Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

Incorporating Group Recommendations to Recommender Systems: Alternatives and Performance

Tesis Doctoral

Fernando Ortega Requena

Graduado en Ingeniería del Software | Máster en Inteligencia Artificial

2015

Departamento de Sistemas Informáticos

Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

Incorporating Group Recommendations to Recommender Systems: Alternatives and Performance

Fernando Ortega Requena

Graduado en Ingeniería del Software | Máster en Inteligencia Artificial

Jesus Bobadilla Sancho

Doctor en Infórmatica

Antonio Hernando Estaban

Doctor en Infromática

2015

A todos los que me habeís ayudado a alcanzar este objetivo, sin vosotros no habría sido posible.

TRIBUNAL

Tribunal nombrado por el Mgfco. y Excmo. Sr. Rector de la Universidad Politécnica de Madrid, el día ___ de ____________ de 2014.

Presidente: D. Miguel Ángel Patricio Guisado (Universidad Carlos III de Madrid)

Vocales: D. Pedro Pablo Alarcón Cavero (Universidad Politécnica de Madrid)

D. Guillermo González de Rivera Peces (Universidad Autónoma de Madrid)

Dª. Meritxell Vinyals Salgado (Commissariat à l'énergie atomique et aux énergies alternatives)

Secretario: D. Víctor José Martínez Hernando (Universidad Politécnica de Madrid)

Suplentes: Dª. Ruth Cobos Pérez (Universidad Autónoma de Madrid)

D. Luís Usero Aragonés (Universidad de Alcalá de Henares)

Realizado el acto de lectura y defensa de la Tesis el día ___ de ____________ de 2015 en la Escuela Técnica Superior de Ingeniería de Sistemas Informáticos de la Universidad Politécnica de Madrid.

Calificación: ____

EL PRESIDENTE LOS VOCALES

EL SECRETARIO

ABOUT THIS WORK

This PhD thesis is the result of several years of research in the area of recommender systems. As result of this work, 15 papers indexed in SCI-JCR and 1 publication in a congress has been published. In this PhD thesis I present the results of one of the most promising papers. Group recommendations have started to expand due to the increasing use of social networks. This type of recommendations poses new challenges to researchers, since it adds more complexity to the recommendation problem.

The following list contains all my publications related to recommender systems:

Paper: Ortega, F., Bobadilla, J., Hernando, A., & Rodríguez, F. (2014). Using Hierarchical Graph Maps To Explain Collaborative Filtering Recommendations. *International Journal of Intelligent Systems, 29*(5), 462–477.
Impact factor: 1.411 (Q2)

Paper: Ortega, F., Bobadilla, J., Hernando, A., & Gutierrez, A. (2013). Incorporating Group Recommendations to Recommender Systems: Alternatives and Performance. *Information Processing and Management, 49*(4), 895–901.
Impact factor: 1.069 (Q3) **Citations:** 6

Paper: Bobadilla, J., Hernando, A., Ortega, F., & Gutierrez, A. (2013). Recommender Systems Survey. *Knowledge Based Systems, 46*, 109-132.
Impact factor: 3.058 (Q1) **Citations:** 113

Paper: Ortega, F., Sanchez, J.L., Bobadilla, J., & Gutierrez, A. (2013). Improving Collaborative Filtering based Recommender Systems using Pareto Dominance. *Information Sciences, 239*, 50-61.
Impact factor: 3.893 (Q1) **Citations:** 10

Paper: Hernando, A., Bobadilla, J., Ortega, F., & Gutierrez, A. (2013). Trees for Explaining Recommendations Made through Collaborative Filtering. *Information Sciences, 239*, 1-17.
Impact factor: 3.893 (Q1) **Citations:** 3

Paper: Bobadilla, J., Ortega, F., Hernando, F., & Glez-de-Rivera, G. (2013). A Similarity Metric Designed to Speed Up, Using Hardware, the Recommender Systems k-Nearest Neighbors Algorithm. *Knowledge Based Systems, 51*, 27-34.

Impact factor: 3.058 (Q1) **Citations:** 2

Paper: Hernando, A., Moya, R., Ortega, F., & Bobadilla, J. (2013). Hierarchical Graph Maps for Visualization of Collaborative Recommender Systems. *Journal of Information Sciences, 40*(1), 97-106.

Impact factor: 1.087 (Q2)

Paper: Hernando, A., Bobadilla, J., Ortega, F., & Tejedor, J. (2013). Incorporating Reliability Measurements into the Predictions of a Recommender System. *Information Sciences, 218*, 1-16.

Impact factor: 3.893 (Q1) **Citations:** 10

Paper: Bobadilla, J., Ortega, F., Hernando, A., & Bernal, J., (2012). Generalization of Recommender Systems: Collaborative Filtering Extended to Groups of Users and Restricted to Groups of Items. *Expert Systems with Applications, 39*, 172-186.

Impact factor: 1.854 (Q2) **Citations:** 21

Paper: Bobadilla, J., Ortega, F., & Hernando, A. (2012). A Collaborative Filtering Similarity Measure Based on Singularities. *Information Processing and Management, 48*(2), 204-217.

Impact factor: 0.817 (Q3) **Citations:** 39

Paper: Bobadilla, J., Ortega, F., Hernando, A., & Bernal, J. (2012). A Collaborative Filtering Approach to Mitigate the New User Cold Start Problem. *Knowledge-Based Systems, 26*, 225-238.

Impact factor: 4.104 (Q1) **Citations:** 49

Paper: Bobadilla, J., Hernando, A., Ortega, F., & Gutierrez, A. (2012). Collaborative Filtering Based on Significances. *Information Sciences, 185*(1), 1-17.

Impact factor: 3.643 (Q1) **Citations:** 25

Paper: Bobadilla, J., Ortega, F., Hernando, A., & Arroyo, A. (2012). A Balanced Memory-Based Collaborative Filtering Similarity Measure. *International Journal of Intelligent Systems, 27*, 939-946.

Impact factor: 1.416 (Q2) **Citations:** 3

Paper: Bobadilla, J., Hernando, A., Ortega, F., & Bernal, J. (2011). A Framework for Collaborative Filtering Recommender Systems. *Expert Systems with Applications, 38*(12), 14609-14623.

Impact factor: 1.854 (Q2) **Citations:** 30

Paper: Bobadilla, J., Ortega, F., Hernando, A., & Alcala, J. (2011). Improving Collaborative Filtering Recommender System Results and Performance Using Genetic Algorithms. *Knowledge-Based Systems, 24*(8), 1310-1316.

Impact factor: 2.422 (Q1) **Citations:** 51

Paper: Ortega, F., Hernando, A., & Bobadilla, J. (2011). Extended Precision Quality Measure for Recommender Systems. In *Advances in Artificial Intelligence* (pp. 433-442). Springer Berlin Heidelberg.

RESUMEN

La importancia de los sistemas de recomendación ha experimentado un crecimiento exponencial como consecuencia del auge de las redes sociales. En esta tesis doctoral presentaré una amplia visión sobre el estado del arte de los sistemas de recomendación. Incialmente, estos estaba basados en fitrado demográfico, basado en contendio o colaborativo. En la actualidad, estos sistemas incorporan alguna información social al proceso de recomendación. En el futuro utilizarán información implicita, local y personal proveniente del Internet de las cosas.

Los sistemas de recomendación basados en filtrado colaborativo se pueden modificar con el fin de realizar recomendaciones a grupos de usuarios. Existen trabajos previos que han incluido estas modificaciones en diferentes etapas del algoritmo de filtrado colaborativo: búsqueda de los vecinos, predicción de las votaciones y elección de las recomendaciones. En esta tesis doctoral proporcionaré un nuevo método que realizar el proceso de unficación (pasar de varios usuarios a un grupo) en el primer paso del algoritmo de filtrado colaborativo: cálculo de la métrica de similaridad.

Proporcionaré una formalización completa del método propuesto. Explicaré cómo obtener el conjunto de *k* vecinos del grupo de usuarios y mostraré cómo obtener recomendaciones usando dichos vecinos. Asimismo, incluiré un ejemplo detallando cada paso del método propuesto en un sistema de recomendación compuesto por 8 usuarios y 10 items.

Las principales características del método propuesto son: (a) es más rápido (más eficiente) que las alternativas proporcionadas por otros autores, y (b) es al menos tan exacto y preciso como otras soluciones estudiadas. Para contrastar esta hipótesis realizaré varios experimentos que miden la precisión, la exactitud y el rendimiento del método. Los resultados obtenidos se compararán con los resultados de otras alternativas utilizadas en la recomendación de grupos. Los experimentos se realizarán con las bases de datos de MovieLens y Netflix.

