energetic young adult and driven towards comedies that are intertwined with dramatic plot will be found around the area of the movie "*Devil Wears Prada*".

Definition 3: Ratings and Tag Assignments

For a collaborative recommendation system to achieve the purpose of its design, the history of the responses of users to items they have previously been interested in would have been collected and stored in the system. These user responses help to establish the level of approval an item receives from the users who assign a value within a scale to indicate how much the item appeals to them. These values are known as ratings and can typically be on multiple numeric ranges (e.g. 1-5 stars) or binary (e.g. like/dislike). Additionally, users may have the choice to annotate items by assigning tags to reveal their opinion after utilising the item. These tags are typically natural language terms and can, therefore, be processed for semantic information using Natural Language Processing (NLP) techniques.

If the sets of all users, items, ratings and tags are respectively denoted by U, I, R and T; then function ρ can be applied to any user-item pair (u,i) to obtain a rating value and a set of tags that user u assigned to item i to register his opinion about the item. Let r_{ui} denote the rating a user u gives to item i and let t_{ui} denote the set of tags that user u gives to item i.

$$\begin{aligned} \rho : U \times I &\to R \times \mathbb{P}(T) \\ (u,i) &\mapsto \rho(u,i) = \left(r_{ui}, t_{ui}\right) \end{aligned}$$

The value of r_{ui} can be any value within the allowable range in a system's preference measuring scale e.g. 5 stars, dislike or like. An instance of tag vector t_{ui} gives the set of free-form text i.e. tags that user u assigns to item i.

Example 3. Let the profile of all users, items with their respective rating and tagging information be in the same latent space with two factors. A possible scenario in the space may be represented as the user-item pairs in figure 3.2. Collaborative filtering models can use the rating and tagging information by implementing neighbourhood techniques or model-based techniques. According to Barbieri et al. (2014), neighbourhood models are effective at detecting strong but local relationships, as they explicitly model local similarities.

Model-based approaches typically utilise dimensionality reduction techniques and hence focus on the estimation of weak but global relationships. To predict missing ratings for "*Bob*" or "*Moses*" in the example of figure 3.2, neighbourhood techniques will compute rating and tagging similarity of other users in their respective vicinity (e.g arc around "*Bob*" and arc around "*Moses*") and make a prediction based on the rating values in the neighbourhood vicinity. Whereas, model-based techniques will compute similarity based on the association all users (including "*Adam*", "*John*" and "*Rita*") and items (including "*Devil wears prada*", "*Zombieland*" and "*Pretty woman*") have with a set of underlying factors such as comedy or pg rating.

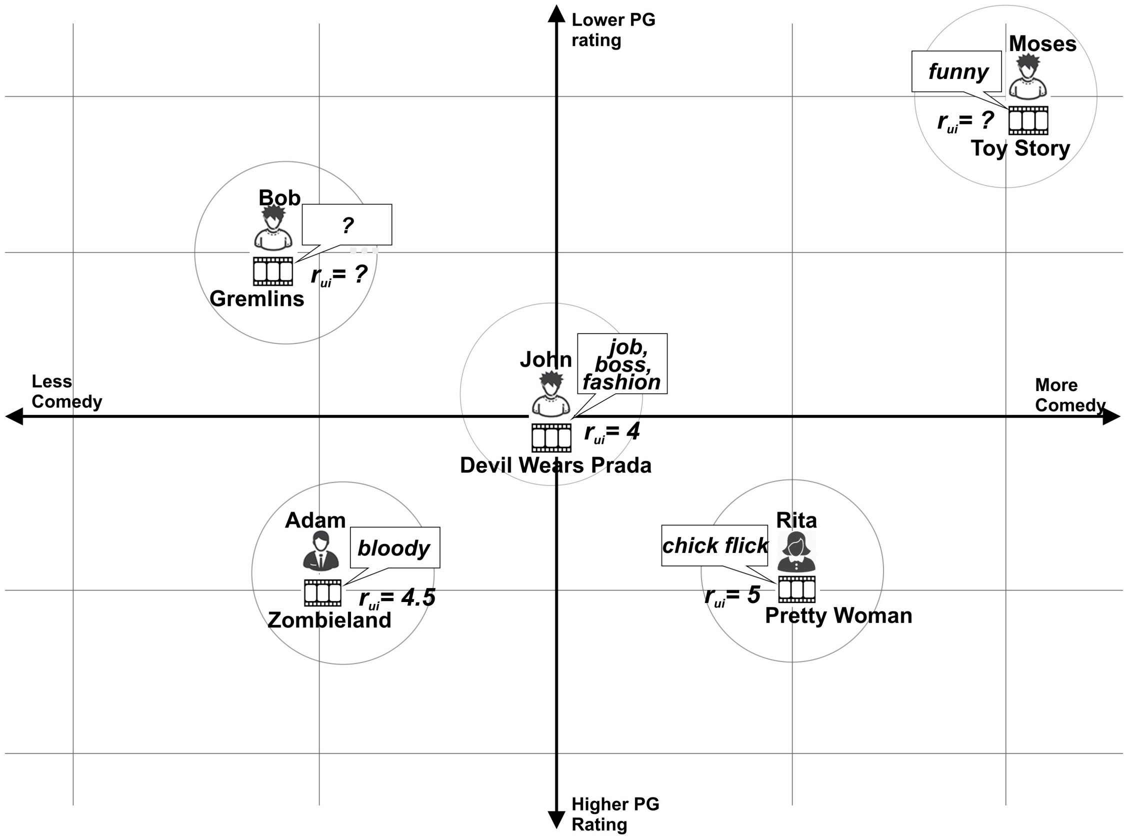


Fig. 3.2 A simplified illustration of the item, users, rating and tag factors, which characterises both users and movies using two axes: comedy vs orientation towards children.

Definition 4: Domains

Different notions of a domain have been considered in literature of recommender systems. In the context of cross-domain recommender systems, Fernández-Tobías (2016) distinguished domain types according to the attributes and types of recommended items. We adopt the *item level* notion where recommended items are not of the same type and differ in most of their attributes. For instance, movies and books belong to different domains, even though they have some attributes in common (title, release/publication year).

Following the definition of items and users in the preceding subsections, we consider a domain as a collection of items and users that have given some ratings and/or tags to items in the collection. The concept of a domain is therefore formalised as follows:

If D is the set of all domains, then the function α maps a domain d to a subset of items belonging to the domain d; while the function β returns the set of users that rate or tag items in the domain d.

$$\alpha : D \to \mathbb{P}(I)$$
$$d \mapsto \alpha(d) = \{i_1, i_2, i_3, \ldots, i_x\}$$

$$\beta : D \to \mathbb{P}(U)$$
$$d \mapsto \beta(d) = \{u_1, u_2, u_3, \ldots, u_y\}$$

The total number of items in the domain is denoted by x, while y is the number of users that tag/rate items in the domain with $1 \leq x \leq \mid \mathbb{P}(I) \mid$ and $1 \leq y \leq \mid \mathbb{P}(U) \mid$ respectively.

In a description that utilises matrix notation, a domain can be represented as a rating matrix that has its users and items arranged in an array of multiple rows and columns. Elements of the matrix are the rating values that users assign to items, and those with missing values imply that users have either not consumed the item yet or did not supply a value to indicate their preference for the item.

Example 4. In order for conventional collaboration and recommendation to be possible in a cross-domain setting, an overlap/intersection of the users and/or items should exist between the two domains i.e.; $\alpha(d_1) \cap \alpha(d_2) \neq \emptyset$ and/or $\beta(d_1) \cap \beta(d_2) \neq \emptyset$. Figures 3.3a, 3.3b and 3.3c illustrate cases where recommendation can be achieved from collaboration of common users and/or items. Examples of the case in figure 3.3a is a book domain and a movie domain with common users subscribed to both. While an example for the case in figure 3.3b can be two different video streaming websites that offer a few common movie items to entirely different users. Figure 3.3c illustrates the case contrary to figure 3.3b where some users are common to both movie streaming sites that stream completely different movie items.

Figure 3.3d exemplifies a case where there are neither users nor items that are common between the domains. Conventional collaborative techniques like the neighbourhood models become less accurate at recommending as there are no direct means of establishing similarity of users. Such a case, therefore, makes it essential to explore techniques that can consider other features (such as implicit or metadata information) for similarity.

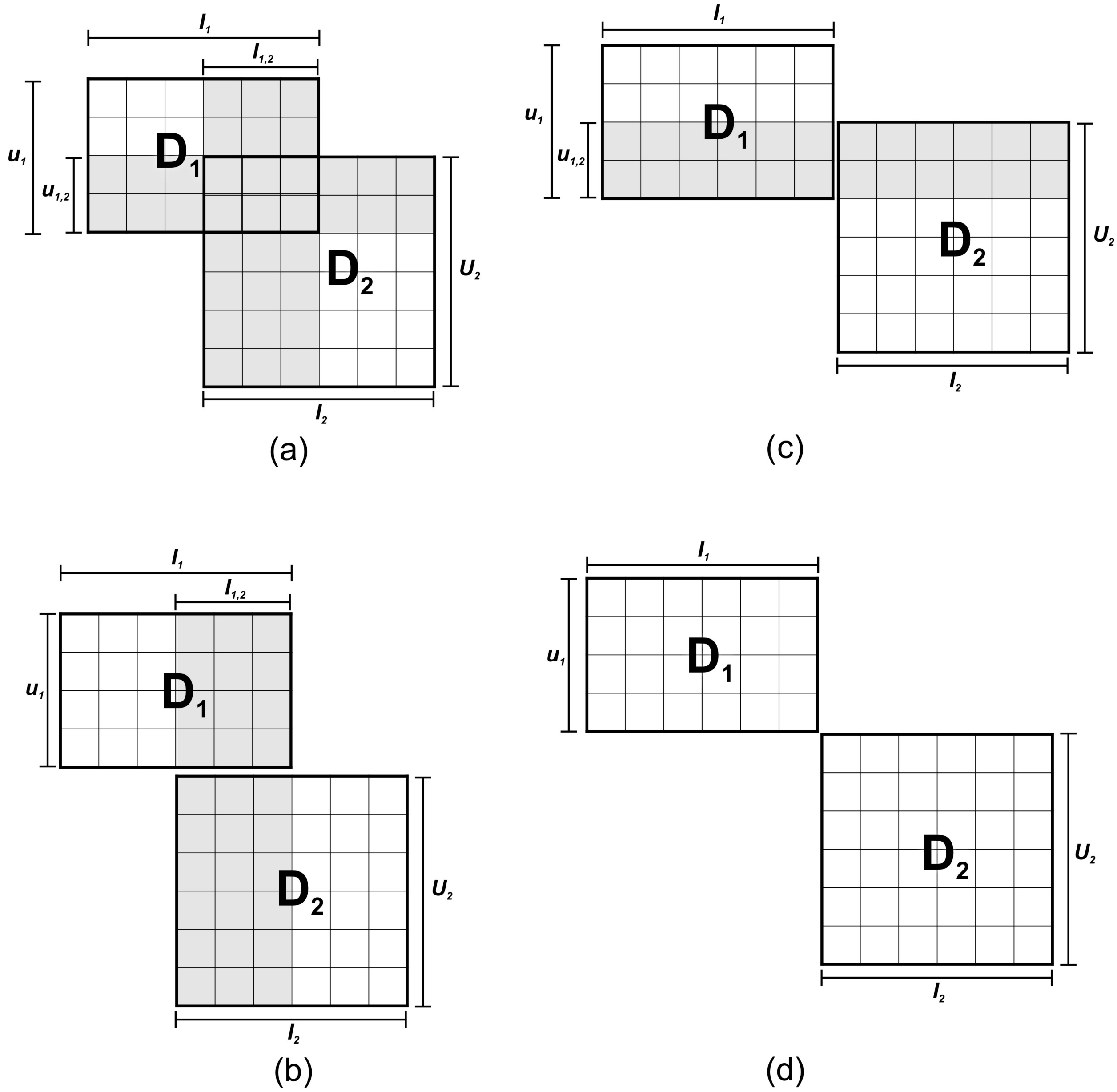


Fig. 3.3 Different scenarios of domain interactions: (a) User and item overlap, (b) Item overlap, (c) User overlap, (d) No overlap.

Definition 5: Matrix Factorization

MF techniques involve decomposing a rating matrix into two smaller matrices that are the best low-rank linear approximation of the original matrix. According to (Koren et al. (2009)), MF is one of the most successful realisations of the latent factor model[3]. A low-rank approximation provides a low-dimensional representation of the original high dimensional rating matrix (Barbieri et al. (2014)). A variation of MF relevant to recommender systems is known as Singular Value Decomposition (SVD). SVD is derived from a linear algebra theorem which states that a rectangular matrix R can be reduced into the product of three matrices - an orthogonal matrix P, a diagonal matrix D, and the transpose of an orthogonal matrix Q (Golub and Van Loan (1989)).

$$R = P \times D \times Q^T \tag{3.1}$$

In equation 3.1, the columns of P are orthonormal singular vectors of RR^T, the columns of Q are orthonormal singular vectors of $R^T R$, and D is a diagonal matrix containing the square roots of singular values from P or Q in descending order.

Example 5. Given a $n \times m$ rating matrix R, where n is the number of users and m is the number of items, the SVD model finds the singular values[4] of R by breaking it down into a product of 3 matrices. Going by the movie example in the definitions above, singular value decomposition of R denoted as $SVD(R)$, given that $k = 2$, is the matrix R_k shown in equation 3.2 reduced to the corresponding user singular vectors of matrix P, the diagonal matrix D with square root of singular values of P and item singular vectors of matrix Q. The derived matrix R_k is not an exact match of R and the process of finding the k largest single values reveals the underlying structure of R and the association its users and items have with the latent factors k.

$$SVD(R) = R_k = P \times D \times Q^T \tag{3.2}$$

If we recall the definition of item factors and user factors in the definitions 1 and 2. Let q_i and p_u be the be the item and user factors of a space with dimension k. The value of each element of R_k can be computed by equation 3.3 below:

$$\hat{r}_{ui} = p_u \cdot q_i^T \tag{3.3}$$

[3]Latent factor models are described in details in section 2.4 of chapter two.

[4]The square roots of the k eigenvalues of $R^T R$ are the singular values of R.

3.2 Related Methods

Recent research works are exploiting the techniques of Cross-domain recommendation with the goal of improving on the limitations of conventional single-domain CF systems. One of the earliest investigations on cross-domain recommendation was that conducted by Winoto and Tang (2008). They speculated that, although cross-domain recommendations may result in lesser precision than traditional recommenders, the former will be more diverse, which may lead to higher user satisfaction and engagement. As further research into the field leads to improved outcomes, more authors have highlighted the advantages of cross-domain recommendation technique in their work. One example is the work of Abel et al. (2011) which gathered additional data from users social web to tackle the cold-start problem. Enrich et al. (2013) worked on minimising the sparsity problem using the commonality of domain tags. According to Moreno et al. (2012), cross-domain techniques typically originate in the machine learning literature and most specifically from Transfer Learning (TL). Details of the work of Zhang et al. (2012) included the use of transfer learning techniques for recommender systems applications in order to improve predictions in sparse target domains by reusing data from a related domain. A more detailed classification of the cross-domain techniques available in literature is presented by Cantador and Cremonesi (2014), where they grouped each technique under two broader categories based on how the knowledge from the source domain is exploited.

3.2.1 Transferring Domain Knowledge

Techniques that use the approach of transferring/linking of knowledge generally relate a source and target domain by means of their shared features or by transferring rating patterns between domains. A prominent work in this category is that of Li et al. (2009) which proposed transferring user-item rating patterns from a dense source rating matrix in a single domain to a sparse rating matrix in a related target domain. However, the method generalises for all cases by assuming that multiple domains share a common rating pattern based on the user-item co-clustering. Other methods in this group have extended the Collective Matrix Factorisation algorithm to learn the latent user and item features and transfer the knowledge from one domain to another. These approaches typically use the latent features from knowledge sources (e.g. item attributes, semantic networks) that are common in both domains as a "bridge" between the domains. These methods, however, tend to be computationally expensive when compared with other approaches and can be limited by its generalisation that contents from knowledge sources in the different domains have the same meaning.

3.2.2 Aggregating Domain Knowledge

The aggregating knowledge category mostly includes techniques that involve merging user preferences into a unified model. A study by Li et al. (2009) on form-based profiles investigated explicit data that users created on the social web and developed cross-system modelling strategies for recommendation systems. The heterogeneity of domains, however, restricts the implementation to profiles that are explicitly provided by users on the social web. The second technique in this group combines user models from the source and target domain into an aggregate two-dimensional matrix representation over which a traditional single domain recommendation technique is then applied. As proposed by Berkovsky et al. (2008), individual recommendations from different single domain recommender systems can be aggregated across the different domains and averaged to obtain the preference of a user in the target domain. For such cross-domain recommendations to be fairly accurate, users or item features have significant overlaps in the source and target domains.

Fernández-Tobías et al. (2011) presents the method of linking domains by using common knowledge and semantics. They introduced a generic framework that uses DBpedia as the basis of integrating knowledge from several domains to provide cross-domain recommendations. The framework demonstrated the benefits of using the semantic information of items to link concepts from two domains. However, this work does not consider historic behaviour of users in the domain when determining item's relevance. Another drawback is that an expert has to identify manually the semantic entities and relations of DBpedia, which can then be used to describe and link the domains of interest. Another approach that used semantic web technologies was proposed by Loizou (2009), which also uses a graph structure to represent relations between domains. A Markov chain model was used to produce recommendations by finding the probability of traversing the graph towards a particular item, using the nodes in the user's profile as starting points. Articles on Wikipedia are used as a universal vocabulary to provide the semantic information on items from various domains. In cases where there are no Wikipedia articles for items being linked, the approach resorts to using free-form tags and discards the conceptual hierarchy of the item.

3.3 Methodology

The general description of our experimental set-up is presented in the following section. This set-up served as the *test-bed* for the experiments in chapters 4 and 5. We note here that our methodology is in agreement with those in previous MF-models such as proposed by Shi et al.

(2011), Enrich et al. (2013) and Fernández-Tobías and Cantador (2014). In a similar manner to these authors, we use a collective matrix factorisation approach as originally proposed in Singh and Gordon (2008). Specifically, a single collection of matrices is constructed from the concatenation of matrices from an auxiliary and a target domain before matrix factorisation. As it is in the previous work, our model does not differentiate between the source and target domains. According to Fernández-Tobías and Cantador (2014), this type of joint factorisation of the auxiliary and target matrices corresponds to the factorisation of the single matrix that results from concatenating the rating matrices of both domains.

In extending MF models for rating prediction task, our methodology differs from previous approaches in the type of tag sets considered for inclusion into the MF model. In addition to tags that are common in the target and auxiliary domain, we considered tags that are also similar between the domains. Primarily, we set-up our experiment to investigate the assumption that rating prediction accuracy is improved when there are more tags common between a target and an auxiliary domain.

We hypothesise that the prediction accuracy of MF-based cross domain recommender model increases when the number of common tags in the target and the auxiliary domain is increased by including tags that are semantically related. We test if the addition of tags that are semantically related to the number of common tags between a target and auxiliary domains have any outcome on the accuracy of predictions. A high-level representation of the phases involved in our first experiment is presented in figure 3.4. The key processes in the phases are described in the following subsections.

3.3.1 Cross-domain Datasets

According to the survey by Fernández-Tobías et al. (2012), the most used datasets have been Movielens, Netflix and EachMovie for the movie domain; BookCrossing and LibraryThing for the book domain; and Last.fm for the music domain. A newly released dataset with multiple domains was made public by He and McAuley (2016)[5]. The dataset contains product reviews and ratings and product metadata such as category information, price and brand from Amazon. The Amazon datasets, however, does not include tagging information since Amazon does not provide a social tagging component on their website.

In order to model a cross-domain collaborative filtering scenario for our experiment, we downloaded two well known, publicly available datasets for the movies and books recommen-

[5]The amazon dataset can be downloaded from http://jmcauley.ucsd.edu/data/amazon/

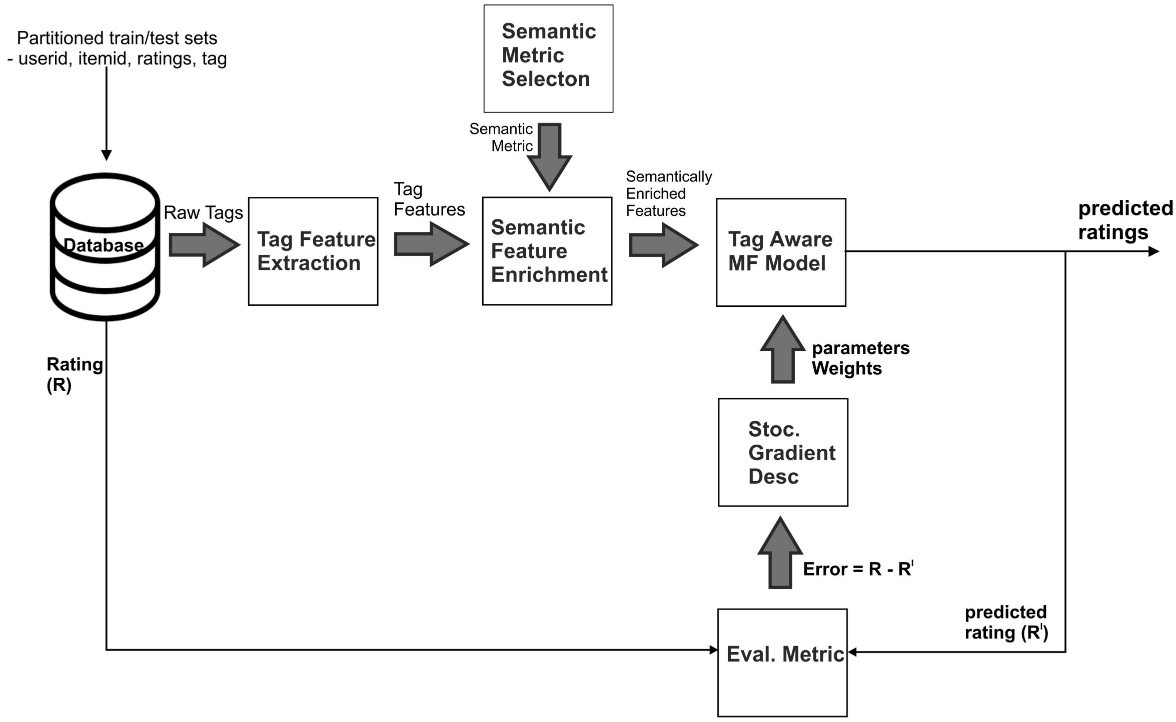


Fig. 3.4 Experimental set up for evaluating our model.

dation domains, namely the MovieLens[6] and the LibraryThing datasets[7]. Furthermore, a dataset from Yelp[8] was downloaded to evaluate the proposed cross-domain recommender model for POI recommendation in Chapter 6.

The general criterion for selecting datasets used in the experiment stage is that the data must have attributes that support evaluation of the cross-domain recommender systems. The performance of the proposed cross-domain models is evaluated against the recommendation goals for which the models were developed—addressing the posed research questions. Specific criteria for selecting the datasets used in our experiments are summarised below:

- **Tags for Inter-Domain Relations** - Cross-domain recommendations require establishing explicit relations between domains. The relations may be as a result of common content-based attributes between items or on rating-based relations between users/items. Inter-domain relations can also be formed from the aggregation of user profiles composed of social tags and semantic concepts. In the last case, there is no need for user or item overlap between domains, since tags and concepts are used as a common representation to establish relationships over multiple domains.

[6] The MovieLens dataset was downloaded from https://grouplens.org/datasets/movielens/
[7] The LibraryThing dataset was downloaded from http://www.macle.nl/tud/LT
[8] The Yelp dataset was downloaded from https://www.yelp.co.uk/dataset/challenge

- **Item Level Domains** - One important issue in the evaluation of cross-domain recommendation approaches is the non-availability of repositories with data to simulate different cross-domain scenarios. In order to address this limitation, a usual approach has been the splitting of datasets into subsets that are then considered as different domains, e.g. books with distinct category. We impose the stricter criterion of using datasets from domains that are different at the item level .i.e, the recommended items are not of the same type. For example, movies and books as opposing to genres of movies.

- **Baseline Models for Comparison** - Cross-domain recommendation is a challenging and still largely underexplored topic (Fernández-Tobías (2016)). There has not yet been a consensus among researchers on what overall evaluation of cross-domain recommendation should entail. As a result, we compared the performance of the proposed models against state of the art cross-domain recommender models using the same datasets in author's work.

3.3.2 Feature Extraction

In order to extract tag features from the datasets, we first carried out text preprocessing by tokenizing the tags, removing punctuations and stop words, and lemmatizing the tags to their dictionary form. After preprocessing the dataset, we further removed tags that were used by less than 1% of users in the domain or tags assigned to less than 1% of items in the domain. Generally, the relevance (importance) of a tag to a domain is estimated by the frequency of use of the tags by users and on items of the domains. We extracted features from the preprocessed tags by using the popular Term Frequency-Item Document Frequency (tf-idf) weighting scheme such as in Roelleke and Wang (2008) and Wu et al. (2008). Tfidf weight of a term in a document is defined by equation (3.4), where $tf_{t,d}$ is the frequency of the term in the document, df_t is the number of documents the term appear in, and N is total number of documents in the corpus.

$$tfidf_{t,d} = tf_{t,d} \times log(\frac{N}{df_t}) \quad (3.4)$$

Formally, users are represented by user profiles and items by item profiles. If the set of users and items are represented by user profiles U and I respectively, then the set of tags in each user or item profile can be denoted by $T(u)$ and $T(i)$, where $u \in U$ and $i \in I$.

Let $tf_{t_u,u}$ denote the term frequency of tag t_u obtained from the number of the times t_u is assigned by user u. Let $tf_{t_i,i}$ denote the term frequency of t_i calculated by the number of the tag assignments made to the items i. The document frequencies $df_{t_u,u}$ and $df_{t_i,i}$ are calculated by respectively counting the number of users that assigned tag t_u and the number of items the tag t_i is assigned to in the domain.

The tfidf weights of tag t_u where $t_u \in T(u)$ is computed according to equation (3.5); while weights of tag t_i, with $t_i \in T(i)$ is computed using equation (3.6).

$$tfidf_{t_u,u} = tf_{t_u,u} \times log\left(\frac{|U|}{df_{t_u,u}}\right) \tag{3.4}$$

$$tfidf_{t_i,i} = tf_{t_i,i} \times log\left(\frac{|I|}{df_{t_i,i}}\right) \tag{3.5}$$

Finally, the average tfidf is computed over all user and item profile to select the most relevant tags in the domains.

3.3.3 Semantic Metric Selection

The five measures [Lin measure (Lin 1998), Resnik measure (Resnik 1995), Leacock-Chodorow measure (Leacock and Chodorow 1998), Wu-Palmer (Wu and Palmer (1994)) and Jiang-Conrath measure (Jiang and Conrath 1997)] that we considered for evaluating semantic similarity/relatedness of *tag-concepts* have been compared by Budanistsky and Hirst (2005) using the survey of the human subjects above to determine how well the measures reflect human judgements of semantic relatedness. While correlating the measures with human judgements is the ideal way to evaluate a measure of similarity or semantic relatedness, in practice the small amount of data available (and only for similarity, not relatedness) is inadequate (Budanistsky and Hirst (2005)). Creating an all-encompassing set of concept pairs that cover the range of tags in our dataset and conducting surveys for responses from human subjects on the degree of similarity would be an enormous and resource intensive task.

Recent approaches such as in Bill et al. (2012) evaluate the various semantic similarity and relatedness measures on how well they predict if concept pairs are drawn from a single category (intra-category) or across different categories (inter-category). According to Bill et al. (2012), a basic aggregate test of automated similarity scores is that the average relatedness of terms within a category is higher than the average relatedness of that category's terms to the terms in a different category. We followed the suggestion of Budanistsky and

Hirst (2005) that one can not tell only by looking at the scores of a measure how good it is but it is when using the measures to perform a task that one can evaluate how well they work. As a result, we treat the metric selection process as a classification task in our experiments were the best at predicting whether tag pairs are drawn from a single domain or otherwise.

Furthermore, we conjecture that the ability to return the largest number of meaningful scores from a set of concept pairs is indicative of a semantic measure that can adequately cover the diverse range of word senses and concepts in the domains. When such measure is used to score similarity there can be higher confidence that the free-form nature of how users initially tagged items are preserved, and the semantic relationships of the concepts measured have not been limited. Our intuition aligns with the ideas considered in Bill et al. (2012) where "Concept Coverage" was utilised in evaluating semantic relatedness/similarity for performing queries in the Medical Dictionary for Regulatory Activities (MedDRA).

3.3.4 Semantic Enhancement

After selecting our semantic metric, we first set aside tags that are members of the intersection (i.e. common tags) between the auxiliary and target domains. The remaining tags were then processed for semantic relatedness by using the Lin[1] similarity metrics. We regarded two tags from the two different domains as being semantically related if their relatedness score is above a set threshold. The performance of the MF model on rating prediction accuracy after enriching tags in the intersect was tested at relatedness score thresholds from 10% - 90%.

WordNet comprises of words that encapsulate concepts which have distinct meanings and are linked together by different types of semantic relationships. The network of words/concepts that make up the WordNet lexical database can be analysed using standard metrics (e.g. Lin similarity metric) that can measure semantic similarity between words. Pairs of tags (each with a tag from the two domains) are mapped to corresponding concepts in the lexical database of WordNet. This is achieved by directly matching word form of the tags with word form of the WordNet concepts that gives the highest semantic similarity value.