ABSTRACT

The importance of recommender systems has grown exponentially with the advent of social networks. In this PhD thesis I will provide a wide vision about the state of the art of recommender systems. They were initially based on demographic, content-based and collaborative filtering. Currently, these systems incorporate some social information to the recommendation process. In the future, they will use implicit, local and personal information from the Internet of Things.

As we will see here, recommender systems based on collaborative filtering can be used to perform recommendations to group of users. Previous works have made this modification in different stages of the collaborative filtering algorithm: establishing the neighborhood, prediction phase and determination of recommended items. In this PhD thesis I will provide a new method that carry out the unification process (many users to one group) in the first stage of the collaborative filtering algorithm: similarity metric computation.

I will provide a full formalization of the proposed method. I will explain how to obtain the *k* nearest neighbors of the group of users and I will show how to get recommendations using those users. I will also include a running example of a recommender system with 8 users and 10 items detailing all the steps of the method I will present.

The main highlights of the proposed method are: (a) it will be faster (more efficient) that the alternatives provided by other authors, and (b) it will be at least as precise and accurate as other studied solutions. To check this hypothesis I will conduct several experiments measuring the accuracy, the precision and the performance of my method. I will compare these results with the results generated by other methods of group recommendation. The experiments will be carried out using MovieLens and Netflix datasets.

INDEX

1. STATE OF THE ART 1

1.1. INTRODUCTION 1

1.1.1. WHAT RECOMMENDER SYSTEMS ARE? 1

1.1.2. RECOMMENDER SYSTEMS CLASIFICATION 3

1.1.3. RECOMMENDER SYSTEMS TRENDS 7

1.2. RECOMMENDER SYSTEMS 10

1.2.1. CONTENT BASED FILTERING 10

1.2.2. COLLABORATIVE FILTERING 15

1.2.2.1. KNN Algorithm 15

1.2.2.1.1. Similarity Metrics 18

1.2.2.2. Model based CF 20

1.2.2.3. The Cold Start Problem 21

1.2.3. RECOMMENDER SYSTEM EVALUATION 23

1.2.3.1. Evaluation Methods Overview 23

1.2.3.2. Quality of Predictions 27

1.2.3.3. Quality of Recommendations 28

1.2.3.4. Other Quality Measures 28

1.2.3.4.1. Quality of the List of Recommendations: Rank Measures 28

1.2.3.4.2. Novelty and Diversity 29

1.2.3.4.3. Stability 30

1.2.3.4.4. Reliability 30

1.2.4. RECOMMENDATIONS EXPLANATION 31

1.2.5. OTHER TYPE OF RECOMMENDER SYSTEMS 34

1.2.5.1. Geospatial RS 34

1.2.5.2. Social RS 37

1.2.5.3. Bio-inspired RS 38

1.3. GROUP RECOMMENDATIONS 40

2. HYPOTHESIS AND MOTIVATION 43

2.1. INTRODUCTION 43

2.2. MOTIVATION 43

2.3. HYPOTHESIS 45

3. FORMALIZATION 47

3.1. PRELIMINARY DEFINITIONS 47

3.2. OVERVIEW 47

3.3. COMPUTING KNN 48

3.4. PREDICTIONS AND RECOMMENDATIONS 49

3.5. RUNNING EXAMPLE 50

4. EXPERIMENTS 53

4.1. THE RECSYS API 53

4.2. QUALITY MEASURES 54

4.3. EXPERIMENTS OVERVIEW 56

5.1. EXPERIMENT 1: ERRS BASELINE 58

5.2. EXPERIMENT 2: ACCURACY TEST 59

5.3. EXPERIMENT 3: PERFORMANCE TEST 61

6. FUTURE WORK 65

7. CONCLUSIONS 69

1. STATE OF THE ART

This section is organized as follows: section 1.1. contains an introduction about recommender systems, section 1.2. presents the recommender systems' related work, and section 1.3. describes the state of the art of group recommendations.

1.1. Introduction

1.1.1. What Recommender Systems Are?

Recommender Systems (RSs) collect information on the preferences of its users for a set of items (e.g., movies, songs, books, jokes, gadgets, applications, websites, travel destinations and e-learning material). The information can be acquired explicitly (typically by collecting users' ratings) or implicitly (Lee, Cho, & Kim, 2010; Choi, Yoo, Kim, & Suh, 2013; Núñez-Valdés, Cueva-Lovelle, Sanjuán-Martínez, Ordoñez de Pablos, & Monegro Marín, 2012) (typically by monitoring users' behavior, such as songs heard, applications downloaded, web sites visited and books read). RS may use demographic features of users (such as age, nationality or gender). Social information, like followers, followed, tweets, and posts, is commonly used in Web 2.0. There is a growing tend towards the use of information from Internet of Things (e.g., GPS locations, RFID or real-time health signals).

RS make use of different sources of information for providing users with predictions and recommendations of items. They try to balance factors like accuracy, novelty, diversity, serendipity, trust and stability in the recommendations. Collaborative Filtering (CF) methods play an important role in the recommendation, although they are often used along with other filtering techniques like content-based, knowledge-based or social ones.

CF is based on the way in which humans have made decisions throughout history: besides on our own experiences, we also base our decisions on the experiences and knowledge that reach each of us from a relatively large group of acquaintances.

Recently, RS implementation in the Internet has increased, which has facilitated its use in diverse areas (Park, Kim, Choi, & Kim, 2012). The most common research papers are focused on movie recommendation studies (Winoto & Tang, 2010; Carrer-Neto, Hernández-Alcaraz, Valencia-García, & García-Sánchez, 2012); however, a great volume of literature for RS is centered on different topics, such as music (Lee, Cho, & Kim, 2010; Nanolopoulos, Rafailidis, Symeonidis, & Manolopoulos, 2010; Tan, Bu, Chen, & He, 2011), television (Yo, Zhou, Hao, & Gu, 2006; Barragáns-Martínez, Costa-Montenegro, Burguillo, Rey-López, Mikic-Fonte, & Peleteiro, 2010), books (Goldberg, Roeder, Gupta, & Perkins, 2001; Núñez-Valdés, Cueva-Lovelle, Sanjuán-Martínez, Ordoñez de Pablos, & Monegro Marín, 2012), documents (Porcel, Moreno, & Herrera-Viedma, 2009; Porcel & Herrera-Viedma, 2010; Serrano-Guerrero, Herrera-Viedma, Olivas, Cerezo, & Romero, 2011; Porcel, Tejada-Lorente, Martínez, & Herrera-Viedma, 2012), e-learning (Zaiane, 2002; Bobadilla, Serradilla, & Hernando, 2009), e-commerce (Huang, Zeng, & Chen, 2007; Castro-Schez, Miguel, Vallejo, & López-López, 2011), applications in markets (Costa-Montenegro, Barragáns-Martínez, & Rey-López, 2012), tourism (Borràs, Moreno, & Valls, 2014) and web search (McNally, O'mahony, Coyle, Briggs, & Smyth, 2011; Zhang, Yu, Fang, You, Liu, Liu, & Yan, 2014), among others.

The kinds of filtering most used at the beginning of the RS (collaborative, content-based and demographic) were described in Pazzani (1999). Breese, Heckerman, and Kadie (1998) evaluated the predictive accuracy of different algorithms for CF; later, Herlocker, Kostan, Terveen, and Riedl (2004) describe the base for evaluating the collaborative filtering RS.

The evolution of RS has shown the importance of hybrid techniques of RS, which merge different techniques in order to get the advantages of each of them. A survey focused on the hybrid RS has been presented by Burke (2002). However, it does not deal with the role of social filtering, a technique which has become more popular in the recent years through social networks.

The neighborhood-based CF has been the recommendation method most popular at the beginning of the RS; Herlocker, Konstan, and Riedl (2002) provide a set of guidelines for designing neighborhood-based prediction systems. Adomavicius and Tuzhilin (2005) present an overview on the RS field standing out the most complex areas on which researchers in RS should focus in the "next generation of RS": limited content analysis and overspecialization in content-based methods, cold-start and sparsity in CF methods, model-based techniques, non-intrusiveness, flexibility (real-time customization), etc.

The process for generating a RS recommendation is based on a combination of the following considerations:

- The type of data available in its database (e.g., ratings, user registration information, features or content for items that can be ranked, social relationships among users or location-aware information).
- The filtering algorithm used (e.g., demographic, content-based, collaborative, social-based, context-aware or hybrid).
- The model chosen (e.g., based on direct use of data: “memory-based”, or a model generated using such data: “model-based”).
- The techniques used: probabilistic approaches, bayesian networks, nearest neighbors algorithm, bio-inspired algorithms such as neural networks and genetic algorithms, fuzzy models, singular value decomposition techniques, etc.
- Sparsity level of the database and the desired scalability.
- Performance of the system: time and memory consuming.
- The objective sought is considered (e.g., predictions or top N recommendations) as well as the desired quality of the results (e.g., novelty, coverage or precision).

1.1.2. Recommender Systems Clasification

The internal functions for RS are characterized by the filtering algorithm. The most widely used classification divides the filtering algorithms into (Adomavicius & Tuzhilin, 2005; Candillier, Meyer, & Boullé, 2007; Schafer, Frankowski, Herlocker, & Sen, 2007): (a) collaborative filtering, (b) demographic filtering, (c) content-based filtering and (d) hybrid filtering.

Content-based filtering (Lang, 1995; Van Meteren & Van Someren, 2000; Salter & Antonopoulus, 2006) makes recommendations based on user choices made in the past (e.g. in a web-based e-commerce RS, if the user purchased some fiction films in the past, the RS will probably recommend a recent fiction film that he has not yet purchased on this website). Content-based filtering also generates recommendations using the content from objects intended for recommendation; therefore, certain content can be analyzed, like text, images and sound. From this analysis, a similarity can be established between objects as the basis for recommending items similar to items that a user has bought, visited, heard, viewed and ranked positively.

Demographic filtering (Krulwich, 1997; Pazzani, 1999; Porcel, Tejeda-Lorente, Martínez, & Herrera-Viedma, 2012) is justified on the principle that individuals with certain common personal attributes (sex, age, country, etc.) will also have common preferences.

Collaborative filtering (Herlocker, Konstan, Bochers, & Riedl, 1999; Herlocker, Konstan, Terveen, & Riedl, 2004; Adomavicius & Tuzhilin, 2005; Candillier, Meyer, & Boullé, 2007; Su & Khoshgoftaar, 2009) allows users to give ratings about a set of elements (e.g. videos, songs, films, etc. in a CF based website) in such a way that when enough information is stored on the system, we can make recommendations to each user based on information provided by those users we consider to have the most in common with them. CF is an interesting open research field (Xie, Han, Yang, Shen, Zeng, & Chen, 2007; Bobadilla, Hernando, Ortega, & Bernal, 2011; Bobadilla, Hernando, Ortega, & Gutiérrez, 2012). As noted earlier, user ratings can also be implicitly acquired (e.g., number of times a song is heard, information consulted and access to a resource).

The most widely used algorithm for collaborative filtering is the *k* Nearest Neighbors (kNN) (Adomavicius & Tuzhilin, 2005; Schafer, Frankowski, Herlocker, & Sen, 2007; Bobadilla, Hernando, Ortega, & Bernal, 2011). In the user to user version, kNN executes the following three tasks to generate recommendations for an active user: (1) determine *k* users neighbors for the active user *a*; (2) aggregate the ratings of the *k* neighbors to make a prediction of the items not rated by *a*; and (3) select the best prediction values to obtain the top *N* recommendations.

Hybrid filtering (Burke, 2002; Porcel, Tejeda-Lorente, Martínez, & Herrera-Videma, 2012) commonly uses a combination of CF with demographic filtering (Vozalis & Margaritis, 2007; Son, 2014) or CF with content-based filtering (Barrangans-Martínez, Costa-Montenegro, Burguillo, Rey-López, Mikic-Fonte, & Peleteiro, 2010; Choi, Yoo, Kim, & Suh, 2012; Zhang, Peng, Su, & Liu, 2014) to exploit merits of each one of these techniques. Hybrid filtering is usually based on bio-inspired or probabilistic methods such as genetic algorithms (Ho, Fong, & Yan, 2007; Gao & Li, 2008), fuzzy genetic (Al-Shamri & Bharadwaj, 2008), neural networks (Lee, Choi, & Woo, 2002; Christakou& Stafylopatis, 2005; Ren, He, Gu, Xia, & Wu, 2008), bayesian networks (Campos, Fernández-Luna, Huete, & Rueda-Morales, 2010), clustering (Shinde & Kulkami, 2012) and latent features (Maneeroj & Takasu, 2009).

A widely accepted taxonomy divides recommendation methods into memory-based and model-based method categories:

Memory-based methods (Adomavicius & Tuzhilin, 2005; Kong, Sun, & Ye, 2005; Candilier, Meyer, & Boullé, 2007; Symeonidis, Nanopoulos, & Manolopoulos, 2009) can be defined as collaborative filtering methods that (a) act only on the matrix of user ratings for items and (b) use any rating generated before the referral process (i.e., its results are always updated). Memory-based methods usually use similarity metrics to obtain the distance between two users, or two items, based on their ratings.

Model-based methods (Adomavicius & Tuzhilin, 2005; Su & Khoshgoftaar, 2009) use RS information to create a model that generates the recommendations. Among the most widely used models we have bayesian classifiers (Park, Hong, & Cho, 2007), bio-inspired networks (Roh, Oh, & Han, 2003; Merve-Acilar & Arslan, 2009), fuzzy systems (Yager, 2003; Nilashi, Ibrahim, & Ithnin, 2014), genetic algorithms (Ho, Fong, & Yan, 2007; Gao & Li, 2008), clustering methods (Wu, Chang, & Liu, 2014), latent features (Zhong & Li, 2010), matrix factorization (Lou, Xia, & Zhu, 2012), and probabilistic models (Mnih & Salakhutdinov, 2007; Wang & Blei, 2011), among others.

The graph in **Figure 1.1** shows the most significant traditional methods, techniques and algorithms for the recommendation process as well as their relationships and groupings. Different sections of this state of the art provide more detail on the most important aspects involved in the recommendation process.

As may be seen in **Figure 1.1**, we can use some of the traditional filtering methods (content-based, demographic and collaborative) applied to databases. Model-based technologies (genetic algorithms, neural networks, etc.) make also use of this information to build the model. Typical memory-based approaches are: item to item, user to user, and hybrids of the two previous. The main purpose of both memory-based and model-based approaches is to get the most accurate predictions in the tastes of users. The accuracy of these predictions may be evaluated through the classical information retrieval measures, like MAE, precision, and recall. Researchers make use of these measures in order to improve the RS methods and technologies.