3.4 Model Evaluation

The training phase of our model uses stochastic gradient decent typically utilized in learning the parameters required for predicting missing rating values in a high dimensional rating

[1] Lin metric was selected among others based on empirical experiments detailed in Chapter 4

matrix [Koren et al. (2009), Enrich et al. (2013), Manzato (2013), Shi et al. (2011), Fernández-Tobías and Cantador (2014)]. General inputs essential to SGD algorithm for learning the parameters used by the prediction models are unique users/items identifiers, rating values, the learning rate, regularisation parameter and latent factors. In cases where the prediction model takes additional inputs to make more accurate predictions, the SGD algorithm will consequently require unique identifies for the attributes.

According to Gantner et al. (2010), additional information about users (user attributes, e.g. gender, age, geographical location, occupation) and items (item attributes, e.g. genres, product categories, keywords) can be added to the latent features of the matrix before the dimensionality reduction process using SVD. Using SGD, the values of the prediction parameters are updated by moving in the opposite direction of the gradient. Let an actual rating value r_{ui} and a predicted rating value $\hat{r_{ui}}$. Let the error e_{ui} be the difference between actual and predicted rating value (r_{ui} - $\hat{r_{ui}}$). Let β be a model's parameter and α be the rate of the gradient's descent. In matrix factorisation, SGD finds a local minimum of an error function by updating the model's parameters after iterating over all known values of matrix elements at the rate α. Stochastic gradient descent shifts β in the direction of maximum descent of the local loss, given by its gradient:

$$\beta \leftarrow \beta - \alpha \left(\frac{\delta e_{ui}}{\delta \beta} \right).$$

The performance of the model we proposed to accomplish the objective above is measured based on its accuracy of predicting missing ratings (on test sets) in a set-up that simulates a cross-domain scenario.

3.5 Evaluation Metrics

An evaluation metric should measure how well a recommender system achieves the purpose for which it was developed. According to Kohavi et al. (2009) and Crook et al. (2009), there is higher confidence that the result of a recommender system will be useful when the metrics selected accurately reflect the specific goals of the recommender system being evaluated. In practice, improved customer satisfaction and higher profitability are general examples of objectives for implementing the system. On the other hand, specific objectives for using a particular recommendation approach may be to address the cold-start and sparsity problems. According to Quadrana et al. (2018), recommender systems are one of the most successful applications of data mining and machine learning technology in practice. Recommender systems can be considered as machine learning systems specialised to suggest products in commerce applications (Schafer et al. (2001)).

According to Hastie et al. (2009), the performance of a system that uses machine learning methods is measured with a loss function that penalises prediction errors. Machine learning methods applied to tasks like recommendation learn the parameters of a model in order to predict/estimate outcome(dependent) variables from predictor(independent) variables. There are two type of recommendation tasks that are distinguishable based on the types of the outcome variable and their measurement scale:

- *Rating Prediction Task* - requires that a numeric outcome variable, e.g. numerical ratings for an item on a scale 1-5. Rating prediction recommenders aim to accurately estimate ratings that users will give to items (e.g. movies, books, music) they are yet to utilise; and recommend items with the highest rating estimations.
- *Item Recommendation or Ranking Task* - requires that a nominal outcome variable, e.g. a list of items users may like to utilise or POIs they may be interested in visiting. Recommenders that address the item prediction task try to determine an ordered list of the items that are most likely to correspond with the preference of to the user.

Prior to selecting an evaluation metric, the purpose of a recommender system must be defined and *mapped* to one of the two tasks above. An evaluation metric is subsequently used to give an order to the performances of different recommendation models used to test how well the system has met the purpose. According to Gunawardana and Shani (2009), it is important that the metric match the task, to avoid an inappropriate ranking of the candidates.

3.5.1 Rating Prediction Accuracy

In order to address the question of how accurate the ratings estimated by a recommender are compared to the actual user ratings, two variant of rating prediction metrics can be considered. They are the mean absolute error (MAE) and root mean squared error (RMSE) and defined as follows:

$$MAE = \frac{1}{\mid R_{test} \mid} \sum_{r_{ui} \in R_{test}} \mid r_{ui} - \hat{r_{ui}} \mid$$

$$RMSE = \sqrt{\frac{1}{\mid R_{test} \mid} \sum_{r_{ui} \in R_{test}} (r_{ui} - \hat{r_{ui}})^2}$$

Where R_{test} contains the ratings in the test set reserved—i.e. as ground truth—for comparing with the estimated rating values during evaluation.

These two types of metrics are popular in the evaluation of recommender systems that use non-binary ratings. This is attributed to the ease of setting up experiments that use the metrics. However, the errors for RMSE are first squared before they the average is computed. As a result, RMSE gives a relatively high weight to large errors in comparison to the MAE metric. The MAE is a linear measurement which means that all the individual deviations in rating prediction are weighted equally in the average.

The models proposed in chapter four and five of this work where the predicted outcome variable (ratings) is of the numeric type were evaluated using a rating prediction metric. Specifically, the rating prediction accuracy of the proposed models was calculated based on the MAE and compared to accuracies of other state-of-the-art models in a cross-domain ‘experimental set-up.

3.5.2 Item Recommendation Accuracy

The accuracy of recommender systems that generate item recommendations relies on information on whether an item was selected or not selected by the user. In contrast to rating prediction task where the dataset is very sparse because users typically rate very small number of items, the feedback in item recommendation techniques are binary and dense since each item is either selected or not by the user. In addition, item recommendation techniques impose an order of preference on the set of items recommended to the user.

When measuring the accuracy of item recommendation methods, we are interested in finding out how many items from a previously held-out set (i.e. set aside as ground truth) gets returned as part of the recommendations. Let U be the set of all users in the recommender system. Let N be the total number of recommended items to user $u \in U$. The item recommendation accuracy is generally evaluated using two metrics based on:

- the ratio of returned items to the total number of recommended item N.
- the ratio of returned items in N to the number of items previously set aside as ground truth.

The former is known as Precision@N while the latter is Recall@N, and collectively referred to as Performance@N. In the experiment of chapter six where multi-category POIs processed as cross-domain items and POI check-ins as binary feedback, we test the performance of our proposed models when N = 5, 10, 20. The overall scores for these metrics are calculated by averaging the values for all the users set aside as ground truth.

$$Precision@N = \frac{1}{\mid U \mid} \sum_{u \in U} \frac{\mid Ret_u@N \mid}{N}$$

$$Recall@N = \frac{1}{\mid U \mid} \sum_{u \in U} \frac{\mid Ret_u@N \mid}{\mid Ret_u \mid}$$

In the formulae above, Ret_u is the set of items returned for user u, and $Ret_u@N$ is the set of items returned for user u that are in the top N positions of the ranked recommendations.

3.5.3 Precision and Recall for Rating Prediction

In the past decade, the rating prediction task has been the most popular approach taken to address the lack of personalisation in commercial platforms. This is likely due to the availability of datasets containing user preferences in the form of ratings, and academic competitions such as the Netflix Prize. In more recent years, however, this trend has taken a turn toward item recommendation task, as experimental evidence such as in McNee et al. (2006) and Cremonesi et al. (2011) have shown that more accurate rating predictions do not necessarily lead to higher user satisfaction. In rating-based recommender systems, it is common practice to recommend items that have the best rating predictions to users. As a result, metrics for item recommendation task such as precision or recall can be used to evaluate the accuracy of the items recommended. However, studies by Marlin and Zemel (2009) and Cremonesi et al. (2010) show that this approach does not result in recommendations that are clearly optimal. Furthermore, Steck (2010) argued that ratings are *missing not at random* and that this causes most rating prediction models to generate biased estimations.

In order to make the explicit rating values given to items to align more with the binary type used in item recommendation task and to avoid the *missing not at random* scenario, ratings can be treated as positive only. On the one hand, the ratings may be considered as binary based on whether the item has been rated or not rated. On the other hand, items with high estimated ratings, e.g. 4 and 5 on a scale of 1-5 can be considered as rated while the remainder are treated as not rated since they are not likely to be recommended to the user. Precision and recall can then be used to evaluate the performance of the cross-domain recommender systems. We leave this as a future direction for researchers who may be interested in evaluating rating-based cross-domain recommenders using precision and recall.

3.5.4 Item Diversity in Cross-domain Recommenders

There are different properties of a recommender system that can be measured as indicators of performance. On the one hand, the whole recommender system can be evaluated by measuring how changes in all system properties affect the overall user experience. On the other hand, the focus of evaluation may be on certain properties of the system. According to Shani and Gunawardana (2011), some of the properties can be traded-off, the most obvious example perhaps is the decline in accuracy when other properties (e.g. diversity) are improved. It is important to understand and evaluate these trade-offs and their effect on the overall performance. According to Adomavicius and Tuzhilin (2008), there is a trade-off between accuracy and diversity, because high accuracy may often be obtained by safely recommending to users the most popular ("bestselling") items, which can lead to the reduction in recommendation diversity, i.e., less personalized recommendations.

As diversity may come at the expense of other properties, such as accuracy, there is a need to compute curves to evaluate the decrease in accuracy against the increase in diversity Zhang and Hurley (2008). Diversity metrics evaluate how different items recommended to a user are with respect to each other. According to Shani and Gunawardana (2011), the most explored method for measuring diversity uses item-item similarity, typically based on item content. The diversity of a recommended list could be measured based on the sum, average, minimum, or maximum distance between item pairs, or measure the value of adding each item to the recommendation (Shani and Gunawardana (2011)).

Diversity is generally considered as the opposite of similarity (Shani and Gunawardana (2011)). There are certain scenarios when recommending a set of similar items may not be as useful for the user, because the user may require more time to go through the range of items in the list. As a result, the set of similar, redundant items in a recommendation list may not add much to the user's satisfaction. In these cases, the diversity of recommendations can be improved by exploiting the choices available in multiple domains. Multiple and cross-domain recommenders may provide better coverage of the range of preferences available for users. Winoto and Tang (2008) conjectured that, although cross-domain recommendations may tend to be less precise than single-domain recommendations, cross-domain recommenders will be more diverse, which may lead to higher user satisfaction and engagement. Subsequently, Li et al. (2009) proposed methods to effectively learn and transfer knowledge from the source domain to the target, and in alignment with the work of Fernández-Tobías et al. (2016) found that the quality of the recommendations improves when the involved domains are semantically more related.

Chapter 4

Cross-domain Recommender System with Semantically Related Tags

Advances in mobile and personal computing technology have led to the success and pervasiveness of social tagging systems. Nowadays, users can collect contents through several devices and label their collection with words (known as tags) of their choice for better organisation and future retrieval. Users can also choose to publish or share their collections with the assigned labels. Several online platforms have emerged to allow users to store their tagged collections such as photos in Flickr, songs in Lastfm, videos on YouTube, and news on Digg.

The platforms providing the tag sharing services can harness the vast collection of tags to provide item recommendation and further encourage collaboration among the community of users. The collaborative process of sharing and using the set of tags generates a tag structure (also known as folksonomy) in a social tagging system. This organisation of the tags is *"user-driven"*, and as a result, it can be a source of implicit user preferences. Features have been extracted from the collection of tags for use in models that are designed to improve user rating prediction and item recommendation accuracy.

In this chapter we used a recent matrix factorisation model for cross-domain collaborative filtering to investigate how rating prediction accuracy is affected as the number of related tags between two domains is increased. In Section 4.1, we highlight the benefits of positive-only data to prediction accuracy; and present our motivation for utilising social tags as additional positive-only data for cross-domain models. In section 4.2 we briefly review the most recent approaches that use tags for recommendation, focusing on those based on matrix factorisation to support cross-domain recommendations. In Section 4.3 we present our proposed approach to improving the performance of the cross-domain recommender model. Next, in Section

4.4 we describe the experiments conducted to evaluate the performance of the model at different thresholds of semantic relatedness, and in Section 4.5 we discuss the results of our experiments. In concluding our work in this chapter, we summarise findings and put forward our contribution to knowledge.

4.1 Introduction

User feedback is referred to as positive-only when the behaviour that creates the feedback is an implicit and binary action. Example of implicit feedbacks that are considered as positive-only include; web browsing history, videos watched, songs listened to, books checked out from a library, adds clicked on. Positive-only data can also be extracted from explicit data. As an example, explicit data such as ratings can give binary implicit data (rated or not rated) or explicit data such as tagging can yield implicit data (i.e tagged or not tagged). Koren (2008) has shown that it is possible to make rating prediction accuracy better by combining both implicit and explicit data.

In proposing the popular SVD++ algorithm, Koren et al. (2009) utilized the pseudo–implicit user feedback and reported that incorporating such simple implicit user feedback increases the prediction accuracy regardless of its binary nature. Social tags have also been used as positive-only feedback data for improving the performance of collaborative filtering algorithms. Prediction accuracy results of models in the work of Enrich et al. (2013) and Fernández-Tobías and Cantador (2014) show that social tags can be used as additional feedback for Matrix Factorization models in cross-domain collaborative recommender systems.

Cross-domain recommender systems are a new and evolving type of recommender systems. According to Fernández-Tobías and Cantador (2014), they exploit more exhaustive multi-domain user models that allow generating item recommendation spanning several domains. In this chapter, we review recent approaches to cross-domain recommendation, with a focus on those that utilise social tags to transfer knowledge from an auxiliary domain for enhancing rating predictions in a target domain. We particularly concentrated on extensions of the matrix factorisation approaches as originally proposed by Enrich et al. (2013) and Fernández-Tobías and Cantador (2014). The authors introduced new latent factors to add the contributions of users' and items' tags attributes to the predicted rating value.

4.2 Related Work

As detailed in definition 5 of chapter 3, SVD reduces a high dimensional matrix to its low dimensional equivalent. The process imposes an order on the low dimensional estimate in a manner that arranges the underlying features of the original dataset according to the effect they have on its variance (i.e. decreasing order from the feature that results to most variation to the one with the least). What makes SVD practical for NLP applications is that one can overlook variations that are below a particular threshold in order to massively reduce the data while still having the assurance that the primary relationships of interest have been preserved (Baker (2005)).

Another important advantage that SVD brings to collaborative filtering is its ability to use various types of data and adopt other requirements that may be specific to a recommender system (Koren et al. (2009)). This allows for the inclusion of other types of user feedbacks such as tags alongside the ratings that were explicitly provided. As a result, many authors have adapted its core principles to incorporate more information that enables a better understanding of item and users in recommender systems. In the following subsections, we discuss the approaches that have extended SVD to include implicit information and/or item and user metadata; and later show how our model builds on them.

4.2.1 SVD++

The accuracy of predicting missing values in a rating matrix can be improved by considering implicit user feedbacks which provide an additional indication of the users' preferences. In this type of MF model, user preferences are recorded as a combination of both the explicit and implicit feedback obtained respectively from the deliberate and indeliberate actions of users as they observe the items. The SVD++ model estimates the values in a rating matrix by supplementing the user factors which models how users rate with implicit feedbacks (i.e. what users rate) as shown in equation 4.1 below. Specifically, the implicit feedback used in this model is obtained from the rating users give to items. Koren et al. (2009) state that even when independent implicit feedback is absent, one can capture a significant signal by considering which items users rate, regardless of their rating value.

$$\hat{r}_{ui} = \vec{q}_i^{\,T} \cdot \left(\vec{p}_u + \mid N(u) \mid^{-\frac{1}{2}} \sum_{j \in N(u)} \vec{y}_j \right) \tag{4.1}$$

The SVD++ model was proposed by Koren (2008) and as it is with the regular SVD model, the parameters $q_i \in \mathbb{R}^k$ and $p_u \in \mathbb{R}^k$ represents item and user factors in a joint latent space. While $y_j \in \mathbb{R}^k$ represents the implicit feedback factor (i.e. rating factor). The parameter $N(u)$ is the set of items for which the user u provided implicit feedbacks, and k is the number of latent factors and the dimensionality of the space. According to Koren et al. (2009), the implicit feedback factor is added to equation 4.1 with a square root in order to stabilise the variance of the factors across the range of observed rating values in $\mid N(u) \mid$.

An objective function denoted hereafter as the error function is derived from the difference between the predicted ratings and the actual values of the known ratings in the original matrix.

$$e_{ui} = r_{ui} - \hat{r_{ui}} \tag{4.2}$$

The best value for the parameters of equation 4.1 is achieved when the error is at its minimum value. The predicted value can be lower or higher than the real value, and as a result, the square of the difference as in equation 4.3 is considered during optimisation.

$$\varepsilon = \sum_{u,i \in R} (r_{ui} - \hat{r}_{ui})^2 \tag{4.3}$$

When equation 4.1 is substituted into 4.3 above and differentiated we can obtain estimates for weights of all the model's parameters needed to make rating predictions. An iterative/incremental learning process known as stochastic gradient decent popularised by Funk (2006) during the Netflix competition has proven to be successful at estimating the parameters.

$$\varepsilon(p,q,y) = \sum_{u,i \in R} \left(r_{ui} - q_i^T \left(p_u + \mid N(u) \mid^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right) \right)^2 \tag{4.4}$$

4.2.2 SVD++ with Metadata Inclusion

One limitation of implementing SVD++ technique in recommender systems is its inability to benefit from cross-domain based collaborative filtering. This is due to the domain-specific nature of the type of feature factor (i.e. implicit rating factor) being introduced to supplement the user factor in the model equation. Since the rating value that users give to items are specific to the user-item pair, they are therefore disjoint across domains. However, the context around the moment a rating is given can be captured by the metadata (e.g. tags) the user provides during the rating process. Such metadata can occur across domains since the circumstance at the moment a user rates an item can reoccur in multiple domains. The main assumption to be considered according to Enrich et al. (2013) is that as a set of common tags that are associated with high ratings exist between the domains. Experimental results from the works of Enrich et al. (2013) and Fernández-Tobías and Cantador (2014) are evidence of the potential of tag-based cross-domain recommender system.

In the model proposed by Enrich et al. (2013), tag factors replace the implicit rating feedback factor in the SVD++ model. The tag factors are added to modify the item factors portion of the SVD++ model as shown in equation 4.5 and then combined with user factors to compute rating estimations.

$$\hat{r}_{ui} = p_u \cdot \left(q_i + \frac{1}{\mid T_R(i) \mid} \sum_{t \in T_R(i)} y_t \right) \tag{4.5}$$

The latent variable $y_t \in \mathbb{R}^k$ represents the feature factor from the metadata (tags). $T_R(i)$ is the set of all the relevant tags assigned by the user community to item i. The dimensionality k of the space is the number of latent factors considered for the items and users vectors.

In order to fully exploit the preferences of users as indicated by the tags they assign, Fernández-Tobías and Cantador (2014) proposed to extend the model by adding a set of latent variables $x_u \in \mathbb{R}^k$. The intentions of Fernández-Tobías and Cantador (2014) were to enrich the user's factors portion and account for the contributions of the tags given by users in the estimation of overall rating value. Specifically, two different set of tag factors (for users and items) were introduced to the original SVD++ model as shown in equation 4.6.

$$\hat{r}_{ui} = \left(p_u + \frac{1}{\mid T_R(u) \mid} \sum_{v \in T_R(u)} x_u \right) \cdot \left(q_i + \frac{1}{\mid T_R(i) \mid} \sum_{t \in T_R(i)} y_t \right) \tag{4.6}$$

The variable $x_u \in \mathbb{R}^k$ is the latent factor for tags that users have given to items, and the variable $y_t \in \mathbb{R}^k$ is the latent factor for tags that have been assigned to items. The parameter $T_R(u)$ is the set of all the tags assigned by the user "u" to any item, and $T_R(i)$ is the set of tags assigned to item "i" by any user.

4.3 Cross-Domain Recommender Model

The goal of our experiments in this chapter is to investigate the changes to rating prediction accuracy when the number of tags considered as common across two domains is increased. We used the model named "TagGSVD++" and proposed by Fernández-Tobías and Cantador (2014) as the MF model for our approach to cross-domain recommender system. Our justification for using TagGSVD++ as a framework for our cross-domain recommender model is based on its improved accuracy over other MF-based models as tested and reported by Fernández-Tobías and Cantador (2014). The authors concluded that exploiting additional tag factors and decoupling user and item components in a matrix factorisation process improves the accuracy of rating predictions. What is not yet clear is how the number of semantically related tags in the target and auxiliary domains contribute to the performance of the model on rating prediction task.

4.3.1 Model Formulation

We grouped the tags in both target and auxiliary domain into five sets. These set of tags are formally defined below, and the relationships they share are as illustrated in figure 4.1:

- **Unique Tag Sets:** this set contain tags whose character string are different in a target domain D_t and an auxiliary domain D_a. If tag $t_t \in D_t$ and $t_a \in D_a$ and characters of t_t are not the same as the characters of t_a, then we regard the pair of tags t_t and t_a as being unique. We denote these sets of tags as $T^u_{D_t}(t)$ and $T^u_{D_a}(t)$ for domains D_t and D_a respectively.
- **Common Tag Set:** the elements of this set are tags in both the auxiliary and target domain that have same string of characters (i.e. they occur word for word). If tags t_t and t_a in their respective domains are such that characters of $t_t = t_a$, then we regard the two tags as common across domains D_t and D_a. We denote the common tag set with $T_c(t)$ and $t_t = t_a \in T_c(t)$.
- **Semantically Related Tag Set:** this set contain tags related but have been assigned with different string characters in the target and auxiliary domains. Relatedness of a tag pair (t_t, t_a) is determined by Lin metric proposed by Lin (1998) and described in chapter two. The semantically related tag set is denoted as $T_r(t)$. If the tags "t_t" and "t_a" are from "D_t" and "D_a" respectively, and characters of $t_t \neq t_a$ but relatedness score of tag pair (t_t, t_a) is greater than threshold[1] "s" then we consider the tags as semantically related and $t_t \simeq t_a$. The tag pair $(t_t, t_a) \in T_r(t)$.
- **Adjusted Unique Tag Sets:** these sets are the subsets of unique tag sets $T_{D_t}(t)$ and $T_{D_a}(t)$. Adjusted unique tag sets are the sets that remain after $T_{D_t}(t)$ and $T_{D_a}(t)$ have been modified to reflect the transfer of tags considered to be related (after the relatedness measure by Lin metric) to the semantically related tag set $T_r(t)$. We denote

[1] The thresholds where set at 10 intervals on a scale between 0.1-1.0 for the Lin metric scores of tag pairs from tag and auxiliary domains.

adjusted unique tag set for domain D_t and D_a as $T^a_{D_t}(t)$ and $T^a_{D_a}(t)$ respectively. We calculate the size of the adjusted unique tag sets as :
$|T^a_{D_t}(t)| = |T^u_{D_t}(t)| - |T_r(t)|$
$|T^a_{D_a}(t)| = |T^u_{D_a}(t)| - |T_r(t)|$

- **Adjusted Domain Tag Sets:** these sets are superset in domains "D_t" and "D_a" that are the union of common tag set, semantically related tag set and the adjusted unique tag set. We denote adjusted domain tag set by $T_D(t)$ and calculate their size in D_t and D_a by:
$|T_{D_t}(t)| = |T_c(t)| + |T_r(t)| + |T^a_{D_t}(t)|$
$|T_{D_a}(t)| = |T_c(t)| + |T_r(t)| + |T^a_{D_a}(t)|$

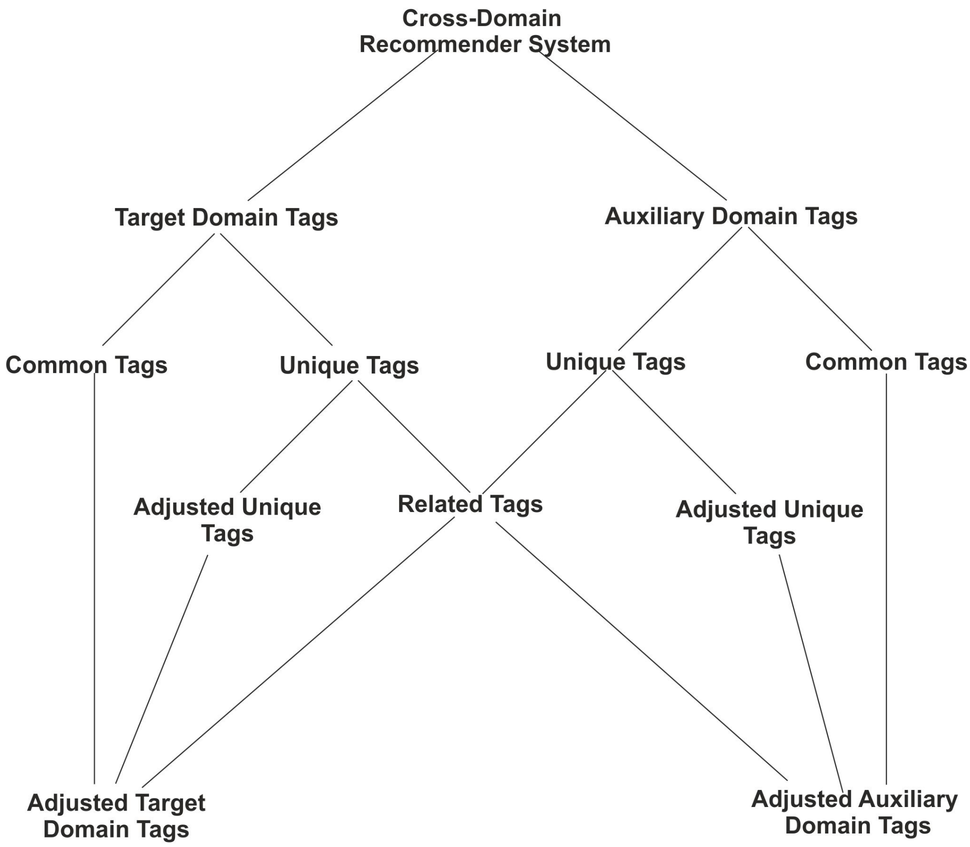


Fig. 4.1 Relationship between tag sets of a cross-domain recommender system.

We recall the model proposed by Fernández-Tobías and Cantador (2014) and discussed in section 4.2 as equation 4.7; and note here that our approach uses this same model. However, the set of tags we use as input to the model have been modified according to the relatedness of tags across the domains. These alternative sets of tags described in 4.3.1

allow us to investigate the contribution of semantically related tags to rating prediction accuracy at different relatedness threshold. We re-write the cross-domain recommender model proposed by Fernández-Tobías and Cantador (2014) and redefine the parameters to reflect the adjustment—after computing semantic relatedness—to tag sets of a target domain where rating prediction is to be estimated.

$$\hat{r}_{ui} = \left(\vec{p}_u + |T_{D_t}(u)|^{-1} \sum_{a \in T_{D_t}(u)} \vec{x}_a \right) \cdot \left(\vec{q}_i + |T_{D_t}(i)|^{-1} \sum_{b \in T_{D_t}(i)} \vec{y}_b \right) \tag{4.7}$$

The predicted rating of an item i for a user u in a target domain D_t is denoted by r_{ui} in equation 4.7. The user and item factors are represented by vectors $\vec{p}_u$ and $\vec{q}_i$. The user and item *tag-factors* are represented by $\vec{x}_a$ and $\vec{y}_b$ respectively. The set of tags that user u in D_t has assigned to any item is represented by $T_{D_t}(u)$. Similarly, the set of tags assigned by any user to item i in D_t is represented by $T_{D_t}(i)$ in equation 4.7.

We note here that sets $T_{D_t}(u)$ and $T_{D_t}(i)$ are the adjusted domain tag sets for user u and item i in the target domain D_t. If the tags assigned by user u to any item in the target domain does not belong to the set of semantically related tags, then our model behaves exactly like TagGSVD++ proposed by Fernández-Tobías and Cantador (2014). For such a case, the tag set $T_{D_t}(u)$ will be a union of common tag set and the unique tag set assigned by user u. Similarly, the tag set $T_{D_t}(i)$ will be the union of common tag set and unique tag set assigned to item i. The predicted rating value r_{ui} of item i to user u will not have any contribution from semantically grouped tags.

4.3.2 Estimation of Semantic Relatedness

In order to find tags in the semantically related set, we first paired tags in the unique tag set of a target domain D_t with tags in the unique tag set of an auxiliary domain D_a. For example, let $T^u_{D_t}(t)$ and $T^u_{D_a}(t)$ be unique tag set of D_t and D_a, if $t_t \in T^u_{D_t}(t)$ and $t_a \in T^u_{D_a}(t)$ then we estimated the semantic relatedness of tag pair (t_t,t_a) using the Lin semantic metric. As proposed by Lin (1998) and presented in equation 2.4 of chapter two, the Lin metric uses the taxonomic hierarchy of WordNet to compute the similarity between two words.

We regard the two tags in the pair (t_t,t_a) as being semantically related if their Lin metric score is higher than a set threshold. We substitute both tags t_t and t_a with the word that corresponds to their least common ancestor in WordNet taxonomy when their relatedness score is greater than the set threshold. The steps of algorithm 4.1 describe our approach of estimating semantic relatedness using Lin semantic metric. Tags that are selected as semantically related are finally added to the sets of common tags and adjusted unique tags to estimate the weights for the user and item *tag-factor* vectors $\vec{x_a}$ and $\vec{y_b}$.

Algorithm 4.1. Lin Semantic Metric for increasing size of semantically related tags between target and auxiliary domain

Input: Unique tag sets $T^u_{D_t}(t), T^u_{D_a}(t)$ in Target and Auxiliary Domain, Threshold r
Output: Adjusted Domain tag sets $T_{D_a}(t), T_{D_a}(t)$ in Target and Auxiliary Domain
procedure Relatedness
  for all $t_t \in T^u_{D_t}(t)$ **do**
    for all $t_a \in T^u_{D_a}(t)$ **do**
      $score_{t_t t_a} \leftarrow \text{LINsim}(t_t, t_a)$
      if $score_{t_t t_a} \geq r$ **then**
        $t_t = lcs(s_t, s_a)$
        $t_a = lcs(s_t, s_a)$
      end if
    end for
  end for
end procedure
function LINsim(t_t, t_a)
  $s_t \leftarrow$ synset of t_t from WordNet
  $s_a \leftarrow$ synset of t_a from WordNet
  $lcs(s_t, s_a) \leftarrow$ lowest common subsumer of $s_t$and s_a in WordNet taxonomy
  $IC(lcs(s_t, s_a)) \leftarrow$ Information content of the lowest common subsumer
  $IC(s_t) \leftarrow$ Information content of synset s_t
  $IC(s_a) \leftarrow$ Information content of synset s_a
  $sim \leftarrow 2 * IC(lcs(s_t, s_a))/(IC(s_t) + IC(s_a))$ # Computed according to Lin (1998)
return max(sim)

4.3.3 Estimation of Model's Parameter Weights

As described in the methodology section of chapter three, we used stochastic gradient descent algorithm to estimate the weights of parameters $\vec{p_u}$, $\vec{q_i}$, $\vec{x_a}$ and $\vec{y_b}$. The *regularized* squared error function of our model is presented in equation 4.8 and minimised using SGD to find parameter weights that best fits the model.

$$\varepsilon(p,q,x,y) = \left[r_{ui} - \left(\left(\vec{p_u} + |T_{D_t}(u)|^{-1} \sum_{a \in T_{D_t}(u)} \vec{x_a} \right) . \left(\vec{q_i} + |T_{D_t}(i)|^{-1} \sum_{b \in T_{D_t}(i)} \vec{y_b} \right) \right) \right]^2$$

$$+\lambda\left[\mid \vec{p}_u \mid^2 + \mid \vec{q}_i \mid^2 + \sum_{a\in T_{D_t}(u)} \mid \vec{x}_a \mid^2 + \sum_{b\in T_{D_t}(i)} \mid \vec{y}_b \mid^2\right] \quad (4.8)$$

A regularisation parameter "λ" is introduced into the squared error function as in equation (4.8) to control the estimated parameter values and to avoid "*overfitting*" the model. The value of λ is determined during experiments by cross-validation. In order apply SGD algorithm to the proposed model, the squared error function of equation 4.8 is first differentiated with respect to each of the factors p_u, q_i, x_a and y_b.

$$\frac{\delta e_{ui}}{\delta \vec{p}_u} = -2e_{ui}\left(\vec{q}_i + \frac{1}{|T_{D_t}(i)|}\sum_{b\in T_{D_t}(i)} \vec{y}_b\right) + 2\lambda \vec{p}_u$$

$$\frac{\delta e_{ui}}{\delta \vec{q}_i} = -2e_{ui}\left(\vec{p}_u + \frac{1}{|T_{D_t}(u)|}\sum_{a\in T(u)} \vec{x}_a\right) + 2\lambda \vec{q}_i$$

$$\frac{\delta e_{ui}}{\delta \vec{x}_a} = -2e_{ui}\left(\vec{q}_i + \frac{1}{|T_{D_t}(i)|}\sum_{b\in T_{D_t}(i)} \vec{y}_b\right) + 2\lambda \vec{x}_a$$

$$\frac{\delta e_{ui}}{\delta \vec{y}_b} = -2e_{ui}\left(\vec{p}_u + \frac{1}{|T_{D_t}(u)|}\sum_{a\in T_{D_t}(u)} \vec{x}_a\right) + 2\lambda \vec{y}_b$$

These derivatives are used during the training phase to simultaneously update the parameters (priorly initialised with random Gaussian values) by looping over the known rating values until its converges. The parameter "α" in the following update rules is known as the learning rate and used to determine how fast the error function converges to obtain the best values for the parameters.

$$\vec{p}_u \leftarrow \vec{p}_u - \alpha\left(\frac{\delta e_{ui}}{\delta \vec{p}_u}\right),$$

$$\vec{q}_i \leftarrow \vec{q}_i - \alpha\left(\frac{\delta e_{ui}}{\delta \vec{q}_i}\right),$$

$$\vec{x}_a \leftarrow \vec{x}_a - \alpha\left(\frac{\delta e_{ui}}{\delta \vec{x}_a}\right),$$

$$\vec{y}_b \leftarrow \vec{y}_b - \alpha\left(\frac{\delta e_{ui}}{\delta \vec{y}_b}\right).$$

The most optimal value for the learning rate "α" is determined by cross-validation during experimentation. The steps in algorithm 4.2 show how SGD is used for estimating the values of the parameters of our model. Specifically, steps 8 to 11 of algorithm 4.2 show how derivatives are used to update the model's parameters at each iteration until convergence is reached.

Algorithm 4.2. SGD algorithm for SemTagGSVD++

Input: Set of ratings r_{ui}, Adjusted Target Domain Tag Sets $T_{D_t}(u), T_{D_t}(i)$, Regularization parameter λ, Learning rate α, Number of latent factors k

Output: Weights of parameters $\vec{q}_i, \vec{p}_u, \vec{x}_a, \vec{y}_b$

Initialize $\vec{q}_i, \vec{p}_u, \vec{x}_a, \vec{y}_b$ with random values;

For *count = 1,....,#Iterations*

Foreach r_{ui} **do**

$$\hat{r}_{ui} \leftarrow \hat{r}_{ui} = \left(\vec{p}_u + |T_{D_t}(u)|^{-1} \sum_{a \in T_{D_t}(u)} \vec{x}_a \right) \cdot \left(\vec{q}_i + |T_{D_t}(i)|^{-1} \sum_{b \in T_{D_t}(i)} \vec{y}_b \right)$$

$$e_{ui} = r_{ui} - \hat{r_{ui}}$$

$$\vec{p}_u \leftarrow \vec{p}_u - \alpha \left(\frac{\delta e_{ui}}{\delta \vec{p}_u} \right)$$

$$\vec{q}_i \leftarrow \vec{q}_i - \alpha \left(\frac{\delta e_{ui}}{\delta \vec{q}_i} \right)$$

$$\vec{x}_a \leftarrow \vec{x}_a - \alpha \left(\frac{\delta e_{ui}}{\delta \vec{x}_a} \right)$$

$$\vec{y}_b \leftarrow \vec{y}_b - \alpha \left(\frac{\delta e_{ui}}{\delta \vec{y}_b} \right)$$

end

end

4.4 Experiment I

The first experiment was carried out to determine which of the standard semantic metric will be best for measuring semantic relatedness/similarity of tags between a target and auxiliary domain. After preliminary observations we were left with four out of the five semantic metrics, namely; Lin measure (Lin 1998), Resnik measure (Resnik 1995), Leacock-Chodorow measure (Leacock and Chodorow 1998) and Jiang-Conrath measure (Jiang and Conrath 1997). The fifth metric known as Hirst-St-Onge measure was described in section 2.9.1. The number of meaningful scores it returned was significantly smaller in comparison with other metrics. It also required a significant amount of system resource in to generate the similarity scores for the tag pairs as indicated by the low number of similarity score returned in the same computation time[1]. Consequently, the concept coverage for the Hirst-St-Onge measure was already too insufficient to compare with the rest of the metrics.

4.4.1 Datasets

Datasets from two different online rating systems and applications (Movie and Book) were used to represent a target and an auxiliary domain. In both datasets, the ratings are on a zero to five scale, with interval increments of 0.5. These datasets are commonly used in recommender system literature [Manzato (2013), Enrich et al. (2013), Fernández-Tobías and Cantador (2014) e.t.c]. The datasets are publicly available and have been made open-source for the research community.

- **MovieLens 10 Million Ratings:** This dataset has 10 million ratings and 100,000 tag assignments applied by 72,000 users on 10,000 movies. We note from the rating and tagging systems of MovieLens that the action of assigning tags to items was optional for users in the compilation of the datasets. Therefore, some items were rated but did not have tags assigned to them. Such ratings were excluded from the dataset since the aim of this work was to investigate the effects on rating prediction accuracy when the number of tags considered to be common with another domain is increased. This resulted in a selection of 44,804 rating data for movie items which had been given a single rating value and assigned one or more tags by users of the system.

- **LibraryThing 700K ratings:** The LibraryThing dataset had over 700 thousand ratings and 2 million tag assignments applied by over 7,000 users on 37,000 books. We preprocessed the dataset by removing ratings of book items which had no tag assigned for the same reason described for the MovieLens dataset. This resulted in 74,191 rating data on the book items which had at least one tag assigned.

The statistics above and other important details about the datasets from the two types of rating systems are as given in table 4.1. It is clear from table 4.1 that several tags are not a member of the intersect (i.e. not common) between both domains. Specifically, there are 7,737 tags of the MovieLens and 3,386 of the LibraryThing that are distinct across both datasets.

We preprocessed these tags by removing most of the tags that were assigned as sentences and compound words that do not occur in the database of Wordnet. This resulted in 2,126 tags

[1]Computational performance of the different Semantic Similarity Metric is shown in table B.1 of Appendix B.

Table 4.1 Tags from 24,564 MovieLens and LibraryThing ratings selected for semantic metrics evaluation

	MovieLens	LibraryThing
Users	2026	244
Items	5,086	12,554
Tags	9,059	4,708
Common tags	1,322	1,322
Tag assignments	44,804	74,191
Average ratings per user	12.12	100.67
Average tag assignment per users	22.11	304.06
Average tag assignment per items	8.81	5.91
% of tags overlapping with LibraryThing / MovieLens	13.81%	28.68%

for Movielens and 1,944 tags for LibraryThing. These tag pairs were used to decide which of the five semantic metrics will be given the most optimal result in increasing the number of commonly shared tags across the two domains.

4.4.2 Methodology

We measured the similarity between the distinct tags from the two datasets by using the Natural Language Tool Kit (NLTK)[2] to obtain numeric values that estimates the similarity according to the algorithms for each metric. The higher the value, the higher the similarity between the tags. In the absence of an extensive inter-human agreement on tag-pairs, we cast the problem of finding the best semantic metric as a classification problem. We took into consideration the knowledge that concepts pairs from the same dataset or domain are generally more related than otherwise. We evaluated the various semantic similarity and relatedness measures on how well they predict if concept pairs are drawn from the same dataset/domain (intra-domain) or different ones (inter-domain). The result of comparing the performance of each of the five different metrics is presented as the area under the curve using receiver operator characteristic (ROC) curves. As earlier indicated, the set of

[2]NLTK is a platform for building Python programs to work with human language data and it implements a variety of semantic similarity and relatedness measures based on information found in the lexical database WordNet.

concept pairs used in generating the curve varied from metric to metric. In each of the cases, the meaningful scores that were returned by the metrics were included as part of the total tag/concept coverage while the undefined results were excluded.

We used pROC15[3] library as methods for constructing ROC curves for each metric. A new variable called outcome was introduced and set to a value of 1 or 0 to respectively indicate if the concept/tag pair belongs to the same dataset/domain or not, as shown in table 4.2 This variable and the similarity scores between the tag pairs were used as input to pROC15 to create the curves.

Sensitivity measured the proportion of correctly classified positive scores (scores greater than the set threshold); while specificity was used to evaluate the proportion of correctly classified negative scores as the threshold is set to all the possible range of tag-pair similarity values. We obtained values for specificity and sensitivity at any threshold t with respect to the outcome variable [4] by following the equations below.

Sensitivity(t) is computed as;

$$\frac{(\text{No. of scores} \geq t \text{ with outcome } 1)}{(\text{No. of scores} \geq t \text{ with outcome } 1) + (\text{No. of scores} < t \text{ with outcome } 1)}$$

while the value of Specitivity(t) is obtained by;

$$\frac{(\text{No. of scores} \geq t \text{ with outcome } 0)}{(\text{No. of scores} \geq t \text{ with outcome } 0) + (\text{No. of scores} < t \text{ with outcome } 0)}$$

4.4.3 Results I

Sensitivity and specificity value for the each of the scores (i.e. set as thresholds) within this group were used to generate the ROC curve. The area under that curve (AUC) was interpreted as the overall effectiveness of the particular similarity metric. We carried out the process for the five different similarity metrics and observed the differences in each area under the curve. Figure 4.2 and table 4.2 respectively shows the resulting ROC and AUC for the Leacock-Chodorow measure, Wu Palmer, Resnik measure, Lin measure and Jiang-Conrath measures. The Lin measure was the best with an overall area of 0.639 under the corresponding ROC curve.

[3] An R package to display and analyse ROC curves

[4] Outcome has a binary value set to 1 for intra-domain and 0 for as inter domain

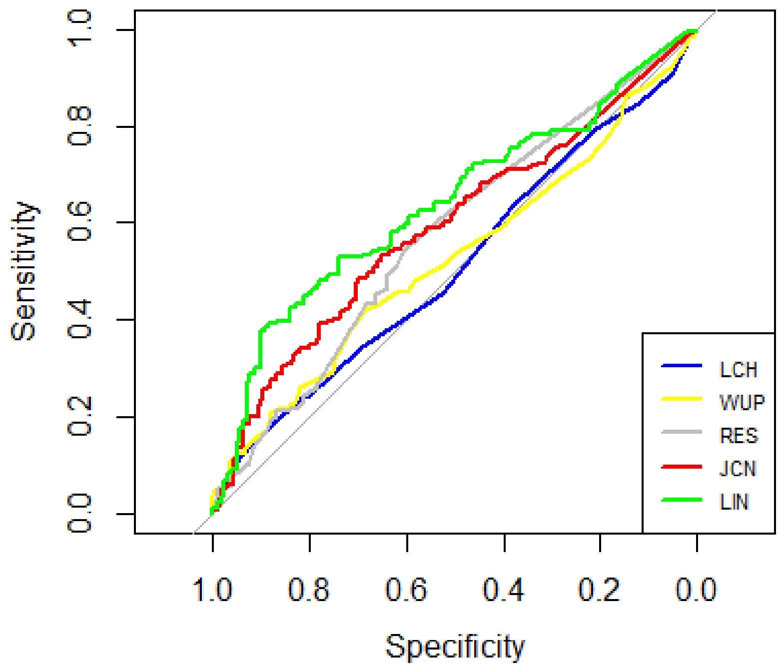


Fig. 4.2 ROC curve for Lin measure (LIN), Resnik (RES) measure, Leacock-Chodorow (LC) measure and Jiang-Conrath (JC) measures and Wu Palmer (WUP)

Table 4.2 Area under the curve for semantic metrics

Metric	Category	Outcome	Thresholds	Area Under Curve
	intra	1	682	
Leacock-Chodorow	inter	0	441	0.514
	intra	1	512	
Wu-Palmer	inter	0	401	0.527
	intra	1	265	
Resnik	inter	0	154	0.576
	intra	1	301	
Lin	inter	0	156	0.639
	intra	1	349	
Jiang-Conrath	inter	0	224	0.592

4.5 Experiment II

The goal of the second experiment in this chapter was to investigate the assumption that an increase in the size of tag intersect between two domains corresponds to an increase in rating prediction accuracy.

4.5.1 Methodology

Our methodology is based on the collective matrix factorisation approached proposed by Singh and Gordon (2008). One advantage of collective matrix factorisation models is that they simultaneously factorise several sparse matrices which share latent factors in the same vector space into a denser equivalent matrix. The benefit of this approach to our rating prediction task is that it allows us to exploit the semantic relationship that tags features in the target domain shares with tag features in an auxiliary domain for improved prediction accuracy. In a similar procedure to related works such as Shi et al. (2011), Enrich et al. (2013), Manzato (2013) and Fernández-Tobías and Cantador (2014), we prepared the target and auxiliary domains for collective matrix factorization by concatenating their rating matrices.

The data on users, items, ratings and tags were structured in a two-dimensional array (i.e. table), with the first-row index as the unique identifiers. We created a single matrix from the concatenation of the target and auxiliary rating matrices. The resulting matrix was then partitioned into different sets for evaluation as illustrated in figure 4.3. Our data partitioning process is similar to the methodologies of Sarwar et al. (2001), Enrich et al. (2013), Manzato (2013) and Fernández-Tobías and Cantador (2014). The authors minimised bias by setting aside a portion of the dataset for validation before training and testing their models.

The total number of ratings for our combined matrix is obtained from the sum of ratings from MovieLens and LibraryThing datasets (i.e. 17,465 + 17,465 = 34,930 ratings). We first shuffled the 17,465 rating data for the domain selected as the target domain and then divided it into ten parts for ten-fold cross-validation. For each round of validation, we used one of the divisions (i.e. 10% x 17,465 = 1,746[5] ratings) as the test and the remaining data as training and validation set. Out of the 90% (15,718 ratings) left, we set aside 20% (20% x 15,718 = 3,143 ratings) as a validation set to determine the optimal value for the model's secondary parameters (i.e the number of factors, learning rate and regularization parameter).

[5]The last division had five ratings more than the rest to cover for the remainder in the other divisions

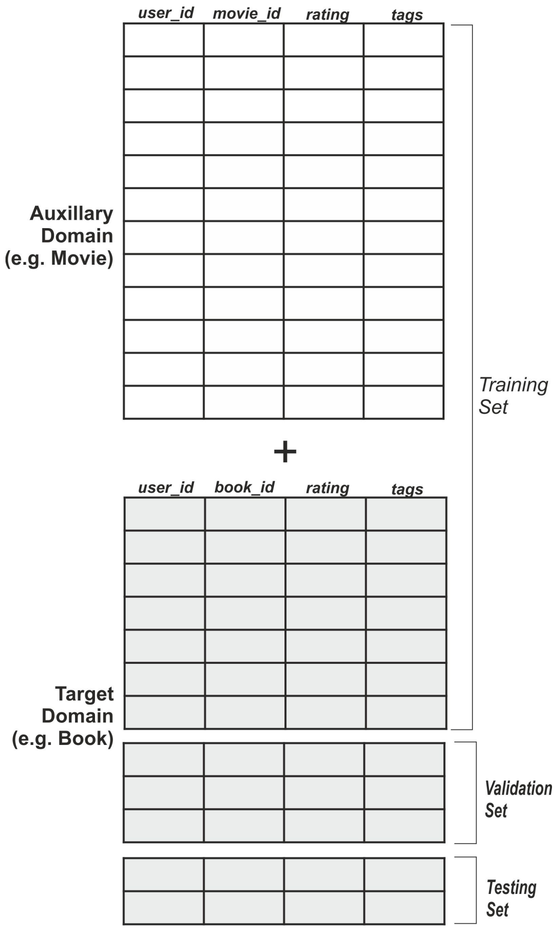


Fig. 4.3 Concatenating Target and Auxiliary Domain Matrices for collective matrix factorization.

The remaining ratings (i.e 12,576 ratings) in the target domain is added to the entire (17,465 ratings) from the auxiliary domain to make up the training set (12,576 + 17,465 = 30,040 ratings). In order to cross-validate rating values predicted by our models, the selection process was repeated according to the partitions after reshuffling the target domain dataset. We conducted ten rounds of validations and measured our model's performance by averaging the mean absolute error of the predicted ratings across all the ten validation folds.

4.5.2 Evaluated Thresholds

To determine how the accuracy of predicted rating values changes with increasing number of semantically related tag features, we considered ten different levels of semantic relatedness between unique tag features from the movie and book domains. As discussed in the methodology section of chapter 3, we used the Lin semantic metric to estimate the semantic relatedness of inter-domain tag features. According to Lin (1998), the Lin semantic metric gives scores for relatedness of word pairs in a range from 0.0 to 1.0. The least related word pairs have similarity scores closer to 0.0, while the most similar words score closer to 1.0. Words that are considered as fairly similar have their Lin semantic metric score around the centre (0.5) of the range. As a result, we used an interval scale of measurement to set the level of semantic relatedness of tag features from the movie and book domain. As a baseline, we used the cross-domain recommender model as proposed by Fernández-Tobías and Cantador (2014) where semantic relatedness of inter-domain tag features is not considered at all. The subsequent thresholds were set to nine different levels from 0.1 (10% semantic relatedness) to 0.9 (90% semantic relatedness) at intervals of 0.1.

As indicated in the preceding subsection, we set aside 20% of the target domain data in each of the ten divisions (i.e. folds) of the target domain dataset for validating the model and finding the optimal values for the model parameters. After several random approximation of the parameter values, we tuned the parameters by using grid search to find values for which the predicted ratings had the least mean absolute error. At every set threshold of semantic relatedness, the model's parameter values (i.e. number of features "f", learning rate "α" and regularization "λ") which gave the best prediction results were determined using grid search.

4.5.3 Results II

As detailed in the methodology section of chapter 3, Tf-IDF weighting scheme was used to extract tag features from each domain. The ranges of the Tf-IDF scores and the number of tag features obtained from the movie and book domain are shown in table 4.3.

Table 4.3 Number of Tag features from MovieLens and LibraryThing tag sets

	No. of Doc.	TF-IDF Range	No. of features
LT User tag profile as Doc.	243	0.0901 - 0.0007	1,197
LT Item tag profile as Doc.	11,285	0.1208 - 0.0048	108
ML User tag profile as Doc.	1,558	0.0364 - 0.0016	298
ML Item tag profile as Doc.	3,775	0.0489 - 0.0020	161

The total number of relevant tags in the target and auxiliary domains were calculated by combining tag features extracted by Tf-IDF. The number of relevant (key) tag features for both target and auxiliary domains as determined by Tf-IDF are shown in table 4.4 with their total ratings.

Table 4.4 Number of relevant (key) Tag features from MovieLens and LibraryThing tag sets

	No. of Key tag features	Total Rating
LibraryThing	1198	23,079
MovieLens	316	17,465

Details of MovieLens and LibraryThing dataset after selecting the most relevant tag features and matching the number of ratings in both domains is shown in table 4.5. There is a percentage difference of 21.97% in the ratio of common tags to relevant tags between MovieLens and LibraryThing dataset.

Table 4.5 Statistics of the MovieLens and LibraryThing datasets after selection of most relevant tags.

	MovieLens	LibraryThing
Ratings	17,465	17,465
Users	1,578	206
Items	4,163	9,848
Average ratings per user	11.06	84.78
Average ratings per item	4.20	1.77
Relevant Tag features	316	750
Common tags features	120	120
Ratio of common to relevant	16%	37.97%

Related tags pairs obtained from the Lin similarity metric are presented in table 4.6 for the nine different relatedness threshold. The threshold with Lin similarity metric score greater had the highest number with 47 related tag-pairs.

The percentages of related tags to overall relevant tags at the set relatedness thresholds are shown in table 4.7 for both domains.

Table 4.6 Number of tags pairs with scores greater than relatedness threshold and the number of most related tag pairs

Threshold (%)	Tag Pairs > Threshold	Related Tag Pairs
10	8,690	56
20	3,478	56
30	1,525	55
40	546	53
50	243	51
60	107	37
70	52	26
80	26	16
90	12	8

Table 4.7 Percentage tag overlap in MovieLens and LibraryThing with increasing tag relatedness

Threshold (%)	Adjusted Common Tags	ML Overlap (%)	LT Overlap (%)
10	176	23.47	55.70
20	176	23.47	55.70
30	175	23.33	55.38
40	173	23.07	54.75
50	171	22.80	54.11
60	157	20.93	49.68
70	146	19.47	46.20
80	136	18.13	43.04
90	128	17.07	40.51

The chat in figure 4.4 and 4.5 show difference between the ratio of common tags to relevant tags for MovieLens and LibraryThing at the nine different semantic relatedness thresholds.

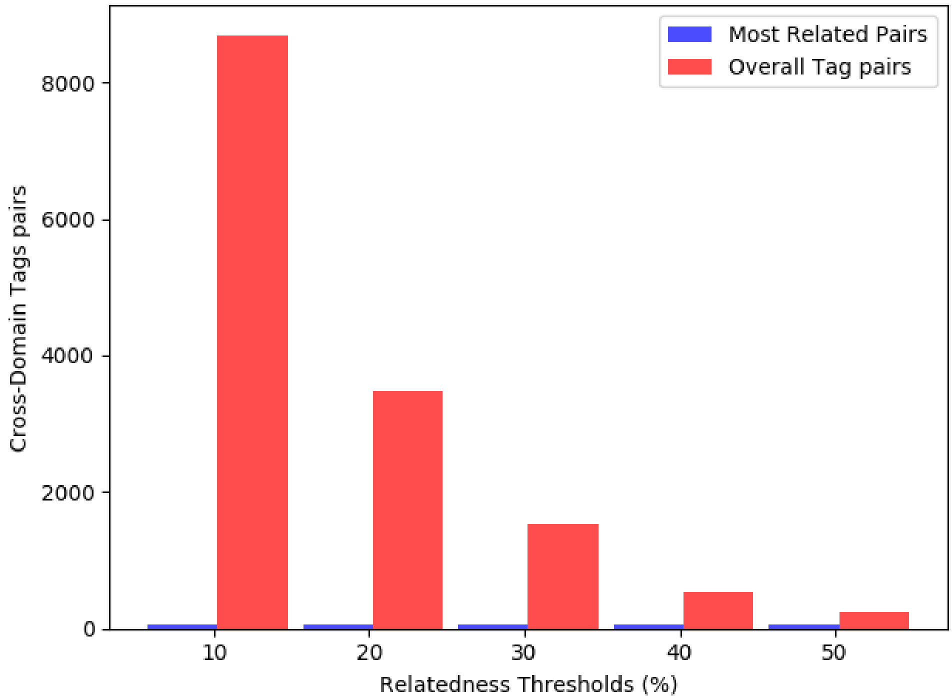


Fig. 4.4 Proportions of overall tag pairs to most related pairs at lower relatedness thresholds.

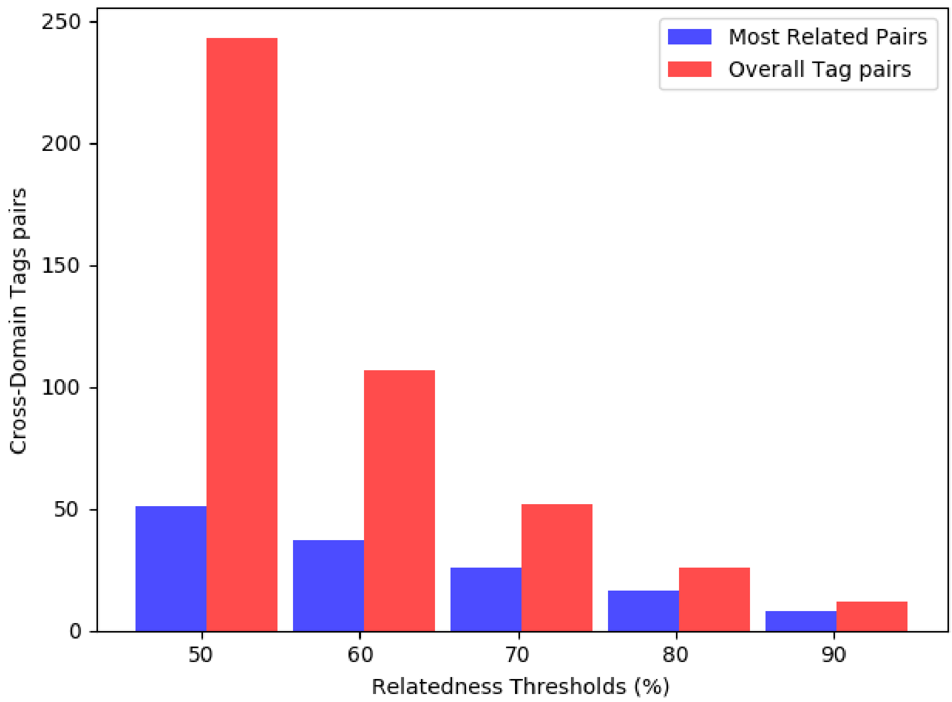


Fig. 4.5 Proportions of overall tag pairs to most related pairs at higher relatedness thresholds.

As described in subsection 4.3.3, we optimised the model's parameter at each set threshold of semantically relatedness using a grid search algorithm on the validation set. Average values of the learning rate α, the amount of regularisation γ, and the number of latent features k that resulted into the least mean absolute error is presented in Table 4.8. We evaluated

Table 4.8 Best parameter values for the model when LibraryThing(LT) is set as auxiliary domain and MovieLens(ML) is set as auxiliary domain and vice versa.

	LT → ML			**ML → LT**		
	k	α	γ	k	α	γ
10% Tag Relatedness	41	0.018	0.017	40	0.018	0.017
20% Tag Relatedness	40	0.018	0.015	42	0.017	0.017
30% Tag Relatedness	41	0.020	0.016	41	0.018	0.015
40% Tag Relatedness	41	0.018	0.016	40	0.016	0.016
50% Tag Relatedness	40	0.020	0.015	40	0.016	0.015
60% Tag Relatedness	40	0.018	0.015	41	0.017	0.015
70% Tag Relatedness	40	0.019	0.015	40	0.016	0.016
80% Tag Relatedness	40	0.019	0.015	40	0.017	0.015
90% Tag Relatedness	40	0.018	0.015	40	0.017	0.015

our cross-domain recommendation approach in two settings; using MovieLens as auxiliary domain and LibraryThing as the target domain, and then alternatively using LibraryThing as auxiliary and MovieLens as target domain. An observation of the prediction is recorded for all the ten thresholds of semantic relatedness. Results of the rating prediction accuracy measured in Mean Absolute Error with increasing number of SGD iterations on the training set is shown in figure 4.6 and 4.7.