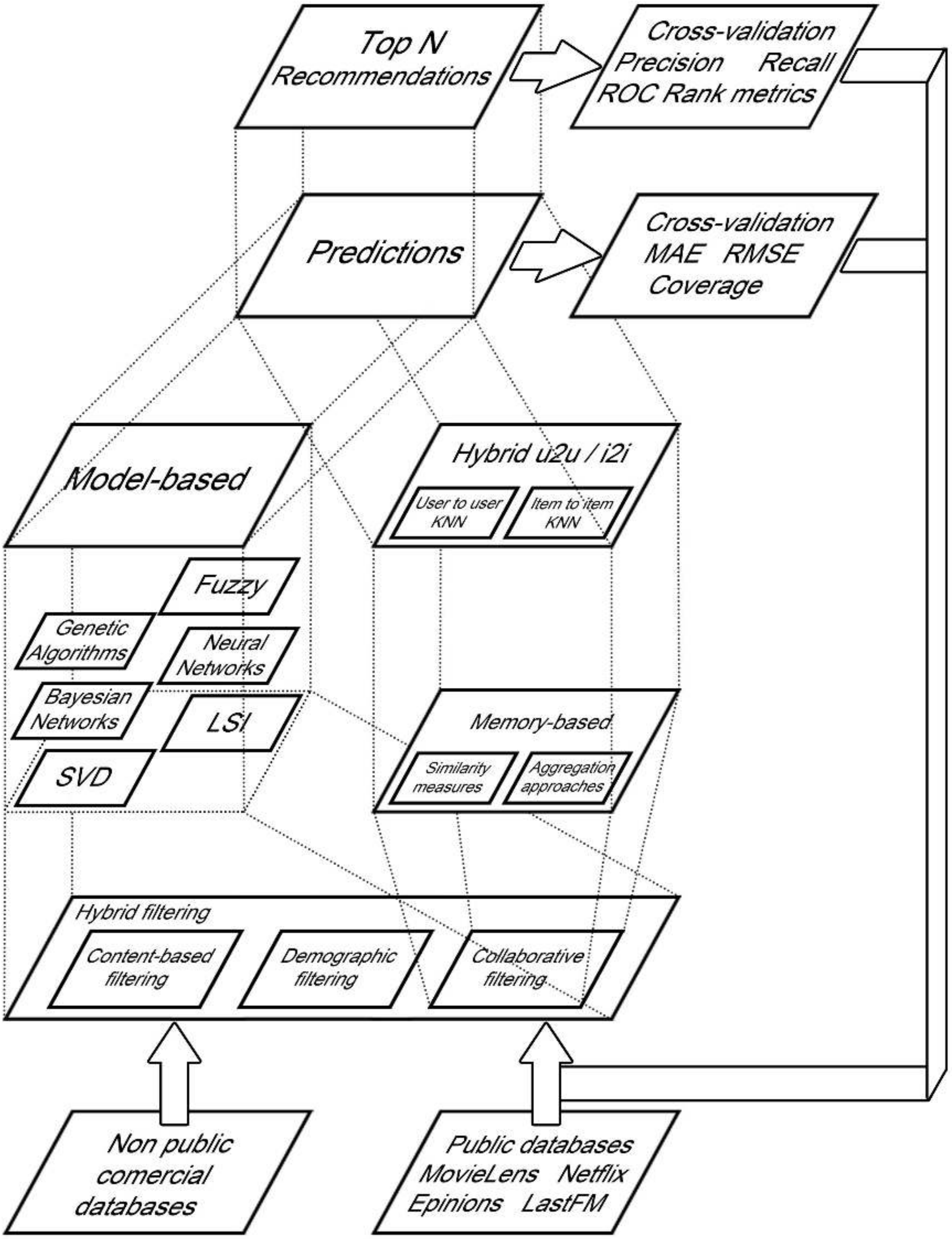


Figure 1.1. Recommender systems clasification.

1.1.3. Recommender Systems Trends

From the evolution of existing RS and research papers in the field, there is a clear tendency to collect and integrate more and different types of data. This trend is parallel to the evolution of the web, which we can define through the following three primary stages. (1) At the genesis of the web, RS used only the explicit ratings from users as well as their demographic information and content-based information included by the RS owners. (2) For the web 2.0, in addition to the above information, RS collect and use social information, such as friends, followers, followed, both trusted and untrusted. Simultaneously, users aid in the collaborative inclusion of such information: blogs, tags, comments, photos and videos. (3) For the web 3.0 and the Internet of Things, context-aware information from a variety of devices and sensors will be incorporated with the above information. Currently, geographic information is included, and the expected trend is gradual incorporation of information, such as radio frequency identification (RFID) data, surveillance data, on-line health parameters, and food and shopping habits, as well as teleoperation and telepresence.

There is a clear trend towards collection of implicit information instead of a traditional explicit evaluation of items by ratings. Last.fm is a good example of this situation; the user ratings are inferred by the number of times they have heard each song. The same can be applied in a number of everyday situations, such as for access to web addresses, use of various public transport systems, food purchased, access to sports facilities or access to learning resources.

Incorporation of implicit information on the daily habits of users allows RS to use a wide variety of data; these data will be used in future CF processes, which are increasingly useful and accurate. Privacy and security considerations will be increasingly important with the widespread trend in using, with consent, devices and sensors for the Internet of Things.

Privacy is an important issue for RS (Bilge & Polat, 2012) because the systems contain information on large numbers of registered users. For privacy preservation in RS, a certain level of uncertainty must be introduced into the predictions (McSherry & Mironov, 2009), primarily through tradeoffs between accuracy and privacy (Machanavajjhala, Korolova, & Sarma, 2011). Furthermore, privacy can be preserved when different RS companies share information (combining their data) (Zhan, Hsieh, Wang, Hsu, Liau, & Wang, 2010; Kaleli & Polat, 2012). Privacy becomes more important as RS increasingly incorporate social information.

Because RS are often used in electronic commerce, unscrupulous producers may find profitable to shill RS by lying to the systems in order to have their products recommended more often than those of their competitors. RS can experience shilling attacks (Lam & Riedl, 2004; Chirita, Nejdl, & Zamfir, 2005), which generate many positive ratings for a product, while products from competitors receive negative ratings. RS are still highly vulnerable to such attacks (Ray & Mahanti, 2009).

Knowledge-based filtering is emerging as an important field of RS. Knowledge RSs (Burke, 2000a) use knowledge about users and products to pursue a knowledge-based approach to generating recommendations, reasoning about what products meet the user's requeriments. Recommendations are based on inferences about users needs and preferences. User models are based on knowledge structures such as querys (preferred features for products) (Jannach, 2009), cases (case-based reasoning) (Bridge, Goker, McGinty, & Smyth, 2005), constraints (constraint-based reasoning) (Felferning & Burke, 2008), ontologies (Middleton, Shadbolt, & De Roure, 2004), matching metrics and knowledge vectors (Rodrigues, Tomaz, De Souza, & Xexéo, 2012), and social knowledge (Carrer-Leto, Hernández-Alcaraz, Valencia-García, & García-Sánchez, 2012).

Workflow is a current knowledge field where the user model is based on users-roles-tasks reference information that describes which member plays which roles or fulfills which tasks (Zhen, Huang, & Jiang, 2009a; Zhen, Huang, & Jiang, 2009b). Peer-to-peer (P2P) networks are other current knowledge field, where user information is based on the distributed information existing from each peer and the set of peers who may need her (Zhen, Jiang, & Song, 2010).

Gradual incorporation of different types of information (e.g., explicit ratings, social relations, user contents, locations, use trends or knowledge-based information) has forced RS to use hybrid approaches. Once the memory-based, social and location-aware methods and algorithms are consolidated, the evolution of RS demonstrates a clear trend toward combining existing collaborative methods.