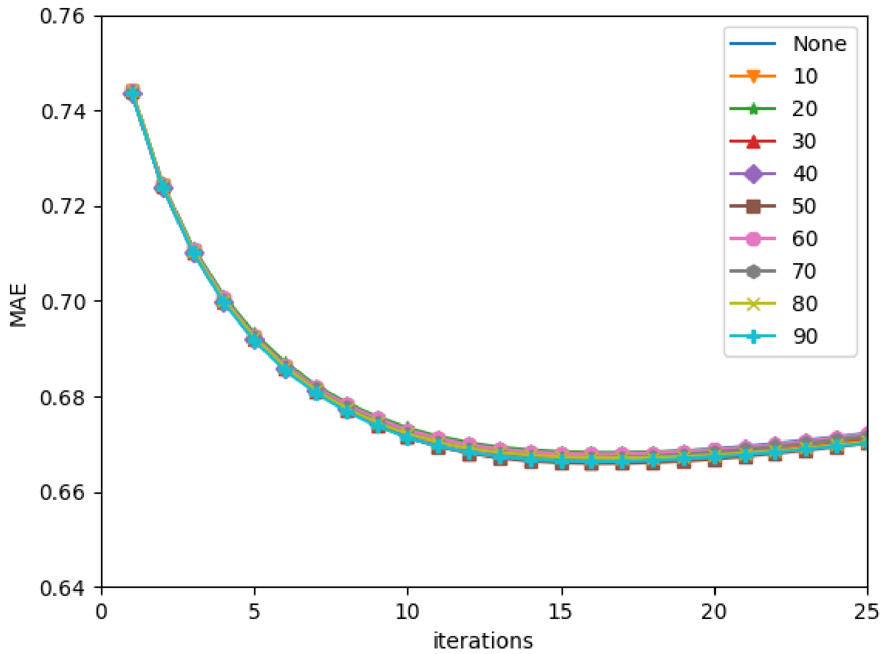


Fig. 4.6 Average prediction error for 10 different semantic relatedness thresholds for TagSVD++ model with MovieLens as target domain and LibraryThing as auxiliary domain.

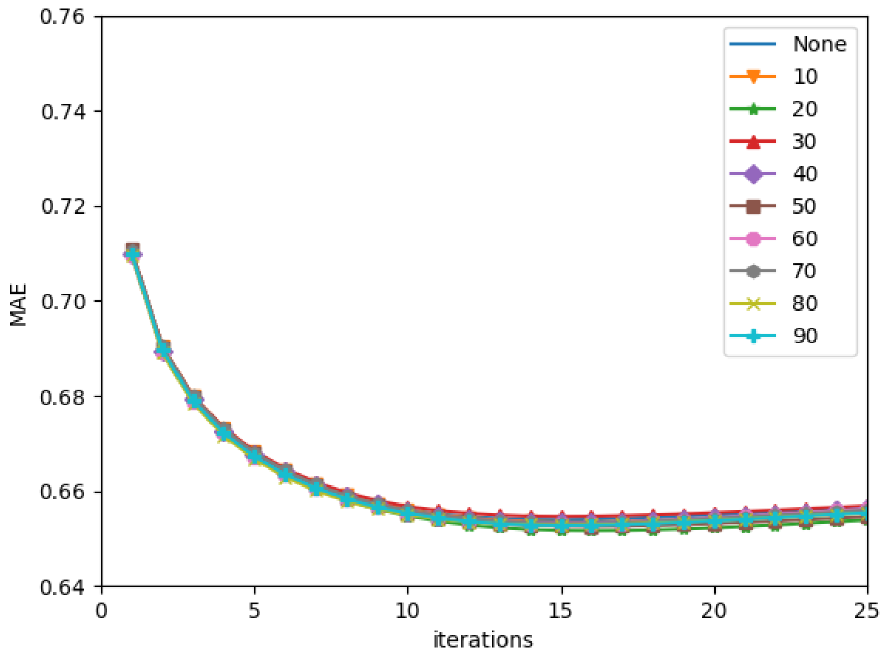


Fig. 4.7 Average prediction error for 10 different semantic relatedness thresholds for TagSVD++ model with LibraryThing as target domain and MovieLens as auxiliary domain.

4.6 Discussion

The first question in this research was asked to investigate if increasing the number of related social tags between a target and auxiliary domain improves the accuracy of rating prediction in the target domain. Our first objective was to use a Tf-IDF weighting technique to independently select the most relevant tag features from both domains. As expected, the weight of tag features are higher when item profiles are used as documents than when user profiles are used as documents for the Tf-IDF weighting technique. This result can be explained by the fact that there are more items than users in both domains (i.e. more item profiles used as document corpus for calculating Tf-IDF weights). Also, there are more tags assigned by each user than there are tags assigned to each item as seen in the corresponding number of tag assignments in table 4.4.

The total number of relevant tags in each domain was the result of combining tag features selected when the user tag profiles were used as documents and tag features selected when item tag profiles were used as documents. The total number of features recorded in table 4.3 shows a general increase by one tag feature for LibraryThing and eighteen tag features for MovieLens after the combination. This result corroborates the fact that there are less distinct tags in LibraryThing and they are also used more repeatedly than in MovieLens.

In order to avoid bias due to the difference in the number of rating data used in the training and testing process, we constrained the number of ratings of LibraryThing dataset to exactly match that of MovieLens. This decision aligns with related works such as Enrich et al. (2013) and Fernández-Tobías and Cantador (2014) where bias was also avoided in the same manner. As a result, only the first 17,465 ratings of LibraryThing was selected from the 23.079 recorded in table 4.3 for evaluating the model's prediction performance at the different thresholds. Details of the key statistics of the dataset after tag feature selection are shown in table 4.4.

The number of unique MovieLens and unique LibraryThing tag features is calculated from the difference between their relevant tag features and common tag features as presented in table 4.4 (i.e. 316-120 = 196 for MovieLens and 750-120 = 630 for LibraryThing). Related cross-domain tag features are then obtained from the pair of unique MovieLens tag features and LibraryThing tag features that have a similarity score above the set relatedness thresholds.

As expected, the number of cross-domain tag pairs obtained from the Lin semantic metric for each threshold was inversely proportional to the cut-off value at the threshold as in table 4.3. One interesting observation made about lower thresholds was that the number of cross-domain tag pairs with the highest Lin semantic metric score (i.e. the most related pairs) was significantly lower than the overall number of tag pairs that were above the thresholds. This behaviour as seen in table 4.5 may be explained by the dependence of Lin semantic metric on the taxonomic structure of WordNet. The formula for Lin semantic metric was introduced in equation 2.4 and applied in algorithm 4.1.

The dividend in the Lin metric formula is a function of the information content of the lowest common ancestor (LCA) concept that subsumes the cross-domain tag pairs. Previous studies such as Miller (1995), Resnik et al. (1999), Lin (1998), Fellbaum (2005) and Harispe et al. (2013) have shown that the information content of a concept has direct proportionality to it's depth in WordNet's taxonomic structure. LCA concepts are less informative when they are higher and closer to the root concept in the hierarchy of WordNet. As a result, the Lin semantic metric score is generally low for cross-domain tag pairs subsumed by LCA that is higher in WordNet's hierarchy. The set of cross-domain tag pairs with Lin semantic scores that are above lower thresholds (e.g. 10%, 20%, 30%, 40% and 50%) have a higher number of generic LCA than those above higher thresholds (e.g. 60%, 70%, 80%, and 90%). This may account for the large difference between the total number of cross-domain tag pairs and the highest scoring ones at lower thresholds in figure 4.3 compared to figure 4.4 at higher thresholds.

As shown in graphs of figures 4.5 and 4.6, the curves for TagSVD++ model at different thresholds of tag relatedness closely follow a similar path in a plane where MAE and SGD iterations represent the y and x coordinates. The "None" curve in figure 4.5 and 4.6 represented the baseline approach which was the TagGsvd++ model as proposed by Fernández-Tobías and Cantador (2014), i.e. without the addition of related tag features. There are no clear differences in the MAE of the baseline and others approach which had an increased number of related tag features. A Wilcoxon signed rank test at the 95% confidence level showed that the small differences observed in the MAE at different relatedness thresholds are not statistically significant.

Another important finding from the experiment was that the values of MAE are generally small for all iterations when LibraryThing is used as a target domain and MovieLens as the auxiliary domain. It is interesting to compare this observation with results reported in Enrich et al. (2013) and Fernández-Tobías and Cantador (2014) where the opposite was the case. In contrast to our result the authors found that the MAE was lower when MovieLens is used as a target domain and LibraryThing as the auxiliary domain. The contradiction can be explained by the Tf-IDF feature extraction process which switched the dataset with the higher number of tags from MovieLens in table 4.1 to LibraryThing in table 4.4. The authors' result aligns with ours when the dataset with the higher number of tags is used as the target domain.

4.7 Conclusion

In this chapter, our first experiment was to evaluate a set of semantic similarity metrics in order to choose the most appropriate for determining the semantic similarity of tag pairs across two different domains. These metrics, namely; Leacock-Chodorow, Wu Palmer, Resnik, Lin and Jiang-Conrath where applied on a list of distinct words/concept from music and a book domain and their performance was assessed based on how well they classified the concepts according to their source. The result from two different experiments confirmed that the Lin measure was the best. Sensitivity and Specificity of the similarity scores returned by the algorithms were used to generate ROC curves, and the size of the areas under the curve was taken as an indication of how efficient the algorithms are at classifying the concepts. Specifically, we showed that the Lin semantic measure is the most semantically effective in establishing concept similarity between a movie and book domain.

The second experiment in this chapter was undertaken to investigate the effect of increasing the number of common social tags between a target domain and an auxiliary domain. The cross-domain model proposed by Fernández-Tobías and Cantador (2014) was selected for testing different percentages of related tag overlap because of its improved rating prediction accuracy over other similar models. Common tags between a target and auxiliary domain are adjusted to include tags that show some level of relatedness as measured by the Lin semantic metric. On the basis of the insignificant differences in MAE when the number of related tags in a cross-domain setting is varied, we summarise the result of our observation as follows:

- The prediction accuracy of TagGSVD++ model is generally lower for a cross-domain set up where the domain with more number of tag features is used as the target domain.
- The prediction accuracy of TagGSVD++ model dependent depends on the number of tag features in target domain rather than on the related tag features shared with an auxiliary domain.

According to Fernández-Tobías (2016), the performance of a cross-domain recommender system is mainly affected by three parameters: the overlap between the source and target domains, the size of the target user's profile, and the density of the target domain data. As a result, the evaluation of a cross-domain recommendation approach mostly considered the sensitivity of the underlying recommendation algorithm with respect to these three parameters (Fernández-Tobías (2016)). In this chapter, we have evaluated the performance of cross-domain recommender systems on rating prediction task based on the size of overlapping tags—inter-domain tags. The experiment did not detect any evidence that increases in the number of related tags between a target domain and auxiliary domain result in better rating prediction. However, we have shown that even though increasing the size of inter-domain tags may not be helpful, the semantic enhancement of the tags did not impact negatively on the performance of the cross-domain recommender model.

Chapter 5

Semantically Enhanced Cross-Domain Recommender Models

The application of artificial intelligence in different systems is becoming more prevalent in everyday life. As a result, ensuring the A.I. can complement human cognition and reproduce useful results for Natural Language processing has become a necessary task for the A.I system expert. Estimation of semantic relatedness and similarity are some of the well-known challenges in Natural language processing. Similarity and relatedness are widely used today to determine the strength of the semantic relationship between entities of various types, e.g. words, sentences and concepts. The importance of semantic relatedness and similarity is also evident in the design of cross-domain recommenders where there may be requirements to compare users and/or items attributes in different domains.

In measuring semantic similarity, two approaches have been widely applied in Natural Language Processing. Word vector representations also known as word embeddings are corpus-based models and they rely on co-occurrence of words in very large corpus for estimating the semantic relationship between words. They differ from their count-part which are knowledge-based models that depend on the structural knowledge in a taxonomy (e.g. depth, path length, common ancestor) and statistical information content (corpus-IC).

In this chapter, we introduce a cross-domain recommender approach that uses the union of unique and semantically related tags for predicting ratings for active users in a system with cold start and sparsity conditions. We investigate the use of knowledge-based (WordNet) and corpus-based approaches in determining the semantic relatedness between social tags in cross-domain recommender systems.

5.1 Introduction

Several research work such as Agirre et al. (2009) and Bill et al. (2012) which involve measuring semantic relatedness have been traditionally based on the hierarchical network of lexical databases such as WordNet and encyclopaedic knowledge bases such as DBpedia. Current approaches to measuring semantic relatedness now take advantage of effective semantic knowledge transfer from pre-trained data to test data. These transfer approaches are enhanced by recent advances in deep learning techniques such as Neural Networks (NN). Specifically, word vector representations (also known as word embeddings) that are based on NN approaches have become popular in NLP tasks. Many studies have demonstrated the effectiveness of word embeddings in capturing both syntactic and semantic information. The most popular model used for word embeddings is known as "Word2Vec" and was proposed by Mikolov et al. (2013). The popularity of Word2Vec and similar word embeddings has been linked by Kiela et al. (2015) to their applicability to a variety of tasks without much adaptation.

Several lines of evidence such as in Faruqui et al. (2014), Liu et al. (2015) and Trask et al. (2015) have however shown that words are not disambiguated during the training phase of Word2Vec. Together, these studies indicate that the semantic relatedness and results of other NLP tasks from Word2Vec may be improved upon. On the contrary, WordNet and encyclopaedic knowledge bases have structures that clearly identify the meaning of words. However, WordNet and similar ontologies are known to be limited in the coverage of words in their taxonomy. Researchers such as Faruqui et al. (2014) and Kiela et al. (2015) have attempted to address the lack of disambiguation and coverage in the respective models by refining vector space representations using relational information from semantic lexicons. The model proposed by the authors encourage related words to have similar vector representations.

Several "blends" (during or post-training) of the two models referred to as retrofitting or specialization have been proposed to address the limitations of both models in NLP tasks. The general assumptions of approaches that refine word embeddings is that word vectors can be "prompted" to align into a particular direction by integrating additional semantic data source. Semantic data sources that have been considered in related researches include semantic lexicons, which provide information about the semantics of words, typically by identifying synonymy, hypernymy, hyponymy. Paraphrase relations have also been used to refine word vectors that are trained solely on unstructured data from large corpora.

5.2 Related Work

As extensions to the popular SVD++ models, several recommender models have been proposed to utilise different types of metadata in order to improve rating prediction accuracy. For example, the evolving taste of users for items has been modelled into SVD++ by incorporating parameters account for the contribution of different time periods. In the work of Koren (2009), a model that accounted for such temporal dynamics was proposed and referred to it as *timeSVD++*. In addition, relational attributes obtained from the social affiliation of users have also been added to SVD++ to model the influence of trusted users. Social trust information has be incorporated by Guo et al. (2015) in their *trustSVD++* model to supplement explicit and implicit influence of item ratings in SVD++ model. All the aforementioned approaches to extending the popular SVD++ are proposed for use in single-domain collaborative recommender systems. In the following section, we briefly review works that use social tags as metadata to extend SVD++ for cross-domain recommender systems.

5.2.1 ItemRelTags Model

This model as proposed by Enrich et al. (2013) focuses on predicting ratings for users in cold start situations where very little information is known about users. The model extends the traditional SVD model by considering metadata about the items in a system. Specifically, the influence of latent factors that are associated with the item metadata replaces those that are obtained from implicit feedbacks as in the standard SVD++ model. As can be seen in the following equation for the model, the contribution of the latent factors associated with the item metadata does not depend on the user portion of the model. Consequently, the ItemRelTags model can be exploited in the rating predictions for new users for whom tagging information is not yet available. The model therefore potentially delivers better predictions results over the standard SVD++ model with the advantage of been implementable in a cross-domain setting where tags get associated with ratings.

$$\hat{r}_{ui} = p_u \cdot \left(q_i + \frac{1}{\mid T_R(i) \mid} \sum_{t \in T_R(i)} y_t \right)$$

The latent variable $y_j \in \mathbb{R}^k$ represents the feature factor from the item metadata. $T_R(i)$ is the set of all the relevant tags assigned by the user community to item i. The dimensionality k is the number of latent factors of the space.

5.2.2 gSVD++ Model

The gSVD++ algorithm further extends SVD++ by considering information about the items' attributes in addition to the users' implicit feedback. The main strategy according to Manzato (2013), is based on exploiting implicit feedbacks from users by considering not only the latent space of factors associated with the user and item but also on the available metadata associated with their respective contents. The model shows the effectiveness of incorporating metadata into a latent factor model such as SVD++. As shown in the equation below, the item factor portion is enhanced with latent factors associated with the metadata, while the influence of latent factors associated with the implicit feedback about items that users rated is added to the user factor portion.

$$\hat{r}_{ui} = \left(q_i + \mid G(i) \mid^{-\beta} \sum_{g \in G(i)} x_g \right) \cdot \left(p_u + \mid N(u) \mid^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right)$$

The gSVD++ model adds a new set of latent variables $x_g \in \mathbb{R}^k$ for item metadata to augment the item factors. The set $G(i)$ has elements that represent the attributes related to the content of the items, e.g. comedy and drama in the case of movie genres. In our experiment, the elements of the set $G(i)$ has been used to represent all tags assigned to items. The parameter β is set to 1 when the $G(i) \neq \emptyset$, and 0 if otherwise (Manzato (2013)).

5.2.3 TagGSVD++ Model

The recommendation strategy behind this model is based on the assumption that ItemRelTags model does not fully exploit the user's preferences expressed in the tags assigned to other items. According to Fernández-Tobías and Cantador (2014), the gSVD++ algorithm above was adapted by introducing an additional set of latent variables that enrich the user's factors and better represent the effect of her tags in the rating estimation. The model identifies two different sets of tags for users and items and adds the latent factors associated with both to the corresponding item and user portions of the SVD model as shown below. Similarly to the case for the two models whose description precedes this, TagGSVD can also be implemented in a cross-domain setting because of the dependency they have with tags whose occurrence can be in different domains.

$$\hat{r}_{ui} = \left(p_u + \frac{1}{\mid T_R(v) \mid} \sum_{v \in T_R(v)} x_v \right) \cdot \left(q_i + \frac{1}{\mid T_R(i) \mid} \sum_{t \in T_R(i)} y_t \right)$$

The parameter $x_v \in \mathbb{R}^k$ is the latent factor for all tags that a user have given to items. The variable $y_t \in \mathbb{R}^k$ is the latent factor for tags that have been given to an item by all users. The parameter $T_R(v)$ is the set of all the tags assigned by the user to any item i. $T_R(i)$ is the set of all the relevant tags assigned by the user community to item i. The dimensionality k is the number of latent factors of the space.

5.3 Proposed Model

In this chapter, we group the tags in the target and the auxiliary domains into a unique set and a related set as described below. We add a new latent factor $\vec{x_c}$ to incorporate the influence of related tags to the rating prediction.

- **Unique Tags:** For an active user u, let $T(u)$ be the set of tags that u assigned to any item and let $T(u)$ and $T(i)$ be the set of tags assign by any user to item i.
- **Related Tags:** these sets are selected after computing the semantic relatedness between all tags in the unique tag set. The sets are denoted by $T^c(u)$ and $T^c(i)$ for users and items factors in the proposed model. These sets contain the top N tags that are most related to the tags in the unique set.

The rating prediction equation for the proposed model is presented in equation 5.1 with each parameter described in the following section to formalise the model.

$$\hat{r}_{ui} = \left(\vec{p_u} + |T(u)|^{-1} \sum_{s \in T(u)} \vec{x_s} + |T^c(u)|^{-1} \sum_{c \in T^c(u)} \vec{x_c} \right) \cdot \left(\vec{q_i} + |T(i)|^{-1} \sum_{t \in T(i)} \vec{y_t} + |T^c(i)|^{-1} \sum_{c \in T^c(i)} \vec{y_c} \right) \tag{5.1}$$

The vector $\vec{p_u}$ is the user factor vector; while $\vec{x_s}$ is the user-tag factor vector. The tags in set $T(u)$ are those that user u assigned to any item. The new latent vector $\vec{x_c}$ is introduced here as the *"user-related-tag"* factor vector, while the corresponding set $T^c(u)$ contain tags selected as semantically related to those assigned by user u to any item.

Similarly, the vector $\vec{q_i}$ consist of the item factor vectors; while $\vec{y_t}$ consist of the item-tag factor vector and $T(i)$ is the set of unique tags that are assigned to item i. The new vector $\vec{y_c}$ is introduced as the *"item-related-tag"* factor vector and tag set $T^c(i)$ contain tags that are semantically related to those assigned to i.

Tags in the semantically related set $T^c(u)$ and $T^c(i)$ are obtained using both knowledge-based and corpus-based semantic similarity metrics. Specifically, Lin similarity metric was used for the knowledge-base approach, and word2vec was used for the corpus-based metrics. Details of steps for using Lin similarity metric and word2vec to obtain semantically related tags are presented in algorithm 5.1-5.2 respectively.

The *regularized* squared error function of our model hereafter referred to a *SemGTagSVD++* is presented in equation 5.2. The general SGD algorithm described in the methodology section of chapter three was adopted to optimize the model and estimate the weights of parameters $\vec{p}_u$, $\vec{q}_i$, $\vec{x}_s$, $\vec{x}_c$, $\vec{y}_t$ and $\vec{y}_c$.

$$
\begin{aligned}
\varepsilon(p,q,x,y) = \Bigg[r_{ui} - \Bigg(\vec{p}_u + |T(u)|^{-1} \sum_{s \in T(u)} \vec{x}_s + |T^c(u)|^{-1} \sum_{c \in T^c(u)} \vec{x}_c \Bigg) \\
\cdot \Bigg(\vec{q}_i + |T(i)|^{-1} \sum_{t \in T(i)} \vec{y}_t + |T^c(i)|^{-1} \sum_{c \in T^c(i)} \vec{y}_c \Bigg) \Bigg]^2 \\
+ \lambda \Bigg[|\, \vec{p}_u \,|^2 + \sum_{s \in T(u)} |\, \vec{x}_s \,|^2 + \sum_{c \in T^c(u)} |\, \vec{x}_c \,|^2 \\
+ |\, \vec{q}_i \,|^2 + \sum_{t \in T(i)} |\, \vec{y}_t \,|^2 + \sum_{c \in T^c(i)} |\, \vec{y}_c \,|^2 \Bigg]
\end{aligned}
\tag{5.2}
$$

We denote the model as $SemTagSVD^{++}_{wordNet}$ when the semantically related tags in sets $T^c(u)$ and $T^c(i)$ are obtained from knowledge-base metric such as Lin metric. On the other hand, we denote the model as $SemTagSVD^{++}_{word2vec}$ when corpus-based word vector representations are used to derive the tags in the sets $T^c(u)$ and $T^c(i)$.

Algorithm 5.1. Knowledge-based algorithm for most related tags

Input: Unique tags set $T(U)$, set of user assigned tags $T(u)$, set of all related tags $T_a^c(u)$, number of top similar tags N
Output: Set of related user tags $T^c(u)$
procedure Relatedness
  for all $t_i \in T(u)$ **do**
    for all $t_j \in T(U)$ **do**
      $T_a^c(u) \leftarrow \text{LINsim}(t_i, t_j)$
    end for
  end for
sort $T_a^c(u)$ in descending order
select N highest values of $T_a^c(u)$ as $T^c(u)$
end procedure
function LINsim(t_i, t_j)
  $s_i \leftarrow$ synset of t_i from WordNet
  $s_j \leftarrow$ synset of t_j from WordNet
  $lcs(s_i, s_j) \leftarrow$ lowest common subsumer of $s_i$and s_j in WordNet taxonomy
  $IC(lcs(s_i, s_j)) \leftarrow$ Information content of the lowest common subsumer
  $IC(s_i) \leftarrow$ Information content of synset s_i
  $IC(s_j) \leftarrow$ Information content of synset s_j
  $sim \leftarrow 2 * IC(lcs(s_i, s_j))/(IC(s_i) + IC(s_j))$ # Computed according to Lin (1998)
return max(sim)

Algorithm 5.2. Corpus-based algorithm for selecting most related tag from a corpus

Input: Set of unique user tags $T(u)$, number of top similar tags N
Output: Set of related user tags $T^c(u)$
procedure SemanticRel
  for all $t_1 \in T(u)$ **do**
    $T^c(u) \leftarrow \text{Word2Vec}(t_1, N)$
  end for
end procedure

5.4 Experiment

5.4.1 Dataset

The MovieLens and LibraryThing dataset presented in section 4.4.1 of chapter 4 were once again used to represent a target and an auxiliary domain and vice versa. The dataset was also processed in the same manner described in chapter 4 by keeping only ratings on movie items for which at least one tag was assigned. The resulting ratings from this selection totalled 24,564 for movie domain. The set up of Enrich et al. (2013) and Fernández-Tobías and Cantador (2014) was adopted in order to compare their cross-domain recommender models with the one proposed in this chapter. As a result, the first 24,546 ratings in the LibraryThing dataset was selected for book domain ratings.

In addition to movie and book rating data downloaded above, we obtained plot summaries as movie and book corpora to train the corpus-based semantic metric used in computing semantic similarity of social tags. The movie corpus contained summaries extracted from Wikipedia for 42,306 movies. The extracted summaries were also enriched with other metadata such as genre, character names and information about the actors who portray them, including gender and estimated age. On the other hand, the book corpus contained plot summaries for 16,559 books extracted from Wikipedia, enriched with other metadata such as book author, title, and genre.

The plot summaries[1] for the book and movie domains were prepared by Bamman et al. (2013) and Bamman and Smith (2013) who have made them open source for the research community. The details of table 5.1 show the statistics for the movie and book summary corpora.

Table 5.1 Tags from 24,564 Movielens and Librarything ratings selected for semantic metrics evaluation

	Movie Summaries	Book Summaries
Sentences	42,306	16,559
Raw words	9,683,176	5,395,853
Vocabulary	89,818	80 697

[1]Plot summaries for book are available at http://www.cs.cmu.edu/ dbamman/booksummaries.html and for movies at http://www.cs.cmu.edu/ ark/personas/

5.4.2 Evaluated Approaches

The performances of the models proposed in this chapter were compared against conventional single-domain baselines and the state of the art tag based models described in Section 5.2. The recommendation approaches of the all the models are summarised as follows:

- $SVD++$: An adaptation of MF by Koren (2008) that uses what users have rated as implicit data. In the experiments carried out in this chapter, the set N(u) contained all the items rated by user u.
- $gSVD++$: This extension of SVD++ models item metadata in addition to implicit feedbacks from rated items into the factorisation process. The tags assigned to items by any user where considered as the set of item attributes $G(i)$.
- $TagGSVD++$: This model extends gSVD++ by considering tags as user metadata instead of implicit feedbacks inferred from items the users have rated.
- $SemGTagSVD^{++}_{wordNet}$: This is our proposed model when the semantically related tags in sets $T^c(u)$ and $T^c(i)$ are obtained from knowledge-base metric such as Lin metric.
- $SemGTagSVD^{++}_{word2vec}$: This is our proposed model when corpus-based word vector representations are used to derive the tags in the sets $T^c(u)$ and $T^c(i)$.

5.4.3 Methodology

We evaluated the performance of our semantically enhanced rating prediction model in a setting that simulates a partial cold start condition and different levels of data sparsity in a target domain. In each of the experimental case mentioned above and for each evaluated model tested, we observed the results in two scenarios; one using MovieLens as target and LibraryThing as auxiliary domain, and another using LibraryThing as target and MovieLens as the auxiliary domain. In order to ensure a high level of confidence in our observations, we performed ten-fold cross-validation for both experiments.

In a similar process to the experimental set up in chapter 4, rating data from the two domains were concatenated and partitioned into training, testing and validation sets. The domain selected as the target is first split into ten non-overlapping sets, and one portion (i.e. 10% of 24,564) is set aside as the test set. The remaining nine portions (i.e 90% of 24,564) were then partitioned into 80% training set (initially empty) and 20% validation set. The training portion was further subdivided into ten portions to create candidate sets with each portions representing 10% of the total training data. The first 10% of the whole training set (i.e. 0.1 * 0.8 * 0.9 * 24,564 = 1,768 ratings) was selected to simulate a partial cold-start situation where only small information about users in the target domain is available. Subsequent additions in 10% batches were used to represent different sparsity levels by removing from the candidate set and adding to the train set; until the whole set of training data is at 100%. To enable cross-domain knowledge transfer, each of the 10% batches of training set in the target domain was added to the whole set of 24,564 ratings from the auxiliary domain to make up the complete training set for the experiment. The dataset divisions and subdivisions as described above for training, testing and validating are illustrated for the cross-validation iterations and the cold start/sparsity simulations in figure 5.1. The procedure was repeated ten times to cross-validate our models for each of the original ten divisions (test set) of the target domain dataset.

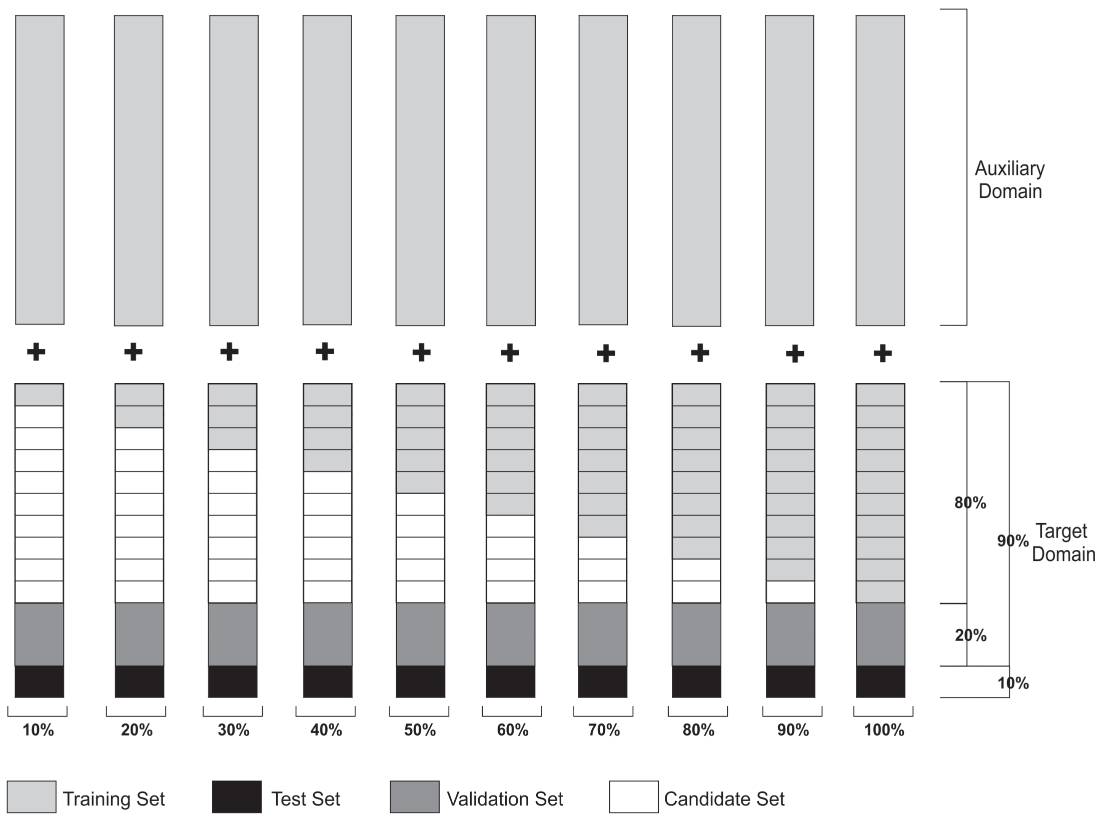


Fig. 5.1 Data partitioning for 10 fold cross-validation with different sparsity levels.

5.5 Results

In partitioning the data for testing the models' performance, 20% of the training data from the target domain was set aside for validating the models and for determining the best parameters for each model. This allows for a fair comparison of the models without underestimating or overestimating the results. The best hyperparameter values for each of the evaluated model is presented in table 5.2. The values obtained for number of latent features k, learning rate α and regularization γ in table 5.2 were obtained using a grid search on the validation set.

Table 5.2 Best hyperparameter values for the evaluated models while setting MovieLens (ML) as target domain and LibraryThing(LT) as auxiliary and vice versa.

	$ML \rightarrow LT$			$LT \rightarrow ML$		
	k	α	γ	k	α	γ
$SVD++$	27	0.020	0.006	27	0.020	0.006
$GSVD++$	28	0.018	0.005	27	0.021	0.005
$TagGSVD++$	29	0.018	0.034	30	0.020	0.032
$SemTagGSVD^{++}_{WordNet}$	28	0.020	0.055	29	0.021	0.055
$SemTagGSVD^{++}_{word2vec}$	29	0.017	0.057	31	0.021	0.059

In table 5.2, the difference in the optimal number of factors and learning rates between the evaluated approaches is minimal. As can be seen in table 5.2, the amount of regularization required for the $TagGSVD++$ and two of the proposed models $SemTagGSVD^{++}_{WordNet}$ and $SemTagGSVD^{++}_{word2vec}$ model is comparatively large. Specifically, regularisation of $TagGSVD++$ is approximately five times that of SVD++, while $SemTagGSVD^{++}_{word2vec}$ requires approximately nine times regularization of $SVD++$. This may as a result of the additional set of latent variables for unique tags and semantically related tags that is modelled into the approach. This phenomenon was also observed by Fernández-Tobías and Cantador (2014) who argued that more complex models are able to account for greater variance in the data and tend to overfit more easily, therefore requiring more regularisation.

The performance of the models in terms of Mean Average Error (MAE) is presented in figure 5.2 when LibraryThing is used as Auxiliary and MovieLens as the target domain. It is clear in figure 5.2 that both semantically enhanced models $\left(SemTagGSVD^{++}_{word2vec}\right.$ and $\left.SemTagGSVD^{++}_{word2vec}\right)$ consistently outperform the other approaches for all sparsity levels in the target domain. In addition, the proposed models also outperform the rest at moderate cold start conditions when about 10%–20% of the ratings are available.

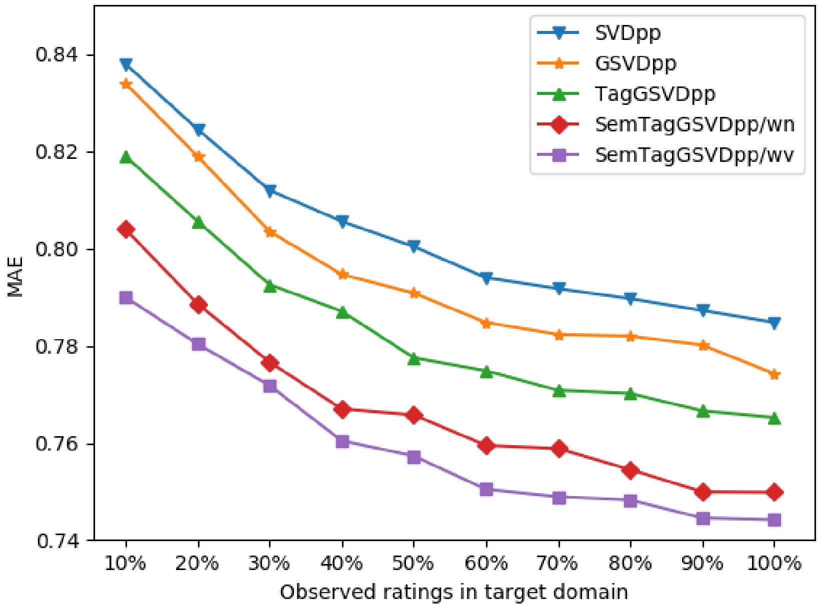


Fig. 5.2 Average MAE over the 10 folds with LibraryThing as auxiliary domain and MovieLens as target domain.

The performance of the models when MovieLens is set as the auxiliary domain and LibraryThing as the target domain is presented in figure 5.3. Once again, $SemTagGSVD^{++}_{WordNet}$ and $SemTagGSVD^{++}_{WordNet}$ achieved the best performance at each of the rating sparsity level considered. When graphs of models from table 5.3 are compared with the ones in figure 5.2, it can be observed the values of MAE are relatively larger when the movie domain is used as the auxiliary domain. This may be an indication that transfer of knowledge is not as effective when LibraryThing is the auxiliary domain and MovieLens is the target domain. Enrich et al. (2013) and Fernández-Tobías and Cantador (2014) also reported similar observation and Enrich et al. (2013) argued that this may be caused by differences in the ratio of intercepting tags between the domains. There are only 13.81% of the tags in MovieLens that are shared in LibraryThing (see Table 4.1), and as a result, less latent tag factors learned in the auxiliary domain can be used in the target to compute rating predictions.

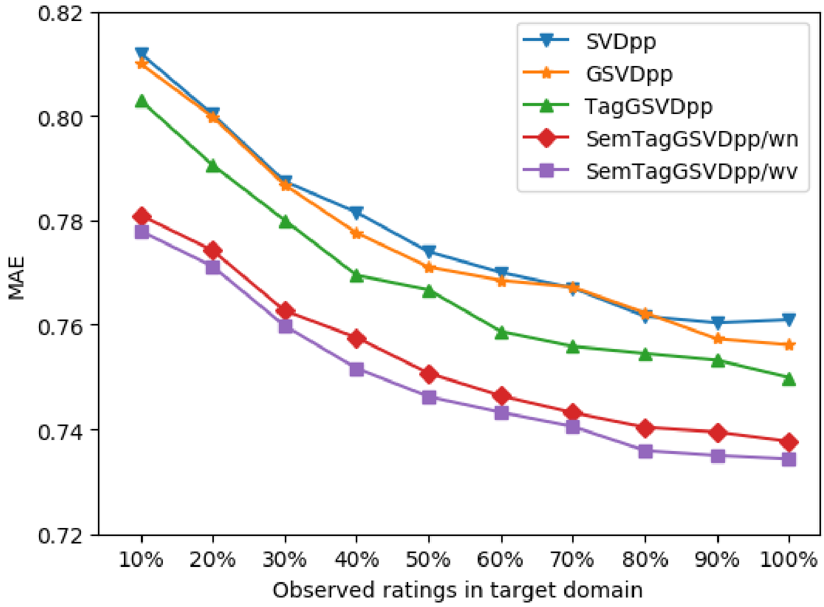


Fig. 5.3 Average MAE over the 10 folds with MovieLens as auxiliary domain and LibraryThing as target domain.

5.6 Conclusion

A major challenge in cross-domain recommender systems is the adaptation or linking of different domains to support knowledge transfer. Recent research efforts by Shi et al. (2011) and Enrich et al. (2013) and Fernández-Tobías and Cantador (2014) have however exploited social tags to adapt domains for cross-domain recommendation.

In this chapter, an extension of the popular SVD++ model Koren (2008) was proposed to incorporate semantically related tags into an MF based rating prediction model. A new set of latent variables was introducing to represent semantically related tags in the user and item profiles. The new parameters enabled modelling of the influence of semantically related tags for the effective transfer of knowledge between the domains. On the basis of the results of our experiments on the movies and books recommendation domains, we conclude that exploiting additional tag factors to represent semantically related tags in the factorisation process improve the transfer of knowledge, and the accuracy of recommendations.

Finally, we note that one of two $semantically-aware$ model we proposed outperformed the other consistently throughout our experiments. Specifically, the model that utilises corpus-based metric (word2vec) which depends on the co-occurrence of words in a corpus gave an overall better performance. Therefore, we conclude that the structure of words sequence is important when incorporating semantically related tags into cross-domain matrix factorisation model.

Chapter 6

Exploiting Category Similarity Attributes In Multi-Category POI Recommendation

Several studies on the business value of recommender systems such as Gomez-Uribe and Hunt (2015) report for Netflix show how recommender systems can be effectively used as a tool to tackle the *"information overload"* problem. Recommendation systems have been broadly used to provide items of interest to the users (e.g., movies, music, books, jokes and news). On the other hand, many of the existing recommendation techniques still rely primarily on the rating value that users give to items. RS based on explicit ratings only do not take context such as location, time or environment into account and may not be best suited in making recommendations about items that depend on these contexts. According to Adomavicius and Tuzhilin (2008), the importance of context to recommendation has been found to be consistent with behavioural research on consumer decision making in marketing. Consumers generally make their purchasing decisions based on context. Therefore, accurate personalisation of products for consumers depend on the extent to which the recommender system has included relevant contextual information into a recommendation technique. As an example, if we consider location as a source of contextual information, users have generally been known to show more interest for items that are nearby (e.g., restaurants, museums, cinemas). Also, there has been an increase in the amount of data that can be labelled with location and tagging information. The popularity of social networking platforms and mobile computing technologies has contributed to a surplus of location data and driven the need for recommender techniques that can process location information for benefit of users.

In order to address the cold start and sparsity problems common to recommender systems, research efforts have been invested in a different type of recommender systems known as Context-Aware Recommendation Systems (CARS) (Levandoski et al. (2012)). The ingenuity of these systems is the inclusion of the context of the user and/or the context of items in the computing rating predictions or item recommendations. Among several types of data that can be considered to represent the context in a recommendation process, Horozov et al. (2006) reported that the location of users and items has more significance in the personalisation process.

6.1 Introduction

Location Aware Recommendation (LAR) has emerged as a subcategory of CARS and led to alternative approaches of achieving more accurate personalisation. According to Sarwat et al. (2014), LAR systems exploit the spatial aspect of ratings when producing recommendations. In general, they can be considered as an adaptation to traditional recommendation systems with the inclusion of location.

The MF based cross-domain recommendation model that we presented in Chapter 4 and 5 belong to a category of MF-based collaborative filtering model referred to as attribute-aware matrix factorization models [Koren et al. (2009), Gantner et al. (2010), Gantner et al. (2011), Manzato (2013)]. Such models typically represent additional information about users (user attributes, e.g. gender, age, geographical location, occupation) and items (item attributes, e.g. genres, product categories, keywords) as features in a latent space.

Location can be considered as a part of a context. According to Gartner et al. (2007), it determines what information and services the user may expect. Location in Geographic Information Systems (GIS) is considered to have two components: spatial information (coordinate and projection information for spatial features) and attribute data. Attribute data is information appended in tabular format to spatial features. The spatial data is the where and attribute data can contain information about the what, why and how. Attribute data provides the qualitative and/or quantitative characteristics about spatial data. Whereas the spatial information of a location is generally fixed, some attributes are unique to a location while others may be shared with other locations. As an example, two shopping centres at different locations on a high street may both be perceived by users to have a classy indoor attribute, but one may have an extra attribute by the outdoor seating they offer.

6.2 Related Work

Item recommendation is the task of predicting a personalised ranking on a set of items (e.g. websites, movies, products), Rendle et al. (2009). Several research efforts have improved the precision of the items recommended by including implicit and explicit feedback from users into the ranking model. One of the most popular models for recommender systems is k-nearest neighbour (kNN) collaborative filtering (Deshpande and Karypis (2004)). In order to find the nearest neighbours to a user, a similarity matrix of users is computed from the users' preference history using standard similarity metrics e.g. the Pearson correlation. A few other works such as Koren (2008) have treated the similarity matrix as MF model parameters learned specifically for the item recommendation task. Singular value decomposition methods proposed by Koren et al. (2009) have also been used to learn the feature matrices of the MF model's parameters. Matrix factorization models learnt by SVD have been shown to suffer from the problem of "overfitting" [Rendle et al. (2009); Koren et al. (2009)] when applied to the item recommendation task.

In other research efforts such as Schmidt-Thieme (2005), the item recommendation problem has been treated as a classification problem, and solutions using a set of binary classifiers were proposed. A more recent and popular optimised ranking model referred to as Bayesian Personalized Ranking (BPR) was proposed by Rendle et al. (2009). BPR uses an optimisation criterion based on pairs of items to compute ranking scores for items recommended to users. Before the work of Rendle et al. (2009), several attempts at solving the item recommendation task generally optimised their model to predict if a user selects an item or not rather than directly optimising the parameters for ranking. According to Li et al. (2015), BPR learns the ranking models parameter based on pairwise comparison of items by optimising Area Under the ROC Curves (AUC).

In POI recommender systems, a POI is considered as an item and a POI recommender model suggests a ranked number of POIs (i.e. the top-N POIs) to users. Several researchers [Ye et al. (2011); Gao et al. (2013a); Liu et al. (2013a) and Li et al. (2015)] have proposed many models that use different contextual factors such as geographical and temporal influence, to improve on performance of POI recommenders. In order to adopt optimised ranking methods to POI recommendation tasks, Li et al. (2015) used an optimisation criterion based on Ordered Weighted Pairwise Classification (OWPC). OWPC was proposed by Usunier et al. (2009) and has been successfully applied in text retrieval and image annotation. According to Li et al. (2015), OWPC though designed to use binary values can be extended to use multi-class values. As a result, OWPC can be adapted to address the POI recommendation task where implicit feedbacks are inferred from the different visiting frequencies of users.

In Li et al. (2015), the authors proposed a geographical factorisation method (RankGeoFM), which uses OWPC as a criterion to derive an objective loss which is subsequently optimised for POI recommendation. First, a user's preference rankings for POIs is inferred from his/her frequency of visits to each POI. The assumption is that the higher a user's visit frequency is to a POI (i.e. the number of check-in), the more preference that user has for the POI. All unvisited POIs by the user are assumed to be less preferred. The authors followed the assumption that a POI's visit frequencies is affected by its geographical neighbours as generalised by Tobler's First Law of Geography -*"Everything is related to everything else, but near things are more related than distant things"*- Tobler (1970).

Furthermore, the authors of RankGeoFM applied a weighting parameter to their method in order to model the contributions of different neighbours to the geographical influence. For clarity, if we consider a user u and POI ℓ, the recommendation score $y_{u\ell}$ as predicted by RankGeoFM is as given below:

$$y_{u\ell} = U_u^{(1)} \cdot L_\ell^{(1)} + U_u^{(2)} \cdot \sum_{\ell^* \in N_k(\ell)} w_{\ell\ell^*} L_{\ell^*}^{(1)} \tag{6.1}$$

On the right side of the equation, the first term $\left(U_u^{(1)} \cdot L_\ell^{(1)}\right)$ models the preferences of user u as typical in traditional matrix factorization. The second term $\left(U_u^{(2)} \cdot \sum_{\ell^* \in N_k(\ell)} w_{\ell\ell^*} L_{\ell^*}^{(1)}\right)$ models geographical influence. The set of parameters $\theta = \{U^{(1)}, L^{(1)}, U^{(2)}\}$ for the geographical factorization model are learned from training data. The authors projected members of set θ as latent factors in a K-dimensional space with matrices $U^{(1)} \in \mathbb{R}^{|U| \times K}$, $U^{(2)} \in \mathbb{R}^{|U| \times K}$ and $L^{(1)} \in \mathbb{R}^{|L| \times K}$. A weighted parameter "$w_{\ell\ell^*}$" is applied to the second term to model geographic influence into the equation. The term $N_k(\ell)$ denotes the k-nearest neighbors of POI ℓ.

6.3 Categorical Correlation

In LBSNs, each POI is grouped under one or more categories. POI categories can be considered as thematic groups that indicate what activities take place in a POI or what kind of service to be expected at the POI in the category. For instance, a person visiting a Sport/Leisure (SL) Centre may participate in a leisure activity. An SL Centre focused on fitness indicate that activities at the POI will be about fitness. Users of LBSN show preferences for different categories of POI. According to Zhang and Chow (2015), people show distinct biases for different categories of POIs, e.g., a food enthusiast likes visiting restaurants to taste the various food and a touristy person will prefer travelling all over the world to view tourism attractions.

In reality, categories may have the same type of locational attributes. For example, Art/Entertainment and Sport/Leisure categories typically provide parking spaces for their customers. Restaurant and Shopping categories may provide wheelchair access for vulnerable users. As a result, the relevance score of an unvisited POI to a user can be computed by exploiting the correlations of location attributes between the categories of the user's visited POIs and the unvisited POI. In order to address the sparsity problem in POI recommenders, we propose a multi-category recommender model which considers the similarity of POI categories in computing a relevance score for unvisited POIs.

We adopted Jensen Shannon Divergence (JSD) as proposed by Remus (2012) to compute POI category similarities. In Remus (2012), the authors used JSD to measure the similarity between an auxiliary domain and a target domain. The similarity value indicated how adaptable a model trained in the auxiliary domain is for testing in the target domain. The domain adaptation approach aligns with our multi-category approach to POI recommendation because they both attempt to overcome domain dependency. JSD similarity metric is based on Kullback-Leibler Divergence and is derived below by following Remus (2012) and Ponomareva and Thelwall (2012) approach to computing cross-domain similarity.
Let $D_{KL}(C_1 \| C_2)$, $D_{JD}(C_1 \| C_2)$ be the Kullback-Leibler and Jensen Shannon Divergence respectively; and C_1, C_2 be the probability distribution over a finite set W of unigrams from category 1 and category 2. The unigrams are obtained from plain text corpora extracted from user reviews on POIs in category 1 and category 2. Let M be the average distribution of C_1 and C_2 i.e $M = \frac{1}{2}(C_1 + C_2)$.

$$D_{KL}(C_1, C_2) = \sum_{w \in W} C_1(w) \log \frac{C_1(w)}{C_2(w)}$$

$$D_{JD}(C_1, C_2) = \frac{1}{2} D_{KL}(C_1, M) + \frac{1}{2} D_{KL}(C_2, M) \tag{6.2}$$

JSD values are bounded between 0 and 1 i.e. $0 \geq D_{JD}(C_1, C_2) \leq 1$. Jesen Shannon Divergence becomes a standard distant metric when its square root is computed [Melville et al. (2005), Aslam and Pavlu (2007) and Briët and Harremoës (2009)].

6.4 Proposed Model

The motivation for our "category-aware" geographical MF model hinges on the assumption that, location attributes can be exploited to *bridge* user preferences across the POI categories/domains in a similar fashion as the social tag-based models presented in Chapter 4. The recommendation approach we adopted in this chapter is based on a recent POI recommender model known as "RankGeoFM" and proposed by Li et al. (2015). RankGeoFM is a personalised ranking based matrix factorisation Model. We selected this model because it can be generalised and extended to include other types of contextual information. Also, it has been reported to perform significantly better than other models for POI recommendation. Li et al. (2015) reported a 30% improvement in POI recommendation precision using real-life location-based datasets. A more comprehensive evaluation of 12 different POI recommenders by Liu et al. (2017) also found RankGeoFM gave the best performance at low data sparsity. They reported that Rank-GeoFM outperformed the second best POI recommender model by 5%-10%.

There has been extensive research carried out on POI recommendation; however, no single study exists to the best of our knowledge which considered cross-domain approaches for POI recommendation. In this section, we present a model which considers the domain/category of POIs as contextual information for ranking places of interest to users. The following notations and definitions cover all the key parameters used in formalising our model.

- **User Set $\mathcal{U}$:** this set represent all the users in the LBSN who go to places they are interested in visiting. $\mathcal{U} = \{u_1, u_2, u_3, ..., u_{|\mathcal{U}|}\}$.
- **POI Set $\mathcal{L}$:** this is the set of places in the LBSN that users can visit as often as they prefer. $\mathcal{L} = \{\ell_1, \ell_2, \ell_3, ..., \ell_{|\mathcal{L}|}\}$.
- **Visited POI Set $\mathcal{L}^u$:** this is the set of places that user u has visited i.e., the histor y of his/her past visits are contained in the set $\mathcal{L}^u$.
- **Category Set $\mathcal{C}$:** this set holds the different categories that can be assigned to places in the LBNS. $\mathcal{C} = \{c_1, c_2, c_3, ..., c_{|\mathcal{C}|}\}$.
- **Check-in Tensor $X_{u\ell c}$:** a $|\mathcal{U}| \times |\mathcal{L}| \times |\mathcal{C}|$ tensor whose entries are the frequency of visits of users to POIs belonging to a category.
- **User-POI-Category Triple $\mathcal{D}$:** a tuple of users, POIs and categories where $(u, \ell, c) \mid x_{u\ell c} > 0$.

- **Nearest Neighbour Set** $N_k(\ell)$**:** this is the set of POIs that are nearest to POI "ℓ" by distance.
- **Recommendation Score** y_{ul}**:** the predicted score of POI "l" for user "u".
- **Recommendation Score** $y_{u\ell c}$**:** the predicted score of POI "ℓ" in category "c" for user "u".

The recommendation problem for our proposed ranking based model for multi-category POI recommendation is defined as follows:

Given a user "u" and a set of POIs "$\mathcal{L}^u$" already visted by "u". Let "c" be a category from the set of all POI categories "C" such that $c \in C$. Recommend a set of POI "$\mathcal{L}$" to user "u" that are not in "$\mathcal{L}^u$" .

In order to clearly identify equations that are part of the model proposed in this chapter, we have enclosed them with rectangular borders. The rest of the equations that are adopted or applied from literature are without borders and referenced in corresponding sections.

6.4.1 Model Formulation

In the following section, we present details of our personalised ranking model which uses similarities of POI categories as context for recommending POI to users. We hereafter refer to our model as RankGeoCatFM and show how we derived and optimised the loss function in the following subsections.

First, we introduce three latent factor matrices in addition to matrices in RankGeo-FM to complete its adaptation to RankGeo-CatFM. Let the first newly introduced matrix $U^{(3)} \in \mathbb{R}^{|U|\times|K|}$ model users' bias for categories. Let the second matrix $C^{(1)} \in \mathbb{R}^{|C|\times|K|}$ represent the latent factors for POI categories. Let a third matrix $U^{(4)} \in \mathbb{R}^{|U|\times|K|}$ for users model their bias for similar categories. Let matrix $W \in \mathbb{R}^{|C|\times|C|}$ be a similarity matrix to model the correlation of categories based on their similarity. Let $x_{u\ell c}$ denote the frequency that a user u checked in at POI ℓ in category c, then w_{cc^*} is derived from equation 6.3.

$$w_{cc^*} = \frac{\sum_{u\in U}\sum_{l\in L} x_{ulc}x_{ulc^*}}{\sqrt{\sum_{u\in U}\sum_{l\in L} x_{ulc}^2}\sqrt{\sum_{u\in U}\sum_{l\in L} x_{ulc^*}^2}} \tag{6.3}$$

However, we propose that each element w_{cc^*} of matrix W can be the metric scores derived from Jensen Shannon Divergence between the categories. Let $D_{JD}(C,C^*)$ be the Jensen Shannon Divergence between visited category C and yet to be visited category C^*, then w_{cc^*} can be computed in equation 6.4 below.

$$w_{cc^*} = \sqrt{D_{JD}(C,C^*)} \tag{6.4}$$

A function for predicting the recommendations score of a POI "ℓ" for user "u" is presented in equation 6.5. The function is rewritten in equation 6.6 to clearly show how it extends Rank-GeoFM with the inclusion of parameters for POI categories.

$$\boxed{y_{ulc} = U_u^{(1)} \cdot L_\ell^{(1)} + U_u^{(2)} \cdot \sum_{\ell^* \in N_k(\ell)} w_{\ell\ell^*} L_{\ell^*}^{(1)} + U_u^{(3)} \cdot C_c^{(1)} + U_u^{(4)} \cdot \sum_{c^* \in C} w_{cc^*} C_{c^*}^{(1)}} \tag{6.5}$$

$$\boxed{y_{ulc} = y_{ul} + U_u^{(3)} \cdot C_c^{(1)} + U_u^{(4)} \cdot \sum_{c^* \in C} w_{cc^*} C_{c^*}^{(1)}} \tag{6.6}$$

In order to learn the parameters in recommendation score function of equation 6.5, we followed Li et al. (2015) who used OWPC criterion to learn the parameters of their geographic MF model.

First, let the visit frequency of user u to POI ℓ in category c be denoted as $v_{u\ell c}$. Given the assumption that preference of a user u is inferable from his/her visit frequency to POI ℓ, it follows that the rank of POI ℓ in a category c for user u should be higher than ℓ' if $v_{u\ell c} > v_{u\ell' c}$. According to Li et al. (2015), the order for ranking a set of POIs to a user can be computed by minimising a ranking incompatibility function. The incompatibility for our proposed model -RankGeo-CatFM- is as given below:

$$Incomp(y_{u\ell c}, \varepsilon) = \sum_{\ell \in \mathcal{L}, u \in U, c \in C} I(v_{u\ell c} > v_{u\ell' c}) I(y_{u\ell c} > y_{u\ell' c} + \varepsilon) \tag{6.7}$$

In equation 6.7, "$I(.)$" is the indicator function, such that $I(A)$ = 1 when A is true and 0 otherwise, ε is the error tolerance hyperparameter. The recommendation score $y_{u\ell c}$ is calculated by our proposed factorization model -RankGeo-CatFM.

After computing the incompatibility for all the visit frequency in the triple D, we obtain the following loss function.

$$\mathcal{O} = \sum_{(u,\ell,c) \in D} E(Incomp(y_{u\ell c}, \varepsilon)) \tag{6.8}$$

The function $E(.)$ in equation 6.8 transforms the ranking incompatibility into a loss using the OWPC as proposed by Usunier et al. (2009) and presented below:

$$E(r) = \sum_{i=1}^{r} \frac{1}{i} \tag{6.9}$$

In equation 6.9, $E(r)$ calculates the sum over losses at each rank position from 1 to rank r. Each of the ranking position is assigned with a loss $\frac{1}{i}$. As a simple illustration, consider case where four POIs have been incorrectly ranked higher than POI ℓ i.e. $Incomp(y_{u\ell c}, \varepsilon) = 4$. The loss for this tuple (u, ℓ, c) will be $E(4) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4}$.

6.4.2 Loss Optimization

In this subsection, we present the method for learning parameters of RankGeo-CatFM by minimising loss function $\mathcal{O}$ of equation 6.8. We followed Li et al. (2015) who adopted Stochastic Gradient Descent (SGD) for optimising their geographic factorisation model (RankGeo-FM). SGD was selected for optimizing our model because it has been shown [Koren (2009), Enrich et al. (2013), Manzato (2013), Fernández-Tobías and Cantador (2014)] to be fast and can produce parameter values with good fit from a training data. In addition, we considered SGD as applicable for minimising RankGeo-CatFM since it is a direct extension RankGeo-FM.

In order to use SGD for learning the parameters of our model, the indicator function $I(v_{u\ell c} > v_{u\ell' c})I(y_{u\ell c} > y_{u\ell' c} + \varepsilon)$ has to be continuous and differentiable. We follow Li et al. (2015) who used a sigmoid function to approximate their indicator function.