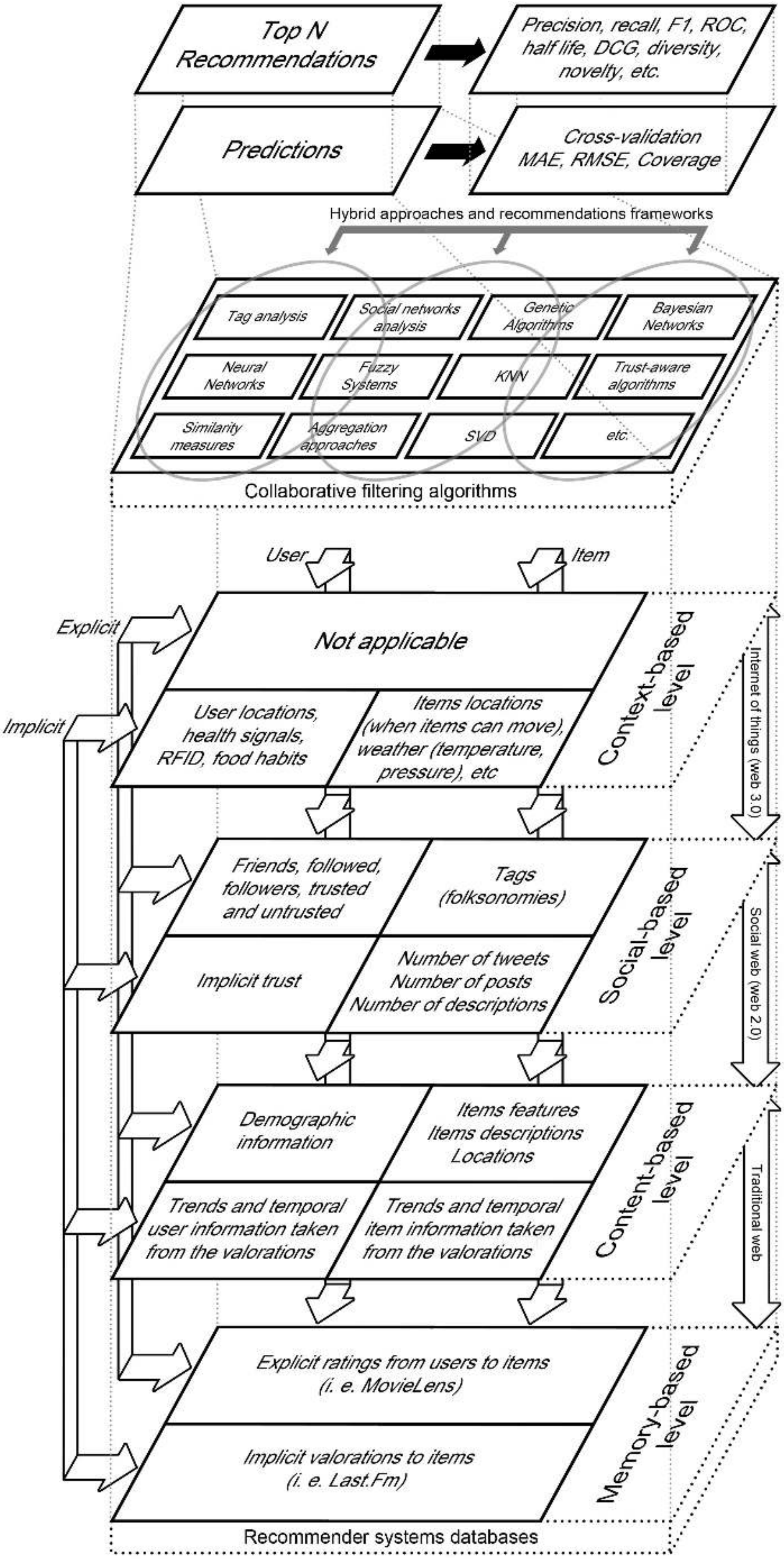

Top N Recommendations
Precision, recall, F1, ROC, half life, DCG, diversity, novelty, etc.
Predictions
Cross-validation MAE, RMSE, Coverage
Hybrid approaches and recommendations frameworks
Tag analysis
Social networks analysis
Genetic Algorithms
Bayesian Networks
Neural Networks
Fuzzy Systems
KNN
Trust-aware algorithms
Similarity measures
Aggregation approaches
SVD
etc.
Collaborative filtering algorithms
User
Item
Explicit
Implicit
Not applicable
User locations, health signals, RFID, food habits
Items locations (when items can move), weather (temperature, pressure), etc
Context-based level
Internet of things (web 3.0)
Friends, followed, followers, trusted and untrusted
Tags (folksonomies)
Implicit trust
Number of tweets Number of posts Number of descriptions
Social-based level
Social web (web 2.0)
Demographic information
Items features Items descriptions Locations
Trends and temporal user information taken from the valorations
Trends and temporal item information taken from the valorations
Content-based level
Traditional web
Explicit ratings from users to items (i. e. MovieLens)
Implicit valorations to items (i. e. Last.Fm)
Memory-based level
Recommender systems databases

Figure 1.2. Recommender systems trends classified by data sources.

The latest research in the CF field has generated only modest improvements for predictions and recommendations from a single type of information (e.g., when the only information used is user ratings, information from social relations, or item content). The results improve further when several algorithms are combined with their respective data types. A growing number of publications address hybrid approaches that use current databases to simultaneously incorporate memory-based, social and content-based information.

To unify the above concepts, **Figure 1.2** provides an original taxonomy for RS. The taxonomy is classified depending on the nature of the data rather than according to the methods and algorithms used. The core of the taxonomy focuses on data classification by three factors: (1) the target of the data: user or item; (2) mode of acquisition: explicit (i.e., ratings to items made by users) or implicit (e.g., number of times a user has heard a song); and (3) information level: memory, content, social or context-based level.

Figure 1.2 shows the recommender methods and algorithms (labeled as "collaborative filtering algorithms"). Depending on the information type in each RS database, it adopts a hybrid filtering approach. Each hybrid approach will use an appropriate subset of algorithms to consider processing of existing information in a coordinated manner. Future developments will include different recommendation frameworks that address the most common situations. These frameworks allow RS to incorporate the CF kernel with the most appropriate recommendations methods based on the available information in a simple and straightforward manner.

At higher levels (prediction and recommendation), **Figure 1.2** incorporates current evaluation quality measures, such as those for diversity and novelty. The importance of such measures, and measures developed in the future will grow as users demand novel, stable and less predictable recommendations.

1.2. Recommender Systems

1.2.1. Content based Filtering

Content-Based Filtering (CBF) is becoming especially important as RS incorporate information on items from users working in web 2.0 environments. The current trend in CBF is to add social information to the items attributes, such as tags, comments, opinion, and social network sharing.

Two challenging problems for content-based filtering are limited content analysis and overspecialization (Adomavicius & Tuzhilin, 2005). Limited content analysis problem arises from the difficulty in extracting reliable automated information from various content (e.g., images, video, audio or text), which can greatly reduce the quality of recommendations. Overspecialization problem refers to the phenomenon in which users only receive recommendations for items that are very similar to items they liked or preferred; therefore, the users are not receiving recommendations for items that they might like but are unknown (e.g., when an user only receives recommendations about fiction films). Recommendations can be evaluated for novelty (Bobadilla, Hernando, Ortega, & Bernal, 2011; Hurley& Zhang, 2011).

The key factor in CBF is the extraction of the item attributes that you wish to recommend (Pazzani & Billsus, 2007). Typically, a set of attributes is manually defined for each item depending on its domain. In certain instances, such as when it is desired to recommend textual information, classic information retrieval techniques must be used to automatically define such attributes (e.g., term frequency, inverse document frequency and normalization to page length).

The main advantage of CBF lies in the fact that this algorithms can work in environments where the ratings of the users are either stored or not. If the system contains information about ratings the recommendations should be based on the idea that items with similar attributes will be similarly rated (Balabanovic & Shoham, 1997; Pazzani, 1999; Schafer, Frankowski, Herlocker, & Sen, 2007) and items similar to those rated positively in the past can be recommended to the active user. If the system does not contain information about users, items similar to the active one can be used to perform recommendations (e.g.: if you like this movie, you will probably like this one).

LSI (Deerwester, Dumais, Landauer, Furnas, & Harshman, 1990), PLSA (Hofmann, 1999) and LDA (Blei, Ng, & Jordan, 2003) are well known methods to obtain similar items recommendations based on description of the items. These methods analyze the description of the items trying to find the hidden topics latent in the text. Once the hidden topics of the description of the items are identified, they can be classified using their hidden topics (Griffiths & Steyvers, 2004).

When the attributes of the items and the user profiles are known, the key purpose for CBF (Van Meteren & Van Someren, 2000) is to determine whether a user will like a specific item. This task is resolved traditionally by using heuristic methods (Salton, 1989; Baeza-Yates & Ribeiro-Neto, 1999; García & Amatriain, 2010) or classification algorithms, such us: rule induction (Cohen, 1995; Kim, Lee, Shaw, Chang, & Nelson, 2001), nearest neighbors methods (Yang, 1999; Billsus, Pazzani, & Chen, 2002), Rocchio's algorithm (Lang, 1995; Balabanovic & Shoham, 1997), linear classifiers (Jeong, Lee, & Cho, 2009), and probabilistic methods (Pazzani & Billsus, 1997; Mooney & Roy, 2000; Gemmis, Lops, Semeraro, & Basile, 2008; Chulyadyo & Leray, 2014).

The pure CBF has several shortcomings (Balabanovic & Shoham, 1997; Pazzani & Billsus, 2007; Su & Khoshgoftaar, 2009):

(a) In certain domains (e.g., music, blogs, and videos), it is a complicated task to generate the attributes of the items.

(b) CBF suffers from an overspecialization problem because by nature it tends to recommend the same types of items.

(c) It is more difficult to acquire feedback from users in CBF because, users do not typically rate the items (like in CF), and, therefore, it is not possible to determine whether the recommendation is correct or not.

Because of these shortcomings, it is not usual to find a pure CBF implementation. It is more common to use the hybrid CBF/CF (Burke, 2002). CF solves CBF's problems because it works in any domain; it is less affected by overspecialization; and it acquires feedback from users. CBF adds the following qualities to CF: improvement to the quality of the predictions, because they are calculated with more information, and reduced impact from the cold-start and sparsity problems.

CBF and CF can be combined in different ways (Adomavicius, & Tuzhilin, 2005). **Figure 1.3** shows the main different alternatives.