First, we rewrite the loss $E(Incomp(y_{u\ell c}, \varepsilon)$ in the same way as Li et al. (2015):

$$E(Incomp(y_{u\ell c}, \varepsilon)) \cdot 1 = E(Incomp(y_{u\ell c}, \varepsilon)) \frac{\sum_{\ell' \in \mathcal{L}} I(v_{u\ell c} > v_{u\ell' c}) I(y_{u\ell c} > y_{u\ell' c} + \varepsilon)}{Incomp(y_{u\ell c}, \varepsilon)} \tag{6.10}$$

$$\approx E(Incomp(y_{u\ell c}, \varepsilon)) \frac{\sum_{\ell' \in \mathcal{L}(u,\ell',c)} s(y_{u\ell' c} + \varepsilon - y_{u\ell c})}{Incomp(y_{u\ell c}, \varepsilon)} \tag{6.11}$$

The function $\mathcal{L}(u, \ell, c) := \{\ell' \mid I(v_{u\ell c} > v_{u\ell' c}) I(y_{u\ell c} > y_{u\ell' c} + \varepsilon) = 1\}$, and the sigmoid function $s(A) := \frac{1}{1+exp(-A)}$ is used to approximate the indicator function. We can compute

SGD for our model's parameter set θ from the derivative below:

$$\frac{\partial E(Incomp(y_{u\ell c},\varepsilon))}{\partial \theta} \approx E(Incomp(y_{u\ell c},\varepsilon))\frac{\sum_{\ell' \in \mathcal{L}(u,\ell',c)} \frac{\partial s(y_{u\ell' c}+\varepsilon - y_{u\ell c})}{\partial \theta}}{Incomp(y_{u\ell c},\varepsilon)} \tag{6.12}$$

$$= \frac{E(Incomp(y_{u\ell c},\varepsilon))}{Incomp(y_{u\ell c},\varepsilon)} \sum_{\ell' \in \mathcal{L}(u,\ell',c)} \delta_s \times \frac{\partial(y_{u\ell' c}+\varepsilon - y_{u\ell c})}{\partial \theta} \tag{6.13}$$

The term $\delta_s = s(y_{u\ell' c}+\varepsilon - y_{u\ell c})(1-s(y_{u\ell' c}+\varepsilon - y_{u\ell c}))$ in equation 6.10.

6.4.3 Enhanced Parameter Learning

Although stochastic gradient can be calculated by equation 6.13, it is however resource intensive and time consuming. This is because terms in the summation and the $Incomp(y_{u\ell c},\varepsilon)$ require the computation of recommendation score $y_{u\ell c}$ for all POIs. We adopt the faster technique proposed by Li et al. (2015) which removes the summation and estimate $Incomp(y_{u\ell c},\varepsilon)$ by sampling.

If we turn our attention back to equation 6.10, we see that the loss function is the sum of all the losses computed for a set of incorrectly-ranked POIs. Each POI that is incorrectly-ranked such as ℓ' has a loss given by equation 6.14. The stochastic gradient can be approximately computed by sampling POI ℓ'.

$$\bar{E} = E(Incomp(y_{u\ell c},\varepsilon))s(y_{u\ell' c}+\varepsilon - y_{u\ell c}) \tag{6.14}$$

In addition, each POI has the constant probability $\frac{1}{Incomp(y_{u\ell c},\varepsilon)}$ of been selected for sampling. As a result, equation 6.12 is reduced to the gradient of equation 6.15:

$$\frac{\partial \bar{E}}{\partial \theta} = E(Incomp(y_{u\ell c},\varepsilon))\delta_s \times \frac{\partial(y_{u\ell' c}+\varepsilon - y_{u\ell c})}{\partial \theta} \tag{6.15}$$

The SGD algorithm finds a local minimum of the loss function by iteratively updating the parameters after each observed tuple (u,ℓ,c) in visit-frequency data D. In general, SGD works by shifting θ in the direction of maximum descent of the loss given by its gradient:

$$\theta \leftarrow \theta - \gamma \frac{\partial \bar{E}}{\partial \theta} \tag{6.16}$$

6.5 Experiment

The multi-category POI recommender proposed in this chapter are modelled to provide recommendations even when there are neither users nor POIs common among the categories (i.e. disjointed). We conjecture that category correlation can be exploited to find user preferences across categories in a similar manner as the social tags in models presented in Chapter 4 and 5. Therefore, we distinguish between two different types of multi-category scenarios for POI recommendation:

1. Scenario I: Multi-category without intersecting POIs and Users

   In this scenario, the user-poi matrices from the different POI categories are concatenated. All *inter-category* duplicates of users and POIs are filtered out. The scenario simulates a situation where there are no intersect of users and POI between the target and auxiliary domain.

2. Scenario II: Multi-category with intersecting POIs and Users

   In this scenario, the user-poi check-in matrices for the different categories are also concatenated. However, all *inter-category* duplicates of users and POIs are not removed. The newly combined matrix is randomised and partitioned into train, validation and test sets.

The dataset used for the experiments is presented next followed by a description of the different POI recommender models that were compared with the proposed in this chapter. The methodology used to empirically validate the models' performance and the results are discussed in the later sections.

6.5.1 Dataset

The Yelp Open Dataset[1] is a subset of Yelp's businesses, business reviews, user and location-based data for personal, educational, and academic purposes. The dataset provides information about local businesses in 12 metropolitan areas across 4 countries. Each business in Yelp is categorised into a set of nearly 1000 types according to the nature of the business. Categories available in the Yelp dataset are grouped into 22 parent categories. These are broad-level groups, e.g. Restaurants, Hotels and Travel, Event planning and Services. There are also child categories which are used to specify fine-grained properties of the local business.

[1]The Yelp dataset was downloaded from https://www.yelp.co.uk/dataset/challenge

For example, a business with parent category of restaurant can have pizza, vegetarian or British as the child categories. We emphasise here that the parent categories are the groups that we used as domains in the experiments of this chapter.

Apart from categorisation, the dataset also contains ratings and reviews where specific preference of users for businesses are recorded as numeric values and textual comments. The dataset also has attributes that explicitly defines the properties of each local business. Some example of attributes in yelp dataset generally includes; type of parking, delivers (or not), noise level or business ambience. Some business attributes are in more that one parent category (e.g. type of parking), while others (e.g. deliveries) are unique to a parent category.

A subset of the 22 parent categories was selected for our experiments due to the size and complexity of the dataset. Specifically, we selected the Restaurants category, Food category and Nightlife category to represent 3 different domains for our experiments. Criteria for selecting the subsets were as follows:

- POI and User Coverage: first we filtered out users profiles where the user had visited less than 20 POIs. We also constrained the POIs to the ones that were visited by more than 20 users in all categories.
- Maximum Number of Check-in: the dataset was processed such that there were no intersect of users or POIs among all the categories. The top three categories with the highest number of POI check-ins were then selected. Restaurant category had 106,704 check-ins, followed by food category with 28,640 check-ins. The category with third highest check-ins was Nightlife category having 10,511 check-ins.
- Total Number of Users: Finally, the number of check-ins for the three categories were constrained by the number of users in the category with the fewest users. The Nightlife category had the fewest number of users at 874 and 3099 users for scenario I and II respectively. The check-ins for the Restaurant and Food categories were therefore bounded by the same number of users. This selection helped avoid bias due to differing number of users in the categories during the training phase of experiments.

The statistics of the dataset selected for scenarios I (without intersecting user/POIs across categories) and scenario II (With intersecting users/POIs) are detailed in table 6.1 and table 6.2 respectively. The venn[2] diagrams in figure 6.1(a) and (b) show the size of User and POI intersections among the 3 sets of categories in the dataset. Except for the intersection of

[2]The curves of the Venn diagram only show relationships and not the exact proportion of the users and POIs in the category sets.

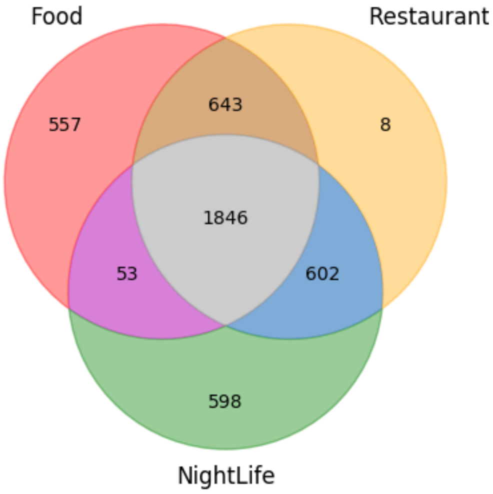
Food and NightLife categories, the larger size of user intersections and POI intersections compared to the size of users and POIs that do not intersect with other categories stands out in both figures 6.1 (a) and (b).

Table 6.1 Statistics for 874 users in Restaurant, Food and Nightlife categories used in performance evaluation of models without intersecting users and POIs.

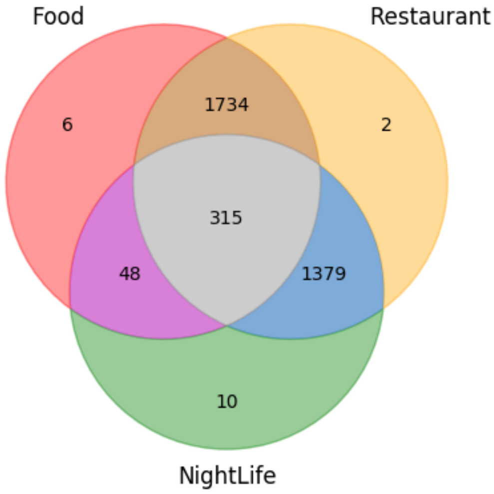
	Restaurant (RT)	Food (FD)	Nightlife (NL)
Number of Check-ins	16688	4216	2661
Number of POIs	3404	897	269
Number of Attributes	135	53	99
Number of Unigrams	28414	17025	10025 31550

Table 6.2 Statistics for 3,099 users in Restaurant, Food and Nightlife categories with intersecting users and POIs.

	Restaurant (RT)	Food (FD)	Nightlife (NL)
Number of Check-ins	74041	45578	41066
Number of POIs	3430	2103	1752
Number of Attributes	135	53	99
Number of Unigrams	124537	78521	70550



(a) Intersecting and Unique user sizes.



(b) Intersecting and Unique POI sizes.

Fig. 6.1 Venn diagrams showing sizes of intersecting and non-intersecting Users and POIs in category for Food, Restaurant and NightLife.

6.5.2 Evaluated Approaches

We compared the performance of our "category-aware" model against the following recommendation approaches:

- **Most popular:** The popularity of a POI is measured as the number of visitations it receives from users in the training sample. This approach is not personalised, does not consider geographic influence and does not use any other contextual information for recommending POIs.
- **Bayesian Personalized Ranking MF:** we considered BRP as an important baseline because it uses an alternative ranking criterion to our proposed models. The Bayesian Personalized Ranking criterion is a pairwise optimisation ranking method proposed by Rendle et al. (2009) and has become one of the most important ranking criteria in item recommendation. BPR MF does not consider geographical influences in it's ranking optimisation.
- **RankGeo MF:** is a ranking-based MF model that learns users' preference rankings for POIs with consideration of the geographical influence of neighbouring POIs.

According to Li et al. (2015), RankGeo MF uses an optimisation criterion known as Ordered Weighted Pairwise Classification (OWPC) to derive an objective loss from a ranking incompatibility function.

- **RankGeo-Cos MF:** this is the first of our proposed ranking-based MF models. The model considers the categories of the POIs in computing the recommendation score used for ranking the POIs. The model considers both geographic influence and POI category similarity as contextual information for improving POI recommendation. The model uses the cosine similarity metric to estimate the similarity between two categories.

- **RankGeo-Cat MF:** this is our second model, and similarly to the previous one, it considers the categories of the POIs in computing the recommendation score used for ranking the POIs. The model also considers both geographic influence and POI categories as contextual information for improving POI recommendation. However, it uses a JSD distance metric to pre-compute category similarity.

6.5.3 Methodology and Metrics

In order to evaluate the performance of POI recommender models in the preceding subsection, we follow the general set up for k-fold cross-validation approach described in our methodology chapter. Our approach aligns with the methodology of Kluver and Konstan (2014) who evaluated performances of recommender models with a user-based cross-validation protocol (i.e. users are split in K sets). Specifically, we first divide the set of users in our dataset into five disjoint sets of approximately equal size. At each stage of cross-validation, all the data from four of the sets are used for training the model. The POIs visited by users in the fifth set (i.e. test) are then randomly split into the three subgroups described below:

- A training subgroup, initially empty and incrementally filled with the POI visited by users to produce different cold start profile sizes.

- A candidate subgroup, is the set of POIs visited by users for incrementing the training subgroup and also used for tuning hyperparameters of the models,

- A testing subgroup used to compute the performance metrics. The POIs selected for the testing subgroup were the most recently visited POIs of the user.

The purpose of the experiments in this chapter was to understand how the different recommender approaches perform as the number of POI visited in the training set increases i.e. at different cold-start levels.

Finally, we followed the evaluation procedure originally proposed in Kluver and Konstan (2014) illustrated in figure 6.2 and described below:

1. We train the recommendation algorithm on the training group, and then we compute the evaluation metrics on the test group.
2. For every user in the test subgroup, a candidate POI is added to the training subgroup and removed from the candidate subgroup.
3. The performance metrics are measured again on the test group, after re-training the models on the extended training group.

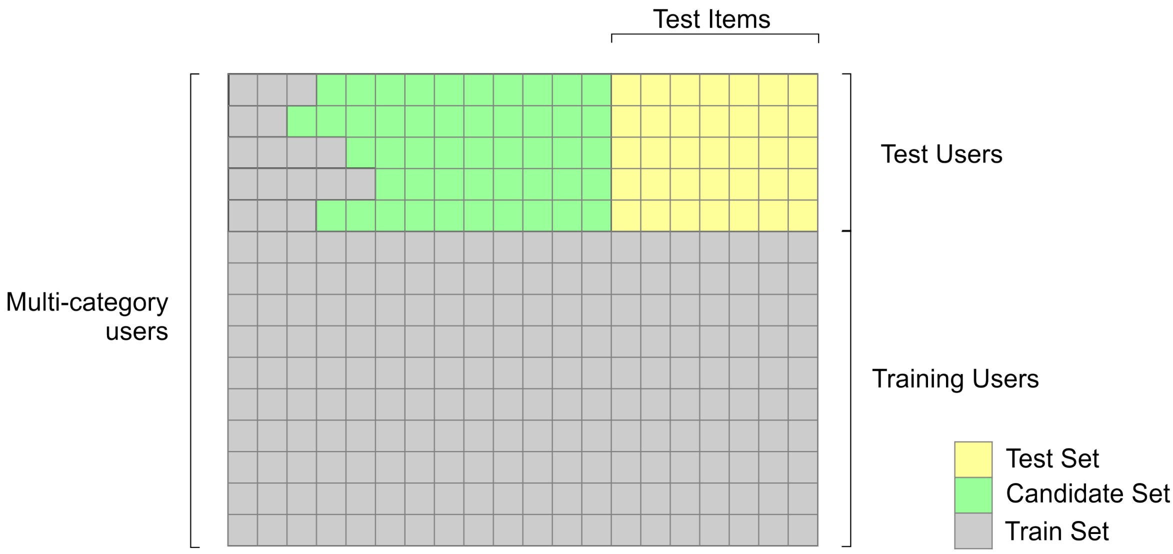


Fig. 6.2 cross-domain POI recommendation

We investigated the models' recommendation performances at different numbers of POI visited by the users in the test set. Starting from no POI visit, i.e., the extreme new user problem, we increment the number of POI in the users visit history one at a time. A POI recommendation algorithm computes a ranking score for each candidate POI (i.e., POI that the user has not yet visited) and returns the top-N highest ranked POIs as recommendations to the targeted user. We use two popular metrics to evaluate the performance of the different recommendation algorithms, namely precision@N and recall@N. In our experiment, we test the performance when N = 5, 10, 20 with 5 as the default value.

6.6 Results

In order to present results in the rest of the section, we differentiate between two cold start states according to the size of the user's profile:

- *Extreme* cold start, in this state none of the POIs already visited by the active[3] user is considered during the training phase of the evaluated models.
- *Moderate* cold start, at least one POI visited by the active user is used by the models during training and their performance is evaluated incrementally as the number of POI visited is increased.

First, the similarity scores between the categories considered for scenario I and scenario II and described in section 6.5 are presented in table 6.3 and 6.4 respectively. The similarity scores were calculated from Jensen Shannon Divergence metric in equation 6.2. It can be seen in table 6.3 and 6.4 that JSD similarity score is lowest for Food and Restaurant Categories at a value of 0.498 and 0.190 respectively. As introduced in section 6.4.1, the JSD metric scores are used to model the contributions of similar categories to the personalised ranking score.

Table 6.3 Jensen Shannon metric scores for Restaurant, Food and Nightlife categories when there are no intersect of users or POI across categories.

Category Pair	Metric Score
Food - Restaurant	**0.498**
Restaurant - Nightlife	0.597
Food - Nightlife	0.604

[3]Active users here refers to users in the test set that are used as ground truths

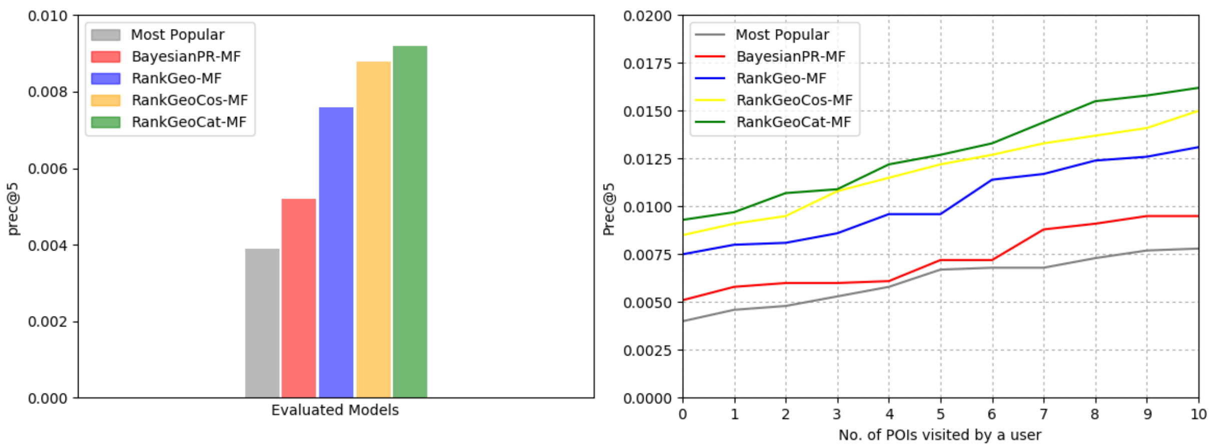
Table 6.4 Jensen Shannon metric scores for Restaurant, Food and Nightlife categories when there are intersect of users or POI across categories.

Category Pair	Metric Score
Food - Restaurant	**0.190**
Restaurant - Nightlife	0.218
Food - Nightlife	0.350

In all the experiments the hyperparameter ε was set to 0.3 following Liu et al. (2015). These values were used as an initial guide and adopted after validation with a grid search algorithm. The rest of the hyperparameters (k - dimension and K - nearest POIs) were also tuned to optimal values using a grid search on the validation set. We found k dimension to be optimal at 300 and K nearest neighbour at 100.

First Scenario: Multi-category without intersecting Users and POIs

The first experiment was carried on the models of section 6.5.2 in a setting where the categories were disjointed, i.e. no intersection of users and POIs among the categories. The performance of the models at different cold start states is presented in figure 6.3. Evaluation of the models started with an extreme cold-start condition where the profile size of users reserved as ground truths (i.e. testing set) was zero. At the extreme cold start setup and

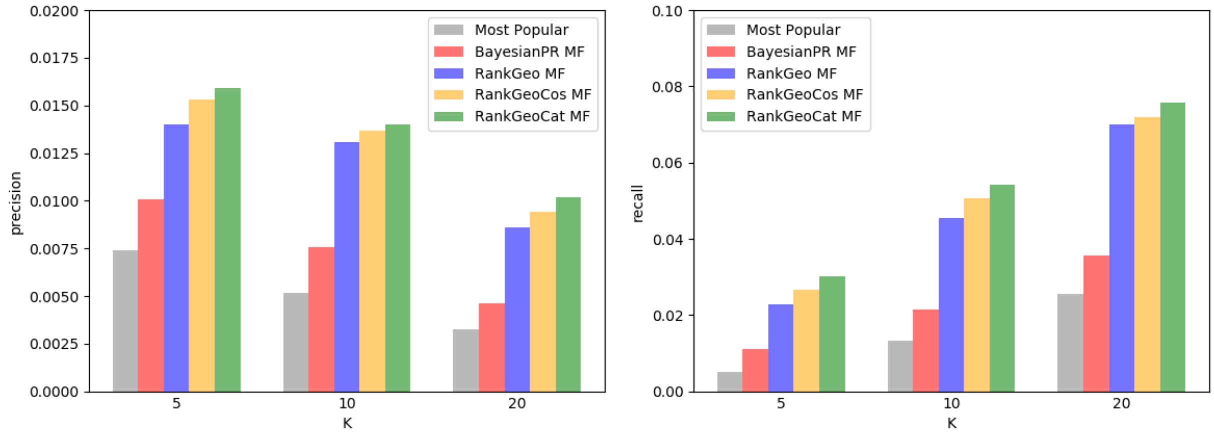


(a) Prec@5 for extreme cold start.

(b) Prec@5 for different user profile sizes.

Fig. 6.3 Evaluated models at different user profile sizes for extreme and moderate cold starts with no intersection of users or POIs among categories

considering precision when 5 POIs are recommended, figure 6.3(a) showed that models



(a) Precision at different K.

(b) Recall at different K.

Fig. 6.4 Performance of evaluated models with increasing number of recommend POIs, in scenarios where there are no intersecting users and POIs among categories.

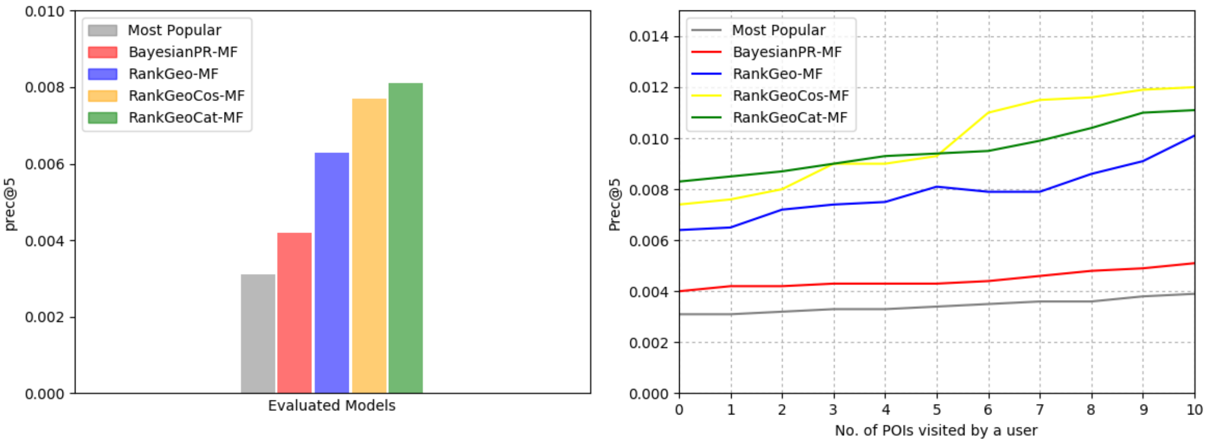
which included both location and category information as context (i.e. RankGeo-Cat MF and RankGeo-Cos MF) performed better than the ones that included only location as context (i.e. RankGeo MF). RankGeo-Cos MF had 9% better performance than RankGeo MF for precision at k=5, while RankGeo-Cat MF had 15% better performance than RankGeo MF. The performance of all the models improved as users' profile size was increased at an interval of one POI until 10 POIs were added to the profile of the active users.

The performance of the models in terms of precision and recall was evaluated at user profile sizes of at least 10 POIs. The results are observed as the number of POIs recommended (k) are varied. The precision and recall values are recorded at k = 5, 10 and 20 and presented in figure 6.4. Looking at figure 6.4(a) and (b), it is clear that the "Most popular" model which ranks POIs based on popularity (i.e. most visited) performed worst for both precision and recall at all evaluated k. As can be seen in figure 6.4, the performance of RankGeo MF, RankGeo-Cos MF and RankGeo-Cat MF are all comparable and better than Most Popular and BayesianPR MF which do not use context in computing ranking scores. However, RankGeo-Cat MF performed better than the rest of the models as the number of POI recommended is varied. RankGeo-Cat MF had the highest precision when 5 POIs are recommended (i.e. k=5) to the users with a value that is 4.17% higher than RankGeo-Cos MF. In comparison with the RankGeo MF when at its best performance, RankGeo-Cat MF had a 13.64% increase in precision at k=5. The second best performing model is RankGeo-Cos MF which also achieved its highest precision value at k=5. RankGeo-Cos MF had a 9.09% increase in performance when compared to RankGeo MF at k=5.

Second Scenario: Multi-category with Users and POIs Intersect

In the second scenario, there were users and POIs occurring in more than one category. Experiments were first carried out to evaluate the model's performance at extreme and moderate cold start states. At the extreme cold start state, it is apparent from figure 6.5 that approaches that modelled context such as geographic and category influence into their function for ranking POIs performed better than those that did not. The model that considered geographic influence only performed less than those that included both geographic and category influence. Specifically, RankGeo-Cat MF performed best with an improvement of 5.2% over RankGeo-Cos MF at extreme cold start condition and when 5 POIs were recommended to the users. RankGeo-Cat MF performed better than the RankGeo MF with an increase in recommendation precision of 28.57% at k=5 and at extreme cold state. Generally, as user profile size is increased and sparsity decreases, the performance of all the models evaluated increased. An interesting aspect of the graph in figure 6.5(b) is the change in best performing model from RankGeo-Cat MF to RankGeo-Cos FM at user profile size of at least 6 POIs. The performance of RankGeo-Cos MF increased by 18.28% when more than 6 POIs were added to the active users' profile.

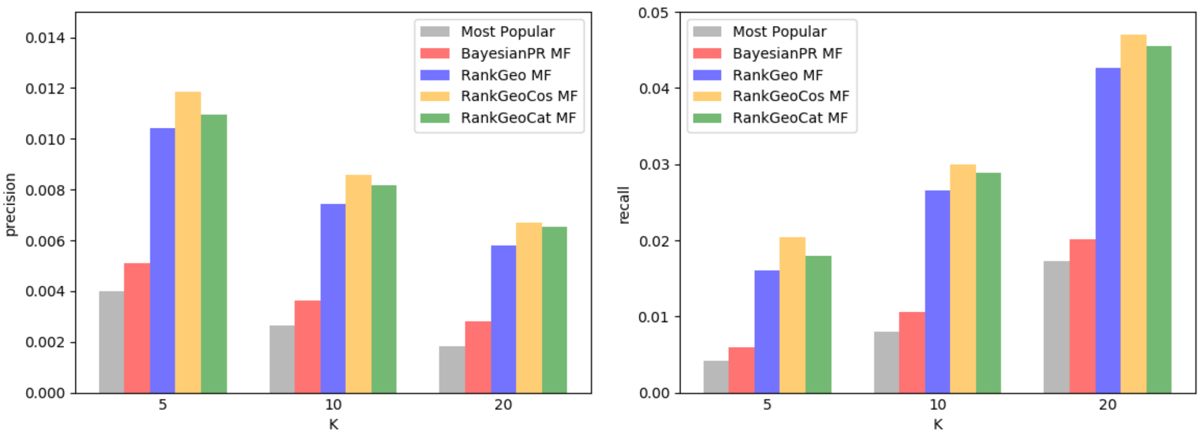
At user profile size of 10, further experiments were carried out to evaluate the models' performance as the number of POIs (k) are increased. As can be seen in figure 6.6(a) and (b), the performance of models that included location and category as contextual information once again outperformed the ones that did not in terms of precision and recall of recommended POIs.



(a) Prec@5 for exterme cold start with user/POI intersection.

(b) Prec@5 for different user profile sizes and user/POI intersection.

Fig. 6.5 Evaluated models at different user profile sizes for extreme and moderate cold start when there are intersecting users and POIs among the categories.



(a) Precision at different K.

(b) Recall at different K.

Fig. 6.6 Performance of evaluated models with increasing number of recommend POIs, in scenario where there are intersecting users and POIs among categories.

RankGeo-Cos MF and RankGeo-Cat MF outperformed their base model (RankGeo MF) at k = 5,10 and 20. Specifically, at k=5 when all the models achieved the best recommendation precision, RankGeo-Cos MF outperformed RankGeo-Cat MF and RankGeo MF with an improvement of 7.97% and 13.71% respectively. RankGeo-Cat MF which models JSD similarity between categories as weights into a function that ranks POI performed better at

5.32% higher precision than RankGeo MF which does not model category influences into its ranking function.

6.7 Conclusion

The experiments reported in this chapter have shown that modelling geographic and category influence into POI ranking functions improves the performance of a POI recommender system. The improvement in performance was observed for cases with new users and at high data sparsity for scenarios. Also, the models proposed outperformed other traditional models in scenarios with and without overlapping users and POIs.