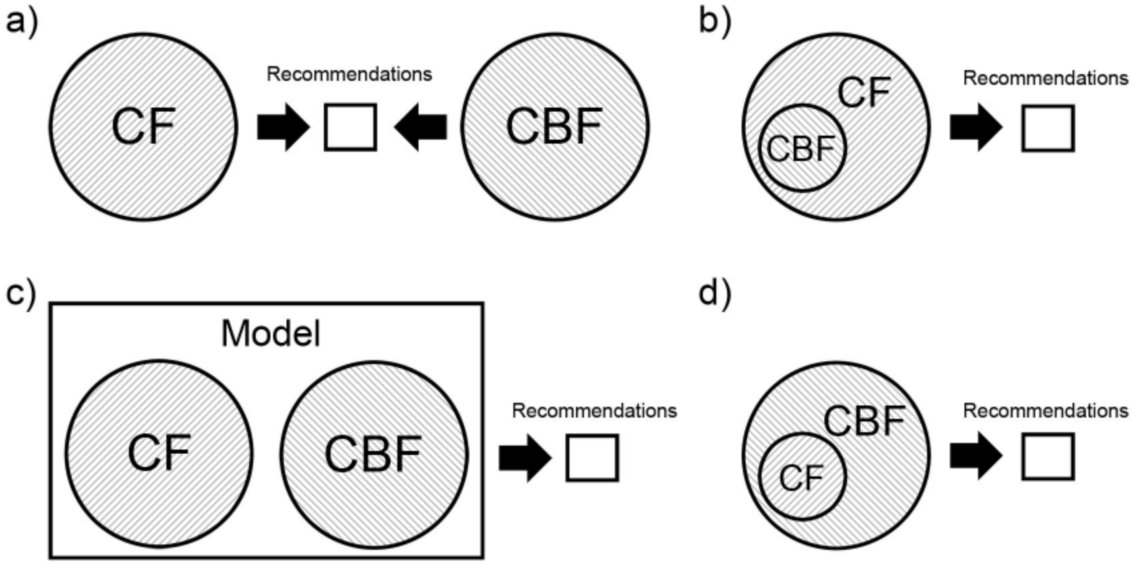


Figure 1.3. Different alternatives for combining Content Based Filtering (CBF) and Collaborative Filtering (CF).

Figure 1.3 a) shows the methods that calculate CBF and CF recommendations separately and subsequently combine them. Claypool, Gokhale, Miranda, Murnikov, Netes, and Sartin (1999) propose to use a weighted average for combining CBF and CF predictions depending on the type of prediction. In another study, Pazzani (1999) proposes combining the CBF and CF recommendation lists by assigning the items scores according to their position on the lists. Additionally, Billsus and Pazzani (2000) and Tran and Cohen (2000) propose to select the CBF or CF prediction based on its quality.

Figure 1.3 b) depicts the methods that incorporate CBF characteristics into the CF approach. Balabanovic and Shoham (1997) maintain user profiles based on content analysis and directly compare the profiles to determine similar users for CF recommendations. Good, Schafer, Konstan, Borchers, Sarwar, Herlocker, and Riedl (1999) construct specialized filterbots using CBF techniques, which later act as neighbors in the CF stage. Melville, Mooney, and Nagarajan (2002) propose to add predictions from the CBF into the ratting matrix employed by the CF. Li and Kim (2003) modifies the ratting matrix, which is input for the CF, by combining it with another matrix generated from clustering the items according to their attributes. In Hu and Pu (2010), authors incorporate personality characteristics in the CF similarity measure to minimize the new-user problem.

Figure 1.3 c) illustrates the methods to design a unified model based on both CBF and CF techniques. Basu, Hirsh, & Cohen (1998) propose using CBF and CF characteristics in a single rule-based classifier. Popescul, Ungar, Pennock, and Lawrence (2001) and Schein, Popescul, Ungar, and Pennock (2002) propose using probability models to combine CBF and CF recommendations. In another studies (Condliff, Lewis, Madigan, & Posse, 1999; Ansari, Essegaier, & Kohil, 2000; Campos, Fernández-Luna, Huete, & Rueda-Morales, 2010), the authors employ bayesian networks to combine CBF and CF characteristics and generate more accurate recommendations. Burke (2000b) and Middleton, Shadbolt, and De Roure (2004) use knowledge-based techniques to solve the cold start problem.

Figure 1.3 d) shows the methods that incorporate CF characteristics into a CBF approach. In Soboroff and Nicholas (1999), the authors use LSI to create the user profiles used in CBF recommendations beginning with the CF ratting matrix. Mooney and Roy (2000) use CF system predictions as input for CBF.

Web 2.0 environments have increased the amount of content information of the items. Social tagging systems are the most popular because they allow users to annotate online resources with arbitrary labels, which produces rich information spaces (folksonomies). These new components have opened novel lines of RS research that can be divided into two categories: (a) tag recommendation systems and (b) use of tags in the recommendation process:

(1) RS tags attempt to provide personalized item recommendations to users through the most representative tags. In Jächke, Marinho, Hotho, Schmidt-Thieme, & Stumme (2007), the authors compare different mechanisms for tags recommendations. Marinho and Schmidt-Thieme (2008) improve tags recommendations by applying classic recommendation methods. Additionally, Landia and Anand (2009) propose a method that combines clusteringbased CBF with CF to suggest new tags to users.

(2) The methods using tags in the recommendation process increase the capacity of traditional RS. Tso-Sutter, Marinho, and Schmidt-Thieme (2008) propose a generic method that allows tags to be incorporated to standard CF algorithms. Bogers and Van Den Bosh (2009) examine how to incorporate the tags and other metadata into a hybrid CBF/CF algorithm by replacing the traditional user-based and item-based similarity measures by tag overlap. Gemmell, Schimoler,

Ramezani, Christiansen, and Mobasher (2009) propose a weighted hybrid recommender, wherein they combine the graph-based tag recommendations with user-based CF and item-based CF. Gedikli and Jannach (2010) propose to use tags as a means to express which features of an item users particularly like or dislike. In Gemmell, Schimoler, Mobasher, & Burke (2010), the authors offer a hybrid RS, wherein they predict the user preferences for items by only consulting the user's tagging history.

1.2.2. Collaborative Filtering

1.2.2.1. KNN Algorithm

The k Nearest Neighbors (kNN) recommendation algorithm is the reference algorithm for the CF recommendation process. Its primary virtues are simplicity and reasonably accurate results; its major pitfalls are low scalability and vulnerability to sparsity in the RS databases. This section provides a general explanation of this algorithm function.

CF based on the kNN algorithm is conceptually simple, with a straightforward implementation; it also generally produces good-quality predictions and recommendations. However, due to the high level of sparsity (Bobadilla & Serradilla, 2009; Luo, Xia, & Zhu, 2012) in RS databases, similarity measures often have problems to be calculated (typically from insufficient mutual ratings for a comparison of users and items) in cold-start situations (users and/or items with low number of rankings) (Schein, Popescul, Ungar, & Pennock, 2002; Leung, Chan, & Chung, 2008; Kim, El-Saddik, & Jo, 2011; Bobadilla, Ortega, Hernando, & Bernal, 2012a).

Another major problem for the kNN algorithm is its low scalability (Luo, Xia, & Zhu, 2012). As the databases (such as Netflix) increase in size (hundreds of thousands of users, tens of thousands of items, and hundreds of millions of rankings), the process for generating a neighborhood for an active user becomes too slow, because the similarity measure must be processed as often as new users are registered in the database. The item to item version of the kNN algorithm significantly reduces the scalability problem (Sarwar, Katypis, Konstan, & Riedl, 2001). To this end, neighbors are calculated for each item; their top *N* similarity values are stored, and for a period of time, predictions and recommendations are generated using the stored information.

Although the stored information does not include the ratings from previous processing/storage, outdated information for items is less sensitive than for the users.

A recurrent theme in CF research is generating metrics to calculate with accuracy and precision the existing similarity for the users (or items). Traditionally, a series of statistical metrics have been used (Adomavicius & Tuzhilin, 2005; Candillier, Meyer, & Boullé, 2007), such as the Pearson correlation, cosine, constraint Pearson correlation and mean squared differences. Recently, metrics have been designed to fit the constraints and peculiarities of RS (Bobadilla, Serradilla, & Bernal, 2010; Bobadilla, Hernando, & Ortega, 2012). The relevance (significance) concept was introduced to afford more importance to more relevant users and items (Wang, De Vries, & Reinders, 2008; Bobadilla, Hernando, Ortega, & Gutiérrez, 2012). Additionally, a group of metrics was specifically designed to correctly work in cold-start situations (Ahn, 2008; Bobadilla, Ortega, Hernando, & Bernal, 2012a).

The kNN algorithm is based on similarity measures. The similarity approaches typically compute the similarity between two users *x* and *y* (user to user) based on both users' item ratings. The item to item kNN version computes the similarity between two items *i* and *j*.

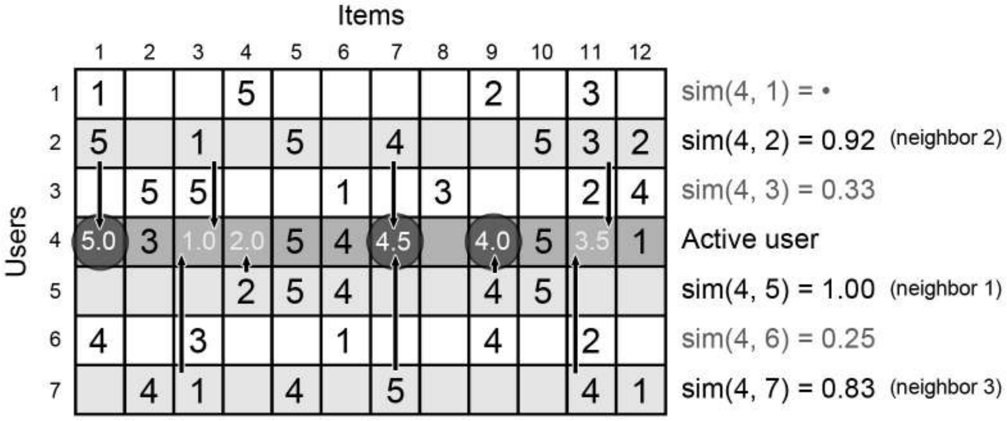
A formal approach of the kNN algorithm may be found in (Bobadilla, Hernando, Ortega, & Bernal, 2011). The method for making recommendations is based on the following three steps:

(1) Using the similarity measure, we obtain the set of *k* neighbors for the active user *a*. The *k* neighbors for *a* are the *k* most similar users of *a*.

(2) After the selection of the set of *k* most similar users (neighbors), an aggergation function should be used in order to obtain the prediction of item *i* on user *a*. The following aggregation approaches are often used: the average, the weighted sum and the adjusted weighted aggregation (deviation-from-mean).

(3) Selecting the items with highest value in the predictions for the active user in order to obtain the top-n recommendations.

Figure 1.4 shows an example of the application of kNN algorithm. Step (1) can be seen on the right side of the figure (similarity computation and selection of the most similar users). Step (2) can be seen with the black vertical arrows inside the matrix: these arrows represent the aggregation of the kNN's ratings. Step (3) can be seen with the dark grey circles inside the matrix: the top-3 higher predictions (white numbers) have been selected to be recommended.



	1	2	3	4	5	6	7	8	9	10	11	12	
1	1			5					2		3		sim(4, 1) = •
2	5		1		5		4			5	3	2	sim(4, 2) = 0.92 (neighbor 2)
3		5	5			1		3			2	4	sim(4, 3) = 0.33
4	5.0	3	1.0	2.0	5	4	4.5		4.0	5	3.5	1	Active user
5				2	5	4			4	5			sim(4, 5) = 1.00 (neighbor 1)
6	4		3			1			4		2		sim(4, 6) = 0.25
7		4	1		4		5				4	1	sim(4, 7) = 0.83 (neighbor 3)

Figure 1.4. User to user collaborative filtering process example.

In the item to item version (Sarwar, Karypis, Konstan, & Riedl, 2001; Gao, Wu, & Jiang, 2011; Pereira, Lopes, Breitman, Mundim, & Peixoto, 2014) of the kNN algorithm, the following three tasks are performed:

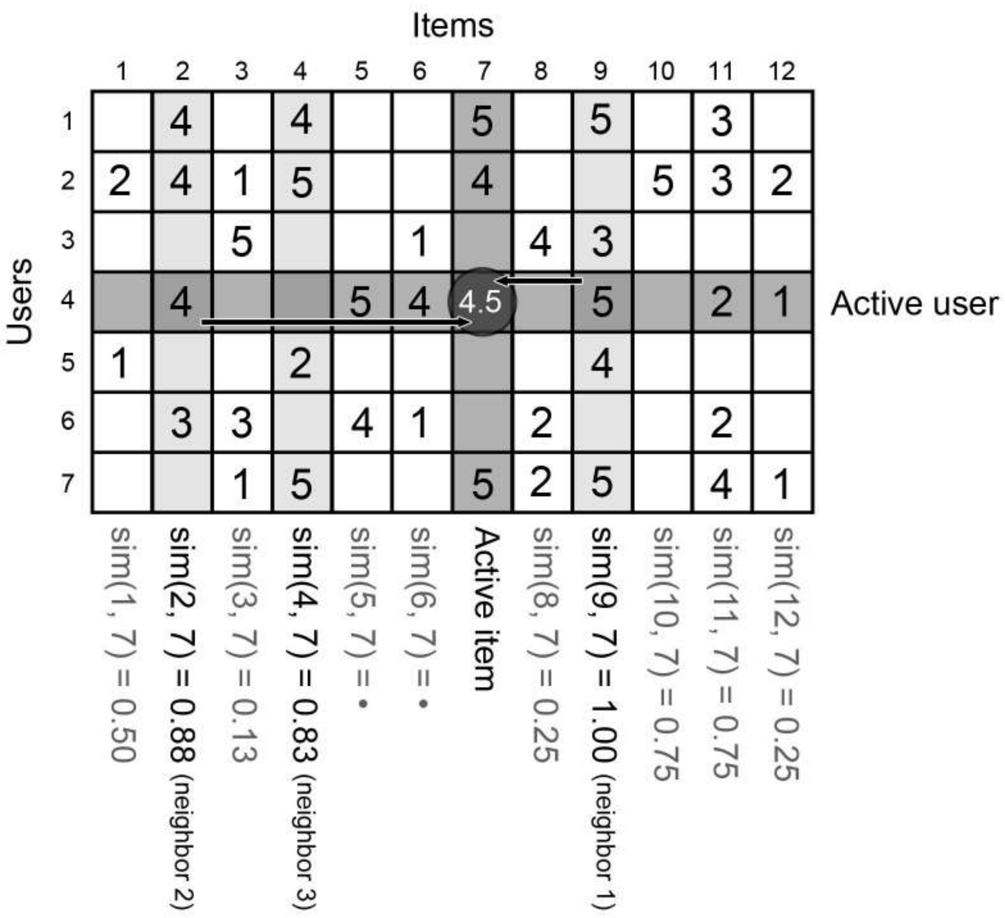
(1) Determine *k* items neighbors for each item in the database. This can be done periodically in order to improve the efficiency of the user to user version.

(2) For each item *i* not rated by the active user *a*, calculate its prediction based on the ratings of *a* and the *k* neighbors of *i*.

(3) Select the top-*n* recommendations for the active user (typically the *n* major predictions from *a*).

Figure 1.5 shows how to compute a prediction using item to item kNN algorithm. On the bottom of the figure the similarity values and the most similar items of the active item are shown. The black horizontal arrows inside the matrix represent the aggregation phase: the ratings of the kNN are combined to compute the prediction. If we compute the predictions of all items unrated by the active user we can obtain the set of top-*n* recommendations.

The item to item and user to user versions of the kNN algorithm can be combined (Qin, Xin, & Liang, 2011) to take advantage of the positive aspects from each approach.



	1	2	3	4	5	6	7	8	9	10	11	12
1		4		4			5		5		3	
2	2	4	1	5			4			5	3	2
3			5			1		4	3			
4		4			5	4	4.5		5		2	1
5	1			2					4			
6		3	3		4	1		2			2	
7			1	5			5	2	5		4	1

Figure 1.5. Item to item collaborative filtering process example.

1.2.2.1.1. Similarity Metrics

A Similarity Metric (SM) determines the similarity between pairs of users (user to user CF) or the similarity between pairs of items (item to item CF). For this purpose, it compares the ratings of all the items rated by two users (user to user) or the ratings of all users who have rated two items (item to item).

The kNN algorithm is based essentially on the use of traditional similarity metrics. These metrics require only the set of ratings made by the users (memory-based CF). Among the most commonly used traditional metrics we have: Pearson correlation, cosine, adjusted cosine, constrained correlation, Mean Squared Differences (MSD) and euclidean distance (Adomavicius & Tuzhilin, 2005; Candillier, Meyer, & Boullé, 2007).