The two models proposed consistently performed better against other models in a 5-fold cross-validation setting. Consequently, we can conclude from the cross-validation results that POI models that add the influence of category similarities to a personalised ranking fuction for POI recommendation improve the accuracy of recommendation.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we set out to address two of the challenges of collaborative recommender systems. We considered the well known cold start and sparsity problem. Information consumers and system users are generally cautious with sharing their preference information and are only more willing to share them when some form of incentives are offered. Paid surveys, loyalty cards or access to social media platforms are examples of incentives offered by companies in exchange for users' preference data. The recent high profile breaches[1] in data protection law which affected millions of users can potentially make users more sceptical of sharing their preferences data. The nonavailability of historic preference data from users will, in turn, make development and successful implementation of effective recommender systems more challenging. Recommender system models proposed in this thesis can use knowledge sources from an auxiliary domain to make recommendations for users in a target domain. Models that function as such may relieve users from the constant need to supply their preferences when interacting with items in different application domains.

Specifically, in this research work, we alleviated the effect of cold start and sparsity on rating prediction accuracy in a target domain by implementing a cross-domain recommender system. We leveraged the available metadata in the form of social tags in a dense auxiliary domain to augment user and item latent factors in a matrix factorisation model. More significantly, we went beyond traditional string pattern matching of the tags to consider inclusion of tags with similar concept into the matrix factorisation model.

[1]The Facebook-Cambridge Analytica data breach in 2018.
https://www.bbc.co.uk/news/topics/c81zyn0888lt/facebook-cambridge-analytica-data-breach

We first introduced a method based on concept coverage to determine the most suitable semantic similarity measure for the tag-pairs in a movie and book domain datasets. The method informed our decision in selecting the most appropriate semantic metric from a set of well known lexical similarity metrics. We then investigated the assumption that rating prediction accuracy improves as the size of the set of inter-domain tags across a target and auxiliary domain increases. The selected semantic similarity metric was used to measure tag similarities and vary the number of social tags shared between a movie and book domain. Furthermore, a new augmentation to a tag-based matrix factorisation model for cross-domain collaborative filtering was proposed to incorporate semantically related social tags as new latent parameters. Finally, the concept of cross-domain recommendation was extended to the field of POI recommendations. We extended a recent POI recommender model to use categories of POI as context information in recommending POI to users. More importantly, we implemented the model across multiple categories and tested the model's performance at handling cold-start and sparsity problems.

In this chapter we present our concluding statements and summarise the main themes that have emerged from our research activities. In Section 7.1 we summarise the work of the thesis and present the contributions that the research work adds to the body of work on cross-domain recommender systems. In Section 7.2 we describe potential research issues for future work.

7.2 Contributions and Summary

In this section, we summarise and highlight the main findings and contributions of this thesis, and show how we have addressed the research questions raised in Chapter 1.

RQ1. Can a cross-domain recommender model perform better when the size of intersect between the set of tags in a target and auxiliary domain increases?

In Chapter 4 of this thesis, we examined the effects on rating prediction accuracy when there is an increase in the number of inter-domain tags between a movie and book domain.

Contribution 1. *The empirical findings in chapter 4 provided a new understanding of the performance of tag-based cross-domain MF models when the size of inter-domain tags is increased using the dataset from a movie and book domain.*

In this section, we note that the matrix factorisation approach used is based on a variety of standard matrix factorisation known as SVD++. We did not develop a new MF approach in this chapter but adopted the MF model (TagGSVD++) proposed by Fernández-Tobías (2016) which is an extension of SVD++. The contribution stated above is achieved by the research activities in this chapter as summarised below:

- We formulated a semantic enhancement approach to modelling Tag-based cross-domain recommenders. We adapted the latent tag-factor parameter in a tag-based MF model (TagGSVD++) to include semantically related tags. A pre-selected semantic relatedness metric was used to enlarge the size of intersect of tag sets from a movie and book domain by grouping different together tags that have high relatedness score as inter-domain tags.
- Through evaluation experiments with real-world datasets from a movie and book domain, we demonstrate that the formulated approach is feasible for cross-domain recommendation. However, the semantic enhancement of the tag-based cross-domain MF models did not result in a significant difference in accuracy of predicted rating for the movie and book domain.

RQ2. Can semantically related tags improve performance of cross-domain recommender model when they are included as additional parameters to the model?

In contrast to the model presented in chapter 4, a new model which extends TagGSVD++ model proposed by Fernández-Tobías and Cantador (2014) is introduced in chapter 5. The proposed model enhanced TagGSVD++ with new latent parameters to account for semantically related tags between a target and auxiliary domain. In addition to knowledge base semantic metrics, semantic relatedness of tags was computed using corpus-based semantic metrics. We follow the general idea of specialising word embeddings by using domain dependent corpora for training the word embeddings as opposing to using generic corpora.

Contribution 2. *This work contributes to existing knowledge by extending the tag-based cross-domain MF model (TagGSVD++) with the addition of new parameters to account for the influence of both unique tags and semantically related tags.*

As in the previous contribution, we note that the MF approach used is based on a variety of standard matrix factorisation known as SVD++. While we did not develop a new MF approach, we extended the SVD++ model with new latent parameters. The contribution in this chapter are achieved by the theoretical models and research work summarised below:

- We present an alternative approach to semantically enhanced cross-domain recommender model by proposing the addition of two new latent parameters to the tag-based cross-domain model. One parameter modelled the influence of unique tags, while the other parameter modelled the influence of semantically related tags.

- We present a variant of the model where the semantically related tags for the second latent parameter were obtained using corpus-based semantic similarity. A series of experiments using corpora from a movie and book domain showed the value of utilising word embeddings with domain knowledge for semantically enhanced cross-domain metrics.

RQ3. Can performance of a multi-category POI recommender be improved by incorporating category similarity as context into the model?

We have so far considered social tags and ratings as inferred and explicit latent factors in an MF-based model for rating predictions. As a point of divergence from the previous chapters, we consider recommending items (i.e. item recommendation task) instead of predicting item ratings in Chapter 6. We focused on a sub-field of recommenders systems known as Point-of-interest (POI) recommendation and investigated how cross-domain recommender approaches can be adapted to outperform conventional POI recommendation techniques. Point-of-interest (POI) recommendation is an essential service to Location-Based Social Networks (LBSNs) that can benefit both users and businesses when their performance is optimal.

Contribution 3. *This is the first study to investigate the effect of including category similarity as contextual information to multi-category POI ranking model for improved POI recommendation score and performance.*

Similarly to previous chapters, contributions in this chapter did not involve the development of new MF approach. A variant of the standard SVD++ MF version which is popular for been adaptable is once again used in this chapter to include context information. The contributions in this chapter are highlighted by the models and experiments summarized below:

- We propose a novel multi-category POI ranking model that uses additional context information in the form of category type to score and recommend POI to users. A new latent parameter is included to the scoring function of a popular ranking recommender (RankGeoFM) to model the interest of users for location categories similar to categories of locations they have previously visited.

- We present a variant of the model where the weights assigned to the parameter that models the influence of category types are calculated from Jensen Shannon Divergence (JSD) similarity metric. The JSD similarity value indicated how adaptable the categories POI are for multi-category POI recommendation.
- Extensive experiments on real-life location based datasets from 3 different POI categories (Food, Restaurant and Nightlife) were carried out to demonstrate that models which include category similarities as additional context can outperform state-of-the-art methods significantly in POI recommendation.

7.3 Future Research Directions

In this thesis, we have presented models that can make use of data available in a dense auxiliary domain to recommend items to users in a target domain with cold-start and sparsity problems. The findings in our experiments, however, shows that exploiting additional information from a dense domain for recommendations in a sparse target domain may not improve the performance of a cross-domain recommender model. Specifically when we used datasets from a movie and a book domain we have concluded that there is no difference in performance of cross-domain recommender models after experimenting with varying inter-domain tags from two particular datasets (MovieLens and LibraryThing) domains. Utilising datasets from other domains for the same experiment may result in a more significant difference in the performance of the cross-domain model. In practice, a system should decide in advance whether the dataset from an auxiliary domain is worth being exploited for cross-domain recommendation or not. An automatic technique of intelligently determining if datasets from an auxiliary domain should be considered will be advantageous to the cross-domain recommender systems. Finding out the adequacy of an auxiliary domain before use in tag-based cross-domain recommender systems is an interesting direction for future research work. The following subsections provide details of other open research issues that emerged from our work.

7.3.1 Future Research for Tag-Based Cross-Domain Recommenders

We note here that it is possible to go beyond single text matching (i.e. tag to tag) between two domains. Text matching methods are usually restricted to lexical categories. We also note that the similarity of tags used is limited to the total number of word concepts available in WordNet. In future, we want to explore more extensive knowledge base networks (e.g.

DBpedia) and more advanced text similarity methods such as those that use bi-grams and considers the context of the tags. Therefore future direction will include using larger lexical knowledge-base such as DBpedia in computing semantic similarity. Also, the work in this thesis, we have mainly computed numeric numbers and used Mean absolute error to determine the accuracy of our model at predicting missing ratings. We did not consider other important factors that may affect user preferences such as the design of the interface of systems that will serve as the front end to the models we have investigated. Our model may have performed well offline, but there are other practical components of creating a successful recommender system that was not within the scope of our work. Optimising the models to take account of memory and time complexity is another possible area of future research.

7.3.2 Future Research for Next POI Recommenders

Regarding improvement of performance of multi-category POIs recommenders, we did not consider a cross-category scenario where one category is set as the target and the others as the auxiliary. Results from a multi-category scenario show that the application of our model to a cross-category scenario is a promising direction for future work. In order to improve POI recommendation to users, recent research works have added the sequential behaviour of human movement to POI recommender models. The additional detail in users movement is essential for POI recommendation because human movement has been shown to follow sequential patterns (Cheng et al. (2013)). The new POI recommender models are used to predict the next POI a user is most likely to visit and known as *Next POI recommenders* (Feng et al. (2015)). Next POI recommender systems can benefit from cross-domain recommendation approaches because cross-domain approaches enhance transfer of knowledge which in this case may be sequential patterns of users movement. The application of cross-domain approaches to Next POI recommenders presents a new line of future research work.

7.3.3 Future Research for Knowledge-Based Recommenders

Different types of content and collaborative filtering models have been proposed and used for improving recommendations in domains of quality and taste products such as books, music and movies. However, more complex product domains such as cars, Computers, apartments, or financial services present a different kind of recommendation problem (such as rarer purchases and feedback). Knowledge-based recommender technologies have been known to be best suited to tackle such recommendation challenges by exploiting explicit user requirements and deep knowledge about the underlying product domain for the calculation of recommendations (Felfernig et al. (2007)).

In contrast to collaborative filtering and content-based recommender models, knowledge-based recommenders do not have cold-start problems because user requirements are generated during the recommendation session. According to Felfernig and Burke (2008), there are two basic procedures to be considered when implementing a knowledge-based recommender application. One is creating a recommender knowledge base and the second involves running a knowledge base recommender process. Extracting domain knowledge for creating the knowledge base require enormous manual work by domain experts before the recommendations can be generated in the knowledge-based recommender application. For this reason, using knowledge transfer techniques in Cross-domain recommender approaches will be an interesting direction for future work. This is because utilizing models that can transfer domain knowledge which has been manually curated from an auxiliary domain to a target domain may help reduce the manual efforts required from domain experts. The techniques used to semantically enhance tag-based cross-domain models may also be applicable in the process of transferring knowledge from a domain with an older knowledge base to a newer domain of interest. Specifically, the semantic relatedness of keywords extracted from an auxiliary domain with already created knowledge base can be compared to keywords of a newer target domain. The average semantic relatedness score between keywords of the auxiliary and target domains may help determine if the domains are similar enough to be considered for transfer of other properties of the knowledge base.

References

Abel, F., Herder, E., Houben, G., Henze, N., and Krause, D. (2011). Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction*, pages 1—-42.

Adomavicius, G. and Tuzhilin, A. (2005). 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):141–178.

Adomavicius, G. and Tuzhilin, A. (2008). Context-aware recommender systems. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 335–336, New York, NY, USA. ACM.

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., and Soroa, A. (2009). A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 19–27, Stroudsburg, PA, USA. Association for Computational Linguistics.

Aslam, J. A. and Pavlu, V. (2007). Query hardness estimation using jensen-shannon divergence among multiple scoring functions. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 198–209, Berlin, Heidelberg. Springer-Verlag.

Azak, M. (2009). Crossing: A framework to develop knowledge-based recommenders in cross domains.

Baker, K. (2005). Singular value decomposition tutorial.

Bamman, D., O'Connor, B., and Smith, N. A. (2013). Learning latent personas of film characters. In *ACL (1)*, pages 352–361. The Association for Computer Linguistics.

Bamman, D. and Smith, N. A. (2013). New alignment methods for discriminative book summarization. *CoRR*, abs/1305.1319.

Barbieri, N., Manco, G., and Ritacco, E. (2014). *Probabilistic Approaches to Recommendations*. Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool Publishers.

Belfiore, J. (2015). Introducing project spartan: The new browser built for windows 10. [Online; posted 30-March-2015].

Berkovsky, S., Kuflik, T., and Ricci, F. (2007). Distributed collaborative filtering with domain specialization. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, pages 33–40, New York, NY, USA. ACM.

Berkovsky, S., Kuflik, T., and Ricci, F. (2008). Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction*, 18(3):245–286.

Bilgic, M. and Mooney, R. J. (2005). Explaining recommendations: Satisfaction vs. promotion. In *Proceedings of Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research at the 2005 International Conference on Intelligent User Interfaces*, San Diego, CA.

Bill, R., Liu, Y., McInnes, B. T., Melton, G. B., Pedersen, T., and Pakhomov, S. V. S. (2012). Evaluating semantic relatedness and similarity measures with standardized meddra queries. In *AMIA 2012, American Medical Informatics Association Annual Symposium, Chicago, Illinois, USA, November 3-7, 2012*.

Braunhofer, M., Elahi, M., and Ricci, F. (2015). User personality and the new user problem in a context-aware point of interest recommender system. In Tussyadiah, I. and Inversini, A., editors, *Information and Communication Technologies in Tourism 2015*, pages 537–549, Cham. Springer International Publishing.

Breese, J. S., Heckerman, D., and Kadie, C. (1998). 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. Morgan Kaufmann Publishers Inc.

Briët, J. and Harremoës, P. (2009). Properties of classical and quantum jensen-shannon divergence. *Phys. Rev. A*, 79:052311.

Brockmann, D., Hufnagel, L., and Geisel, T. (2006). The scaling laws of human travel. *Nature*, 439(7075):462–465.

Budanistsky, A. and Hirst, G. (2005). Evaluating wordnet-based measures of lexical semantic relatedness. *Association for Computational Linguistics*, 32(1):13–47.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370.

Burke, R. D., Felfernig, A., and Göker, M. H. (2011). Recommender systems: An overview. *AI Magazine*, 32(3):13–18.

Candillier, L., Meyer, F., and Boullé, M. (2007). Comparing state-of-the-art collaborative filtering systems. In Perner, P., editor, *Machine Learning and Data Mining in Pattern Recognition*, volume 4571 of *Lecture Notes in Computer Science*, pages 548–562. Springer Berlin Heidelberg.

Cantador, I. and Castells, P. (2006). Multilayered semantic social network modeling by ontology-based user profiles clustering: Application to collaborative filtering. In *Proceedings of the 15th International Conference on Managing Knowledge in a World of Networks*, EKAW'06, pages 334–349.

Cantador, I., Castells, P., and Bellogín, A. (2011). An enhanced semantic layer for hybrid recommender systems: Application to news recommendation. *Int. J. Semant. Web Inf. Syst.*, 7(1):44–78.

Cantador, I. and Cremonesi, P. (2014). Tutorial on cross-domain recommender systems. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 401–402, New York, NY, USA. ACM.

Cantador, I., Fernández-Tobías, I., and Bellogín, A. (2013). Relating personality types with user preferences in multiple entertainment domains. In *UMAP Workshops*, volume 997 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Cao, B., Liu, N. N., and Yang, Q. (2010). Transfer learning for collective link prediction in multiple heterogenous domains. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 159–166, USA. Omnipress.

Chen, X., Qin, Z., Zhang, Y., and Xu, T. (2016). Learning to rank features for recommendation over multiple categories. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 305–314, New York, NY, USA. ACM.

Cheng, C., Yang, H., King, I., and Lyu, M. R. (2012). Fused matrix factorization with geographical and social influence in location-based social networks. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 17–23. AAAI Press.

Cheng, C., Yang, H., Lyu, M. R., and King, I. (2013). Where you like to go next: Successive point-of-interest recommendation. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 2605–2611. AAAI Press.

Cho, E., Myers, S. A., and Leskovec, J. (2011). Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1082–1090, New York, NY, USA. ACM.

Chung, R., Sundaram, D., and Srinivasan, A. (2007). Integrated personal recommender systems. In *Proceedings of the Ninth International Conference on Electronic Commerce*, ICEC '07, pages 65–74, New York, NY, USA. ACM.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., and Sartin, M. (1999). Combining content-based and collaborative filters in an online newspaper. *SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation*.

Codina, V. and Ceccaroni, L. (2010). Taking advantage of semantics in recommendation systems. In *Proceedings of the 2010 Conference on Artificial Intelligence Research and Development: Proceedings of the 13th International Conference of the Catalan Association for Artificial Intelligence*, pages 163–172. IOS Press.

Codina, V. and Ceccaroni, L. (2012). *Semantically-Enhanced Recommenders: In Ria~no, D., Onaindia, E., and Cazorla, M. (eds.): Artificial Intelligence Research and Development - Proceedings of the 15th International Conference of the Catalan Association for Artificial Intelligence*. IOS Press.

Cremonesi, P., Garzotto, F., Negro, S., Papadopoulos, A., and Turrin, R. (2011). Comparative evaluation of recommender system quality. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, pages 1927–1932, New York, NY, USA. ACM.

Cremonesi, P., Koren, Y., and Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 39–46, New York, NY, USA. ACM.

Crook, T., Frasca, B., Kohavi, R., and Longbotham, R. (2009). Seven pitfalls to avoid when running controlled experiments on the web. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 1105–1114, New York, NY, USA. ACM.

Degemmis, M., Lops, P., and Semeraro, G. (2007). A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction*, 17(3):217–255.

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

Elahi, M., Ricci, F., and Rubens, N. (2016). A survey of active learning in collaborative filtering recommender systems.

Enrich, M., Braunhofer, M., and Ricci, F. (2013). Cold-start management with cross-domain collaborative filtering and tags. *In Proceedings of the 14th International Conference on E-commerce and Web Technologies*, pages 101–112.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E. H., and Smith, N. A. (2014). Retrofitting word vectors to semantic lexicons. *CoRR*, abs/1411.4166.

Felfernig, A. and Burke, R. D. (2008). Constraint-based recommender systems: Technologies and research issues. In *Proceedings of the 10th International Conference on Electronic Commerce*, ICEC '08, pages 3:1–3:10, New York, NY, USA. ACM.

Felfernig, A., Erdeniz, S. P., Jeran, M., Akcay, A., Azzoni, P., Maiero, M., and Doukas, C. (2017). Recommendation technologies for iot edge devices. *Procedia Computer Science*, 110:504 – 509. 14th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2017) / 12th International Conference on Future Networks and Communications (FNC 2017) / Affiliated Workshops.

Felfernig, A., Teppan, E., and Gula, B. (2007). Knowledge-based recommender technologies for marketing and sales. 21:333–354.

Fellbaum, C. (2005). *WordNet and wordnets. In: Brown, Keith et al. (eds.), Encyclopedia of Language and Linguistics, Second Edition*. Elsevier.

Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y. M., and Yuan, Q. (2015). Personalized ranking metric embedding for next new poi recommendation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 2069–2075. AAAI Press.

Fernández-Tobías, I. (2016). Matrix factorization models for cross-domain recommendation: Addressing the cold start in collaborative filtering.

Fernández-Tobías, I., Braunhofer, M., Elahi, M., Ricci, F., and Cantador, I. (2016). Alleviating the new user problem in collaborative filtering by exploiting personality information. *User Modeling and User-Adapted Interaction*, 26(2):221–255.

Fernández-Tobías, I., Cantador, I., Kaminskas, M., and Ricci, F. (2011). A generic semantic-based framework for cross-domain recommendation. In *Proceedings of the 2Nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '11, pages 25–32, New York, NY, USA. ACM.

Fernández-Tobías, I., Cantador, I., and Plaza, L. (2013). An emotion dimensional model based on social tags: Crossing folksonomies and enhancing recommendations. In *E-Commerce and Web Technologies - 14th International Conference, EC-Web 2013, Prague, Czech Republic, August 27-28, 2013. Proceedings*, pages 88–100.

Fernández-Tobías, I., Tomeo, P., Cantador, I., Di Noia, T., and Di Sciascio, E. (2016). Accuracy and diversity in cross-domain recommendations for cold-start users with positive-only feedback. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 119–122.

Fernández-Tobías, I. and Cantador, I. (2014). Exploiting social tags in matrix factorization models for cross-domain collaborative filtering. *In: Proceeding of the 1st International Workshop on New Trends in Content-based Recommender Systems*, pages 34–41.

Fernández-Tobías, I., Cantador, I., Kaminskas, M., and Ricci, F. (2012). Cross-domain recommender systems: A survey of the state of the art. In *Proceedings of the 2nd Spanish Conference on Information Retrieval. CERI*.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2001). Placing search in context: The concept revisited. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 406–414, New York, NY, USA. ACM.

Forbes, P. and Zhu, M. (2011). Content-boosted matrix factorization for recommender systems: Experiments with recipe recommendation. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 261–264, New York, NY, USA. ACM.

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

Gantner, Z., Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. (2011). MyMediaLite: A free recommender system library. In *5th ACM International Conference on Recommender Systems (RecSys 2011)*.

Gao, H., Tang, J., Hu, X., and Liu, H. (2013a). Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 93–100, New York, NY, USA. ACM.

Gao, H., Tang, J., Hu, X., and Liu, H. (2015). Content-aware point of interest recommendation on location-based social networks. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 1721–1727. AAAI Press.

Gao, H., Tang, J., and Liu, H. (2012). gscorr: Modeling geo-social correlations for new check-ins on location-based social networks. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1582–1586, New York, NY, USA. ACM.

Gao, S., Luo, H., Chen, D., Li, S., Gallinari, P., and Guo, J. (2013b). Cross-domain recommendation via cluster-level latent factor model. In *Proceedings, Part II, of the European Conference on Machine Learning and Knowledge Discovery in Databases - Volume 8189*, ECML PKDD 2013, pages 161–176, New York, NY, USA. Springer-Verlag New York, Inc.

Gartner, G., Cartwright, W., and Peterson, M. P. (2007). *Location Based Services and TeleCartography (Lecture Notes in Geoinformation and Cartography)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

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

Golder, S. and Huberman, B. A. (2006). The structure of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208.

Golub, G. and Van Loan, C. (1989). *Matrix Computations*. Johns Hopkins University Press, Baltimore, 2nd edition.

Gomez-Uribe, C. A. and Hunt, N. (2015). The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.*, 6(4):13:1–13:19.

Gonzalez, M. C., Hidalgo, C. A., and Barabasi, A.-L. (2008). Understanding individual human mobility patterns. *Nature*, 453(7196):779–782.

Guan, Z., Wang, C., Bu, J., Chen, C., Yang, K., Cai, D., and He, X. (2010). Document recommendation in social tagging services. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 391–400, New York, NY, USA. ACM.

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

Guo, G., Zhang, J., and Yorke-Smith, N. (2015). Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 123–129. AAAI Press.

Guy, I., Zwerdling, N., Ronen, I., Carmel, D., and Uziel, E. (2010). Social media recommendation based on people and tags. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 194–201, New York, NY, USA. ACM.

Harispe, S., Ranwez, S., Janaqi, S., and Montmain, J. (2013). *A Comprehensive Survey and a Technical Introduction to Knowledge-based Measures Using Semantic Graph Analysis*. LGI2P/EMA Research Center, Parc scientifique, France.

Harispe, S., Ranwez, S., Janaqi, S., and Montmain, J. (2015). *Semantic Similarity from Natural Language and Ontology Analysis*. Morgan & Claypool Publishers.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition.

He, R. and McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. *CoRR*, abs/1602.01585.

Heather, P. L. (2016 Accessed March 30, 2017). Gartner predicts our digital future. https://www.gartner.com/smarterwithgartner/gartner-predicts-our-digital-future/.

Herlocker, J. L., Konstan, J. A., and Riedl, J. (2000). 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. ACM.

Hill, F., Reichart, R., and Korhonen, A. (2015). Simlex-999: Evaluating semantic models with genuine similarity estimation. *Comput. Linguist.*, 41(4):665–695.

Hirst, G. and St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. In Fellbaum, C., editor, *WordNet: An Electronic Lexical Database*, pages 305–332. MIT Press.

Horozov, T., Narasimhan, N., and Vasudevan, V. (2006). Using location for personalized poi recommendations in mobile environments. In *Proceedings of the International Symposium on Applications on Internet*, SAINT '06, pages 124–129, Washington, DC, USA. IEEE Computer Society.

Hu, L., Cao, J., Xu, G., Cao, L., Gu, Z., and Zhu, C. (2013). Personalized recommendation via cross-domain triadic factorization. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 595–606, New York, NY, USA. ACM.

Jamali, M. and Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 135–142, New York, NY, USA. ACM.

Jian, J. and Conrath, D. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *In: Proceedings of International Conference Research on Computational Linguistics*, pages 30–40.

Jiang, J. and Conrath, D. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the Int'l. Conf. on Research in Computational Linguistics*, pages 19–33.

Kaminskas, M., Fernández-Tobías, I., Cantador, I., and Ricci, F. (2013). Ontology-based identification of music for places. In Cantoni, L. and Xiang, Z. P., editors, *Information and Communication Technologies in Tourism 2013*, pages 436–447, Berlin, Heidelberg. Springer Berlin Heidelberg.

Kiela, D., Hill, F., and Clark, S. (2015). Specializing word embeddings for similarity or relatedness. In Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., and Marton, Y., editors, *EMNLP*, pages 2044–2048. The Association for Computational Linguistics.

Kluver, D. and Konstan, J. A. (2014). Evaluating recommender behavior for new users. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 121–128, New York, NY, USA. ACM.

Kobsa, A., Koenemann, J., and Pohl, W. (2001). Personalised hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review*, 16(2):111–155.

Kohavi, R., Longbotham, R., Sommerfield, D., and Henne, R. M. (2009). Controlled experiments on the web: Survey and practical guide. *Data Min. Knowl. Discov.*, 18(1):140–181.

Koren, Y. (2008). 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. ACM.

Koren, Y. (2009). Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 447–456, New York, NY, USA. ACM.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender system. *IEEE Computer*, 42(8):426–434.

Leacock, C. and Chodorow, M. (1998). *Combining Local Context and WordNet Similarity for Word Sense Identification.* MIT Press, Cambridge.

Lee, D. D. and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press.

Levandoski, J. J., Sarwat, M., Eldawy, A., and Mokbel, M. F. (2012). Lars: A location-aware recommender system. In *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*, ICDE '12, pages 450–461, Washington, DC, USA. IEEE Computer Society.

Li, B., Yang, Q., and Xue, X. (2009). Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In Boutilier, C. and Boutilier, C., editors, *Proceeding of International Joint Conferences on Artificial Intelligence*, pages 2052––2057.

Li, X., Cong, G., Li, X., Pham, T.-A. N., and Krishnaswamy, S. (2015). Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

Lian, J., Zhang, F., Xie, X., and Sun, G. (2017). Cccfnet: A content-boosted collaborative filtering neural network for cross domain recommender systems. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 817–818, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Liu, B., Fu, Y., Yao, Z., and Xiong, H. (2013a). Learning geographical preferences for point-of-interest recommendation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1043–1051, New York, NY, USA. ACM.

Liu, X., Liu, Y., Aberer, K., and Miao, C. (2013b). Personalized point-of-interest recommendation by mining users' preference transition. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 733–738, New York, NY, USA. ACM.

Liu, Y., Liu, Z., Chua, T.-S., and Sun, M. (2015). Topical word embeddings. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2418–2424. AAAI Press.

Liu, Y., Pham, T.-A. N., Cong, G., and Yuan, Q. (2017). An experimental evaluation of point-of-interest recommendation in location-based social networks. *Proc. VLDB Endow.*, 10(10):1010–1021.

Loizou, A. (2009). How to recommend music to film buffs: enabling the provision of recommendations from multiple domains.

Loni, B., Shi, Y., Larson, M., and Hanjalic, A. (2014). Cross-domain collaborative filtering with factorization machines. In *Proceedings of the 36th European Conference on IR Research on Advances in Information Retrieval - Volume 8416*, ECIR 2014, pages 656–661, Berlin, Heidelberg. Springer-Verlag.

Lops, P., Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. *In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook*, pages 75–105.

Ma, H., Yang, H., Lyu, M. R., and King, I. (2008). Sorec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 931–940, New York, NY, USA. ACM.

Maidel, V., Shoval, P., Shapira, B., and Taieb-Maimon, M. (2008). Evaluation of an ontology-content based filtering method for a personalized newspaper. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 91–98. ACM.

Manzato, Garcia, M. (2013). gsvd++: Supporting implicit feedback on recommender systems with metadata awareness. *In Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 908–913.

Marlin, B. M. and Zemel, R. S. (2009). Collaborative prediction and ranking with non-random missing data. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 5–12, New York, NY, USA. ACM.

McNee, S. M., Riedl, J., and Konstan, J. A. (2006). Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 1097–1101, New York, NY, USA. ACM.

McSherry, D. (2005). Explanation in recommender systems. *Artif. Intell. Rev.*, 24(2):179–197.

Melville, P., Yang, S. M., Saar-Tsechansky, M., and Mooney, R. (2005). Active learning for probability estimation using jensen-shannon divergence. In *Proceedings of the 16th European Conference on Machine Learning*, ECML'05, pages 268–279, Berlin, Heidelberg. Springer-Verlag.

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

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Miller, G. A. and Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

Moreno, O., Shapira, B., Rokach, L., and Shani, G. (2012). Talmud – transfer learning for multiple domains. In *Proceeding of the 21st ACM Conf. on Information and Knowledge Management*, pages 425–434.

Oard, D. (1997). The state of the art in text filtering. *User Modeling and User-Adapted Interaction*, 7(3):141–178.

O'Donovan, J. and Smyth, B. (2005). Trust in recommender systems. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, IUI '05, pages 167–174, New York, NY, USA. ACM.

Pagano, R., Quadrana, M., Elahi, M., and Cremonesi, P. (2017). Toward active learning in cross-domain recommender systems. *CoRR*, abs/1701.02021.

Pakhomov, S., McInnes, B., Adam, T., Liu, Y., Pedersen, T., and Melton, G. B. (2010). Semantic Similarity and Relatedness between Clinical Terms: An Experimental Study. *Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2010:572–576.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10):1345–1359.

Pan, W., Xiang, E. W., and Yang, Q. (2012). Transfer learning in collaborative filtering with uncertain ratings. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 662–668. AAAI Press.

Papagelis, M., Plexousakis, D., Kutsuras, Themistoklis", e. P., Issarny, V., and Shiu, S. (2005). Alleviating the sparsity problem of collaborative filtering using trust inferences. In *Trust Management*, pages 224–239. Springer Berlin Heidelberg", address="Berlin, Heidelberg.

Park, M.-H., Hong, J.-H., and Cho, S.-B. (2007). Location-based recommendation system using bayesian user's preference model in mobile devices. In *Proceedings of the 4th International Conference on Ubiquitous Intelligence and Computing*, UIC'07, pages 1130–1139, Berlin, Heidelberg. Springer-Verlag.

Pazzani, Micheal, J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13:393–408.

Pedersen, T. (2010). Information content measures of semantic similarity perform better without sense-tagged text. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 329–332.

Peis, E., Morales-del Castillo, J., and Delgado-Lopez, J. (2008). Semantic recommender systems. analysis of the state of the topic. *Hipertext.net*, 6(17):1–5.

Ponomareva, N. and Thelwall, M. (2012). Do neighbours help?: An exploration of graph-based algorithms for cross-domain sentiment classification. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 655–665, Stroudsburg, PA, USA. Association for Computational Linguistics.

Quadrana, M., Cremonesi, P., and Jannach, D. (2018). Sequence-aware recommender systems. *ACM Comput. Surv.*, 51(4):66:1–66:36.

Remus, R. (2012). Domain adaptation using domain similarity- and domain complexity-based instance selection for cross-domain sentiment analysis. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining Workshops (ICDMW 2012) Workshop on Sentiment Elicitation from Natural Text for Information Retrieval and Extraction (SENTIRE)*, pages 717–723.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States. AUAI Press.

Resnick, P. and Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3):56–58.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *In: Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1:448–453.

Resnik, P. et al. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res.(JAIR)*, 11:95–130.

Ricci, F., Rokach, L., and Shapira, B. (2011). *Introduction to recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.).* Springer, New York.

Roelleke, T. and Wang, J. (2008). Tf-idf uncovered: A study of theories and probabilities. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 435–442, New York, NY, USA. ACM.

Roh, T. H., Oh, K. J., and Han, I. (2003). The collaborative filtering recommendation based on som cluster-indexing cbr. *Expert Systems with Applications*, 25(3):413–423.

Rowe, M. (2014). Semanticsvd++: Incorporating semantic taste evolution for predicting ratings. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 01*, WI-IAT '14, pages 213–220, Washington, DC, USA. IEEE Computer Society.

Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Commun. ACM*, 8(10).

Sahebi, S. and Brusilovsky, P. (2013). Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation. In Carberry, S., Weibelzahl, S., Micarelli, A., and Semeraro, G., editors, *User Modeling, Adaptation, and Personalization*, pages 289–295, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sahebi, S., Brusilovsky, P., and Bobrokov, V. (2017). Cross-domain recommendation for large-scale data. In *RecSysKTL*, volume 1887 of *CEUR Workshop Proceedings*, pages 9–15. CEUR-WS.org.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA. ACM.

Sarwat, M., Levandoski, J. J., Eldawy, A., and Mokbel, M. F. (2014). Lars*: An efficient and scalable location-aware recommender system. *IEEE Trans. on Knowl. and Data Eng.*, 26(6):1384–1399.

Schafer, J. B., Konstan, J. A., and Riedl, J. (2001). E-commerce recommendation applications. *Data Min. Knowl. Discov.*, 5(1-2):115–153.

Schmidt-Thieme, L. (2005). Compound classification models for recommender systems. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, ICDM '05, pages 378–385, Washington, DC, USA. IEEE Computer Society.

Schwartz, H. A. and Gomez, F. (2011). Evaluating Semantic Metrics on Tasks of Concept Similarity. In *FLAIRS Conference*.

Sen, S., Lam, S. K., Rashid, A. M., Cosley, D., Frankowski, D., Osterhouse, J., Harper, F. M., and Riedl, J. (2006). Tagging, communities, vocabulary, evolution. In *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work*, CSCW '06, pages 181–190, New York, NY, USA. ACM.

Sen, S., Vig, J., and Riedl, J. (2009). Tagommenders: Connecting users to items through tags. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 671–680, New York, NY, USA. ACM.

Shani, G. and Gunawardana, A. (2011). *Evaluating Recommendation Systems*. Springer US, Boston, MA.

Shapira, B., Rokach, L., and Freilikhman, S. (2013). Facebook single and cross domain data for recommendation systems. *User Modeling and User-Adapted Interaction*, 23(2):211–247.

Shi, Y., Larson, M., and Hanjalic, A. (2011). Tags as bridges between domains: improving recommendation with tag-iduced cross-domain collaborative filtering. *In Proceeding of the 19th international conference on User modeling, adaption, and personalization. UMAP'11*, pages 305–316.

Shi, Y., Larson, M., and Hanjalic, A. (2013). Generalized tag-induced cross-domain collaborative filtering. *Computing Research Repository*, abs/1302.4888.

Sieg, A., Mobasher, B., and Burke, R. (2010). Improving the effectiveness of collaborative recommendation with ontology-based user profiles. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '10, pages 39–46. ACM.

Singh, A. P. and Gordon, G. J. (2008). Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 650–658, New York, NY, USA. ACM.

Sinha, R. and Swearingen, K. (2002). The role of transparency in recommender systems. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '02, pages 830–831, New York, NY, USA. ACM.

Steck, H. (2010). Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 713–722, New York, NY, USA. ACM.

Stewart, A., Diaz-Aviles, E., Nejdl, W., Marinho, L. B., Nanopoulos, A., and Schmidt-Thieme, L. (2009). Cross-tagging for personalized open social networking. In *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*, HT '09, pages 271–278, New York, NY, USA. ACM.

Tintarev, N. and Masthoff, J. (2007). A survey of explanations in recommender systems. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, ICDEW '07, pages 801–810, Washington, DC, USA. IEEE Computer Society.

Tiroshi, A., Berkovsky, S., Kaafar, M. A., Chen, T., and Kuflik, T. (2013). Cross social networks interests predictions based ongraph features. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 319–322, New York, NY, USA. ACM.

Tiroshi, A. and Kuflik, T. (2012). Domain ranking for cross domain collaborative filtering. In *Proceedings of the 20th International Conference on User Modeling, Adaptation, and Personalization*, UMAP'12, pages 328–333, Berlin, Heidelberg. Springer-Verlag.

Tobler, W. (1970). A computer movie simulating urban growth in the detroit region. *Economic Geography*, 46(sup1):234–240.

Trask, A., Michalak, P., and Liu, J. (2015). sense2vec - A fast and accurate method for word sense disambiguation in neural word embeddings. *CoRR*, abs/1511.06388.

Tso-Sutter, K. H. L., Marinho, L. B., and Schmidt-Thieme, L. (2008). Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, pages 1995–1999, New York, NY, USA. ACM.

Uchygit, G. (2010). *Data Mining Techniques for Web Personalization: Algorithms and Applications. In: Ali, S., Xiang, Y. eds. eds. Dynamic and Advanced Data Mining for Progressing Technological Development: Innovations and Systemic Approaches*. Information Science reference. New York.

Uchyigit, G., Clark, K. L., and Coullon, D. (2007). KEXPLORATOR - A 2d map exploration user interface for recommender systems. In *ICEIS (5)*, pages 223–228.

Usunier, N., Buffoni, D., and Gallinari, P. (2009). Ranking with ordered weighted pairwise classification. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1057–1064, New York, NY, USA. ACM.

Vig, J., Sen, S., and Riedl, J. (2009). Tagsplanations: Explaining recommendations using tags. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, pages 47–56, New York, NY, USA. ACM.

Vozalis, M. and Margaritis, K. G. (2006). On the enhancement of collaborative filtering by demographic data. *Web Intelli. and Agent Sys.*, 4(2):117–138.

Wang, J., de Vries, A. P., and Reinders, M. J. T. (2006). Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 501–508, New York, NY, USA. ACM.

Willmott, C. J., Matsuura, K., and Robeson, S. M. (2009). Ambiguities inherent in sums-of-squares-based error statistics. *Atmospheric Environment*, 43(3):749 – 752.

Winoto, P. and Tang, T. (2008). If you like the devil wears prada the book, will you also enjoy the devil wears prada the movie? a study of cross-domain recommendations. *New Generation Computing*, 26(3):5666–5672.

Wu, H. C., Luk, R. W. P., Wong, K. F., and Kwok, K. L. (2008). Interpreting tf-idf term weights as making relevance decisions. *ACM Trans. Inf. Syst.*, 26(3):13:1–13:37.

Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q., and Sun, J. (2010). Temporal recommendation on graphs via long- and short-term preference fusion. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 723–732, New York, NY, USA. ACM.

Yager, R. R. (2003). Fuzzy logic methods in recommender systems. *Fuzzy Sets and Systems*, 136(2):133–149.

Yang, D., Zhang, D., Yu, Z., and Wang, Z. (2013). A sentiment-enhanced personalized location recommendation system. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT '13, pages 119–128, New York, NY, USA. ACM.

Yao, L., Sheng, Q. Z., Ngu, A. H. H., and Li, X. (2016). Things of interest recommendation by leveraging heterogeneous relations in the internet of things. *ACM Trans. Internet Technol.*, 16(2):9:1–9:25.

Ye, M., Yin, P., and Lee, W.-C. (2010). Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 458–461, New York, NY, USA. ACM.

Ye, M., Yin, P., Lee, W.-C., and Lee, D.-L. (2011). Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 325–334, New York, NY, USA. ACM.

Yuan, Q., Cong, G., Ma, Z., Sun, A., and Thalmann, N. M. (2013). Time-aware point-of-interest recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 363–372, New York, NY, USA. ACM.

Yuan, Q., Cong, G., and Sun, A. (2014). Graph-based point-of-interest recommendation with geographical and temporal influences. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 659–668, New York, NY, USA. ACM.

Zhang, J.-D. and Chow, C.-Y. (2015). Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 443–452, New York, NY, USA. ACM.

Zhang, M. and Hurley, N. (2008). Avoiding monotony: Improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 123–130.

Zhang, Y., Cao, B., and Yeung, D. (2012). Multi-domain collaborative filtering. *Computer Research Repository*, abs/1203.3535.

Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., and Ma, S. (2014). Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 83–92, New York, NY, USA. ACM.

Zhang, Z., Jin, X., Li, L., Ding, G., and Yang, Q. (2016). Multi-domain active learning for recommendation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2358–2364. AAAI Press.

Zhao, L., Pan, S. J., Xiang, E. W., Zhong, E., Lu, Z., and Yang, Q. (2013). Active transfer learning for cross-system recommendation. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI'13, pages 1205–1211. AAAI Press.

Zhao, S., King, I., and Lyu, M. R. (2016). A survey of point-of-interest recommendation in location-based social networks. *CoRR*, abs/1607.00647.

Zhong, J. and Li, X. (2010). Unified collaborative filtering model based on combination of latent features. *Expert Systems with Applications*, 37(8):5666–5672.

Appendix A

Experiment Tools

A.0.1 Software Tools/APIs

In order to satisfy the functional requirements of our model as specified in the introduction, we utilized and extended several methods and classes across three different programming languages (Python and C-sharp). We maintained connectivity and data flow between applications of the different programming languages by following a sequential order of operation. The data output from the application of a specific language is first stored in a system directory [1] from where the applications in the other languages can subsequent read and process the data.

In order to further support reproducibility of our experiments, we implemented our model using free/open source software distributed under the terms of the GNU General Public License. The Jiang-Conrath semantic similarity scores were computed using the NLTK implementation of the popular WordNet::Similarity, which is a Perl module available for download from https://sourceforge.net/projects/wn-similarity.

Natural Language Tool kit - Python

This provides a python Application Programming Interface (API) for a variety of semantic similarity and relatedness measures based on information found in the lexical database WordNet. It supports well established semantic measures such as Resnik, Lin, Jiang-Conrath, Leacock-Chodorow and Hirst-St.Onge. After selecting the Jiang-Conrath (JC) measure as the most optimal from the experiment in Chapter 3, we obtained a list of tag-pairs with JC

[1] Databases and database drivers were used to handle the application connectivity in later experiments that include larger datasets from multiple domains

scores which were sorted from the highest to the lowest. Table 4.3 show a sample of the tag-pairs and their corresponding JC scores.

MyMediaLite Library - C-sharp

MyMediaLite is a recommender system library for the Common Language Runtime (.NET). According to Gantner et al. (2011), it addresses the two most common scenarios in collaborative filtering: rating prediction (e.g. on a scale of 1 to 5 stars), and item recommendation from positive-only feedback (e.g. from clicks, likes, or purchase actions). Figure 4.8 shows the general overview of interfaces/class structure in MymediaLite. We extended the SVDPlusPlus class in the library under rating prediction in order to create and evaluate our model according to our experimental design.

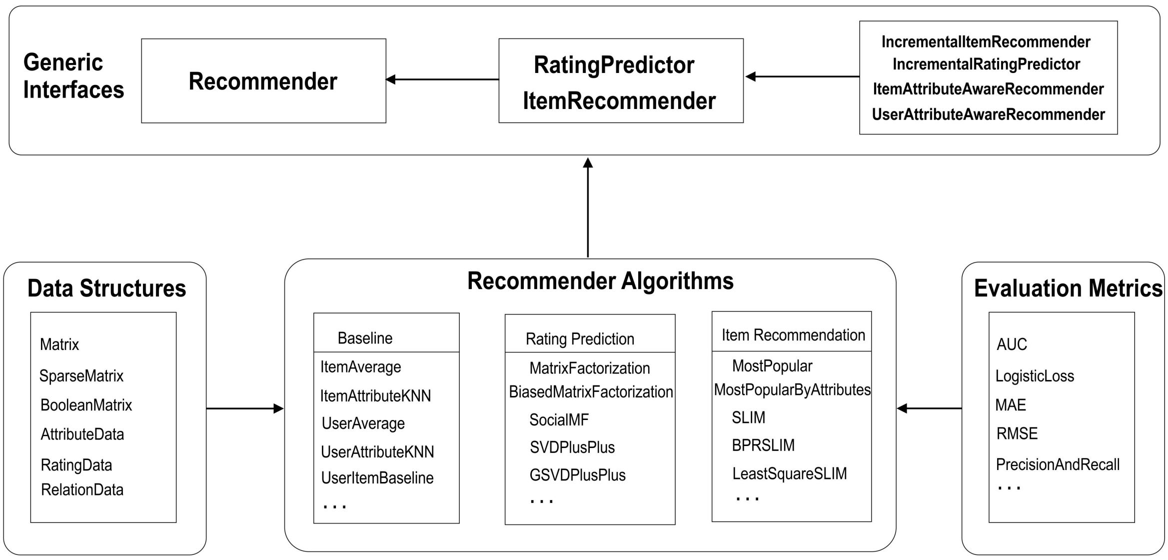


Fig. A.1 General overview of main interfaces and classes in MymediaLite.

The SVDPlusPlus class like the model it represents specifically considers only explicit ratings given by user as implicit feedback when predicting missing values in the test set. We created our model's class by extending the functionality of the SVDPlusPlus class and implementing the interface that allows taking other item and user attributes (ItemAttributeAware/UserAttributeAwareRecommender Interfaces) into account when predicting missing rating values. As detailed in equation 4.10, our model adopt tags —words/texts that users assigns to items— as attributes for the respective user and item in the matrix factorization (SVD) model.

The training phase of our model uses stochastic gradient decent typically utilized in learning the parameters required for predicting missing rating values in a high dimensional rating matrix [Koren et al. (2009), Enrich et al. (2013), Manzato (2013), Shi et al. (2011), Fernández-Tobías and Cantador (2014)]. General inputs essential to SGD algorithm for learning the parameters used by the prediction models are; unique users/items identifiers, ratings values, the learning rate, regularization parameter and a number of latent factors. In cases where the prediction model takes additional inputs to make more accurate predictions, the SGD algorithm will consequently require unique identifies for the attributes.

Additional information about users (user attributes, e.g. gender, age, geographical location, occupation) and items (item attributes, e.g. genres, product categories, keywords) can be added to the latent features of the matrix before the dimensionality reduction process using SVD Gantner et al. (2010). The AttributeData class in the Mymedialite library performs this mapping function for the uniquely identified tag attributes while the RatingData class maps the unique IDs for user/item to the latent features of a matrix.

The matrices of user, item and tag attribute make up the set of parameters to be estimated for predicting rating values. These parameters ($q_i, p_u, x_v, x_s, y_t, y_s$) as shown in algorithm 4.1 are initialized to small values at the start of the SGD iteration. The algorithm then loops through all ratings in the training set and updates the learning parameters until the error function of equation is approximately zero. The final values of these parameters are then used to estimate the ratings of test set.

Our model introduced the parameter x_s, y_s which respectively represent user and item tag attributes that are semantically similar (based on high JC scores) between two domains as part of the parameters to be learnt for estimating the rating values in the test set. The validation sets described in the subsection above were first used to determine optimal number of factors to utilize. The graph of figure 4.9 which shows the number of factor for which the model performs best is generated by plotting the accuracy of the predicted ratings against increasing numbers of common tags. We set the other inputs to our model i.e. learning rate and regularization to the default [2] value used in the SVDPlusPlus model (regularization=0.045, learning rate=0.019). Finally, we investigated the effect of increasing inter-domain tags between the movie domain and the book domain by plotting the rating prediction accuracy of the model over 30 consecutive iterations.

[2] The values of these parameters are optimized in chapter 5 using the gridsearch optimization techniques.

Appendix B

Preliminary Experiment on Semantic Metric Selection

B.1 Evaluating Semantic Measures

As systems that make use of artificial intelligence become more commonplace, there is a corresponding increase in the need to ascertain the type of semantic relationships that exits between operating entities of the system. According to Harispe et al. (2013) semantic measures are widely used today to estimate the strength of the semantic relationship between elements of various types: units of language (e.g., words, sentences, documents), concepts or even instances semantically characterized (e.g., diseases, genes, geographical locations). They have been utilized to compare these elements based on the closeness of the knowledgerepresentations that underpins their meaning or describe their natural sense. Semantic measures are therefore essential for designing intelligent agents such as recommender systems that can take advantage of semantic inferences that are close to the human ability to compare objects and make useful recommendation there off.

Table 3.6 shows the relevant statistics of the metrics and similarity score in each batch. The metric with the highest cumulative AUC over the subset of 2000 similarity scores was taken as the most effective. The charts of figure 3.2 - 3.5 show the specific and total effectiveness of the similarity metrics. We found the Jiang-Conrath measure to have the highest culmulative area under a curve equalled to 4.413 over 8 bins of 2000 highest scoring pairs.

Table B.1 Computational performance of the different Semantic Similarity Metrics

Metric	Scores Returned	Time (min)	Scores $\geq 75th\%$
Leacock-Chodorow	508,923	45	24,084
Resnick	687,459	45	9,944
Lin	508,923	45	5,926
Jiang-conrath	493,461	45	9,308
Hirst-St-Onge	48,294	45	543

Table B.2 Area under the curve for all metrics with ranked similarity score

Score Bin	Thresholds	Type	Leacock-Chod.	Resnik	Lin	Jiang-Con.
		intra	164	170	167	171
1 - 250	250	inter	86	80	83	79
		AUC	0.549	0.546	0.515	0.573
		intra	138	138	143	145
256 - 500	250	inter	112	112	107	105
		AUC	0.513	0.532	0.506	0.535
		intra	156	142	147	147
501 - 750	250	inter	94	108	103	103
		AUC	0.500	0.552	0.484	0.634
		intra	151	169	156	148
751 - 1000	250	inter	99	81	94	102
		AUC	0.500	0.622	0.570	0.477
		intra	137	141	142	135
1001 - 1250	250	inter	113	109	108	115
		AUC	0.545	0.514	0.501	0.559
		intra	131	162	135	155
1251 - 1500	250	inter	119	88	115	95
		AUC	0.514	0.420	0.493	0.557
		intra	155	150	159	155
1501 - 1750	250	inter	95	100	91	95
		AUC	0.500	0.550	0.532	0.563
		intra	153	160	142	153
1751 - 2000	250	inter	97	90	108	97
		AUC	0.500	0.599	0.540	0.515
Cumulative AUC			4.121	4.335	4.141	4.413

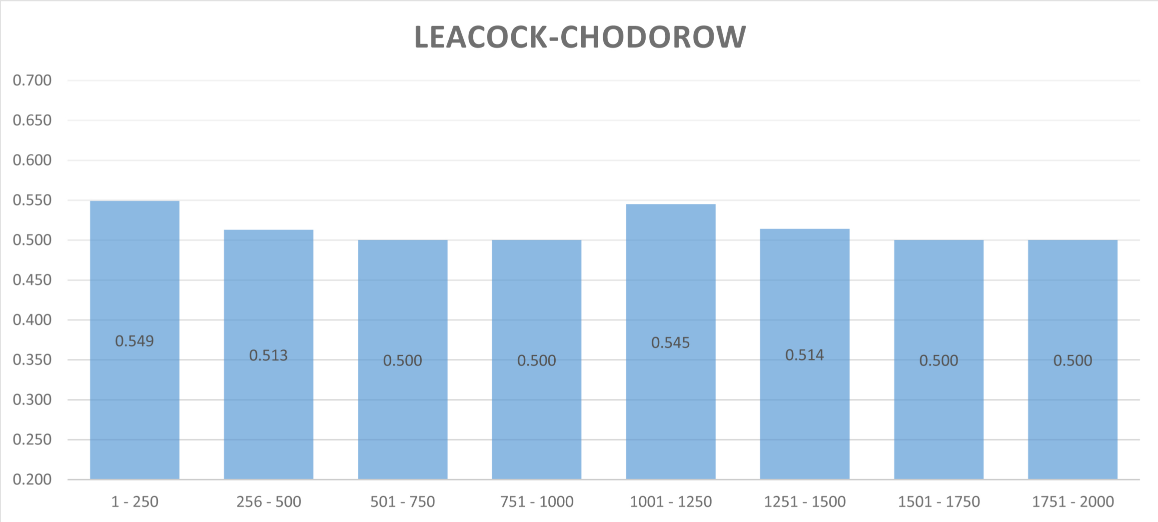


Fig. B.1 Chart showing area under the curve for leacock Chodorow measure

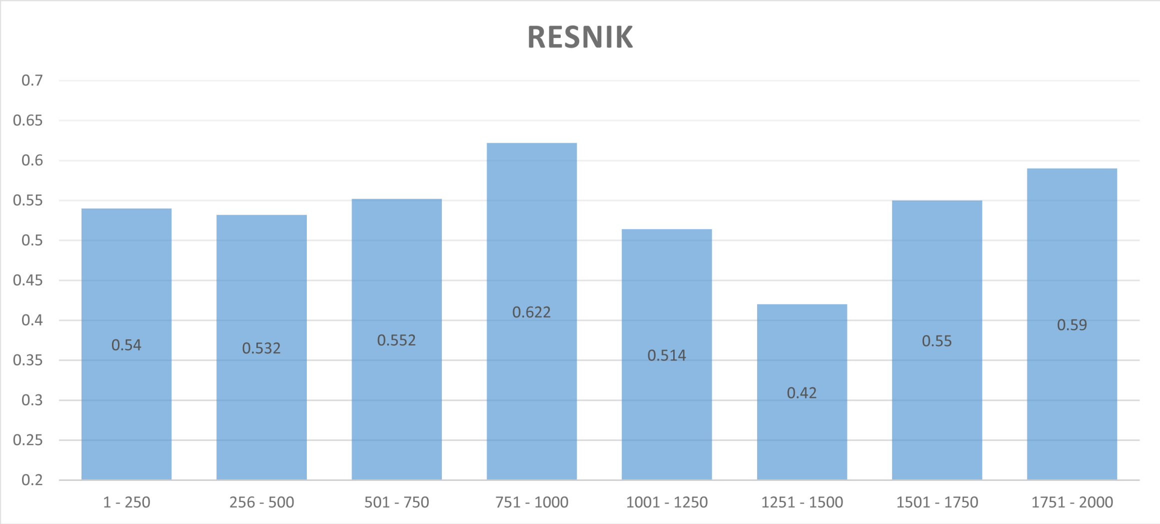


Fig. B.2 Chart showing area under the curve for Resnik measure

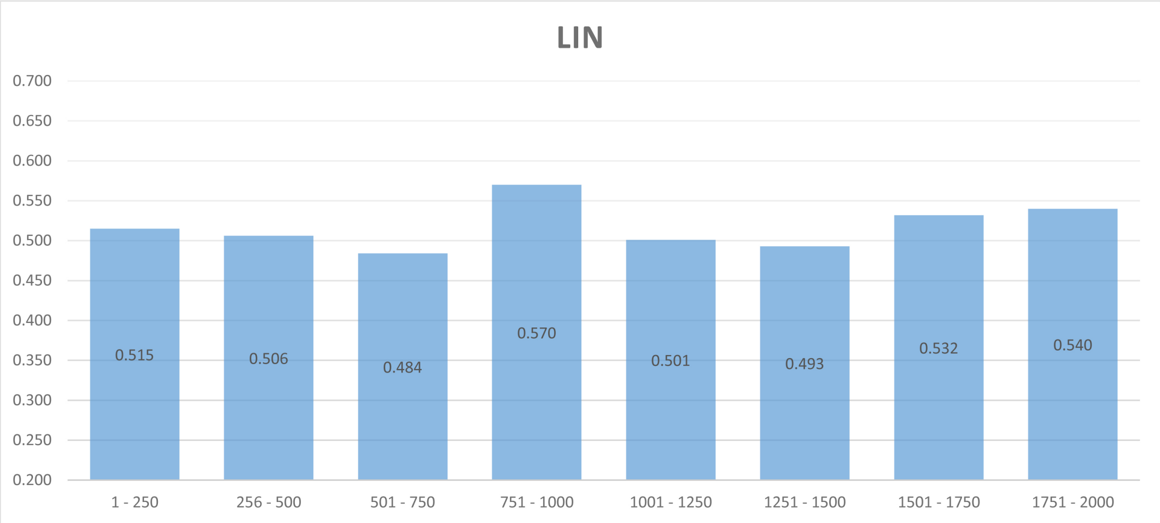


Fig. B.3 Chart showing area under the curve for lin measure

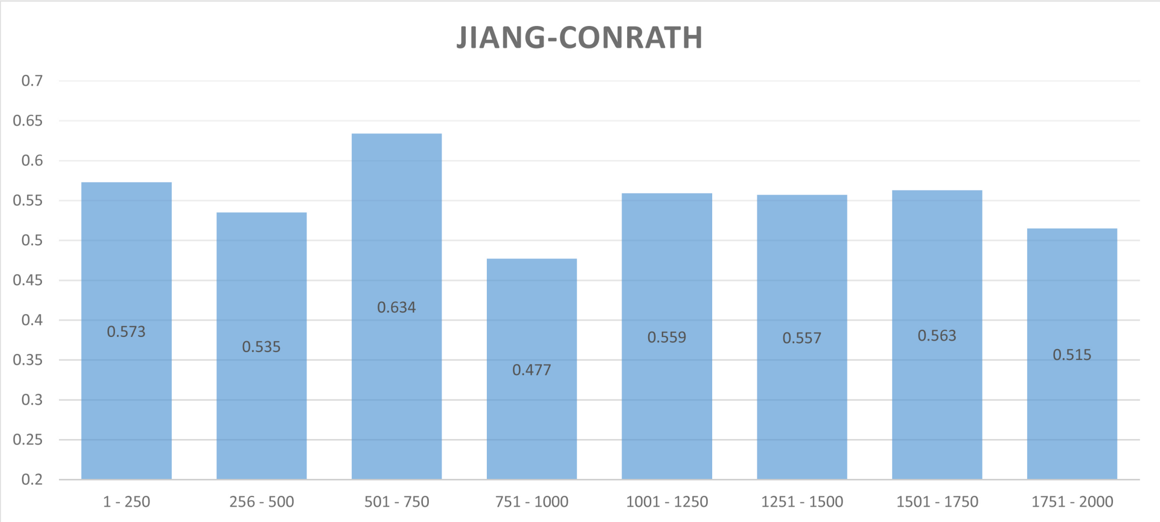


Fig. B.4 Chart showing area under the curve for Jiang-Conrath measure