A new metric (JMSD) has recently been published (Bobadilla, Serradilla, & Bernal, 2010), which uses the numerical information of the ratings (via mean squared differences) as well as the non-numerical information provided by the arrangement of them (via Jaccard). Ortega, Sánchez, Bobadilla, & Gutiérrez (2013) use Pareto dominance to perform a pre-filtering process eliminating less representative users from the k-neighbors selection process while retaining the most promising ones. An specialization of the memory-based CF SM (Bobadilla, Ortega, & Hernando, 2012) uses the information contained in the ratings of all users, instead of considering only the ratings of two users (user to user) or the two items (item to item). Currently, two new interesting similarity measures have obtained much more coverage (Bobadilla, Ortega, Hernando, & Arroyo, 2013) and accuracy (Choi & Suh, 2013).

The computation of smilarity measure is the most expensive step in the CF algorithm. Some authors have designed hardware-based methods to speed it up. Bobadilla, Ortega, Hernando, and Glez-de-Rivera (2013) propose an implementation of the SM that can be run in a specific hardware. Wang, Liu, and Ma (2014) uses CUDA to parallelize the execution of the CF algorithm increasing up to 48x the performance of the system.

Different papers have designed similarity measures (Jeong, Lee, & Cho, 2009; Kwon, Cho, & Park, 2009) which aim to extract information related to trust and reputation by only using the users' set of ratings (memory-based CF). The advantage is that their use can be generalized to all CF RS; the drawback of this approach lies in the fact that the social information obtained is really poor.

There are so far research papers dealing with the cold-start problem through the users' ratings information. Ahn (2008) presents PIP, a heuristic similarity measure that outperforms the traditional statistical SM (Pearson correlation, cosine, etc.). Kim, El-Saddik, and Jo (2011) propose a method that firstly predicts actual ratings and subsequently identifies prediction errors for each user. Taking into account this error information, some specific "error-reflected" models, are designed. Bobadilla, Ortega, Hernando, and Bernal (2012a) presents a metric based on neural learning (model-based CF) and adapted for new user cold-start situations.

Bobadilla, Hernando, Ortega, and Gutiérrez (2013) compare the quality and trust of predictions and the quality and novelty of the recommendations for different similarity measure. They also classified the experiment as standard systems and cold-start systems using the MovieLens and Netflix public datasets.

1.2.2.2. Model based CF

Another interesting approach different from kNN algorithm, is to learn a model (model-based CF) to predict how users rate. This approach usually involves a significant improvement in the accuracy of the predictions as well as that they can be calculated much faster. Main drawback of this approach is that the model must be regularly learnt in order to consider the last set of ratings introduced by users.

One popular trend on model based CF is to learn a model, which will be used along with memory based CF similarity metrics. These metrics could not be regarded as strictly memory-based CF because they require a model to work. Bobadilla, Ortega, Hernando, and Alcalá (2011) provide a metric based on a model generated using genetic algorithms. As a result of the increase in web 2.0 websites on the Internet, different researchers (Ekström, Björnsson, & Nass, 2005; Li & Kao, 2009; Yuan, Guan, Lee, Lee, & Hur, 2010) have proposed a set of metrics based on social information (friends, followers, followers, etc.) in order to learn the models.

A wide variety of models has been recently proposed to improve the performance of the CF RS: Merve-Acilar and Arslan (2009) use artificial immune networks to build a model that reduces the sparsity and increase scalability; Nilashi, Ibrahim, and Ithnin (2014) combines a fuzzy similarity metric with ANFIS model to improve recomendations quality on multi-criteria RS; Dao, Jeong, and Ahn (2012) presents a model based on genetic algorithms applied to context-aware recommender systems; Park, Hong, and Cho (2007) propose a map-based personalized recommendation system which reflects user's preference modeled by bayesian networks.

In order to reduce the problems from high levels of sparsity in RS databases, some studies have proposed models based on dimensionality reduction techniques (Sarwar, Karypis, Konstan, & Riedl, 2000b). The reduction methods are based on matrix factorization (Koren, Bell, & Volinsky, 2009; Lou, Xia, & Zhu, 2012; Lou, Xia, & Zhu, 2013). Methods based on matrix factorization are especially suitable for processing large RS databases and providing scalable approaches (Takács, Pilászy, Németh, & Tikk, 2009). These methods methods provide good prediction results but are computationally very expensive to learn, and therefore they can only be used in static off-line settings where the known preference information does not change with time.

RS can use clustering techniques to improve the prediction quality and reduce the cold-start problem when applied to hybrid filtering. It is typical to form clusters of items in hybrid RS (Yao & Zhang, 2009; Shinde & Kulkami, 2012). A different common approach uses clustering both for items and users (bi-clustering) (George & Meregu, 2005; Zhu& Gong, 2009). RS comprising social information have been clustered to improve the following areas: tagging (Shepitsen, Gemmell, Mobasher, & Burke, 2008), explicit social links (Pham, Cao, Klamma, & Jarke, 2011) and explicit trust information (Dobois, Golbeck, Kleint, & Srinivasan, 2009; Pitsilis, Zhang, & Wang, 2011). Wu and Chang (2014) propose an approach based on models that combine content-based with collaborative filtering by means of co-clustering with augmented matrices (CCAM).

The current model based CF techniques are focused on probabilistic models. These models try to find the hidden (latent) topic distribution that represents each user and each item of the system (e.g. movies rating are usually grouped by movie genres: adventures, action, comedy, etc.). Some authors (Mnih & Salakhutdinov, 2007; Koren, Bell, & Volinsky, 2009) try to find these topics factoring the matrix where the ratings are stored. The CTR model (Wang & Blei, 2011) adds additional information (item description) to the inference process. Kang & Lerman (2013a) improve the CTR model including social and psychological factors.

1.2.2.3. The Cold Start Problem

The cold-start problem (Adomavicius & Tuzhilin, 2005; Schafer, Frankowski, Herlocker, & Sen, 2007) appears when it is not possible to make reliable recommendations due to an initial lack of ratings. We can distinguish three kinds of cold-start problems: new community, new item and new user. The last kind is the most important in RS that are already in operation.

The new community problem (Schein, Popescul, Ungar, & Pennock, 2002; Lam, Vu, Le, & Duong, 2008) refers to the difficulty, when starting up a RS, in obtaining, a sufficient amount of data (ratings) for making reliable recommendations. Two common ways are used for tackling this problem: to encourage users to make ratings through different means (e.g.: ratings tours, explain the benefits of the ratings, or implicit rating systems) and combine CF-based recommendations with other types of RS until enough users and ratings exists.

The new item problem (Park & Tuzhilin, 2008; Park & Chu, 2009) arises because the new items entered in RS do not usually have initial ratings, and therefore, they are not likely to be recommended. In turn, an item that is not recommended keeps unnoticed by a large part of the community of users, and as they are unaware of it, they do not rate it; this involves vicious circle in which a set of items of the RS are left out of the ratings/recommendations process. The new item problem has less importance on RS where the items can be discovered via other means (e.g. movies). A common solution is to always recommend some novel items to increase the rating probability over these items.

The new user problem (Ryan & Bridge, 2006; Rashid, Katypis, & Riedl, 2008) represents one of the great difficulties faced by the RS in operation. Since new users in the RS have not yet provided any rating in the RS, they cannot receive any personalized recommendations based on memory-based CF; when the users enter their firsts ratings they expect the RS to offer them personalized recommendations, but the number of ratings introduced in the RS is usually not yet sufficient to be able to make reliable CF-based recommendations, and, therefore, new users may feel that the RS does not offer the service they expected and they may stop using it. The common strategy to tackle the new user problem consists of turning to additional information to the set of ratings in order to be able to make recommendations based on the data available for each user.

The cold-start problem is often faced using hybrid approaches (usually CF-content based RS, CF-demographic based RS, and CF-social based RS) (Loh, Lorenzi, Granada, Lichtnow, Wives, & Oliveira, 2009; Kim, Ji, Ha, & Jo, 2010). Leung, Chan, and, Chung (2008) propose a novel content-based hybrid approach that makes use of cross-level association rules to integrate content information about domains items. Kim, Ji, Ha, and Jo (2010) use collaborative tagging employed as an approach in order to grasp and filter users' preferences for items and they explore the advantages of the collaborative tagging for data sparseness and cold-start users (they collected the dataset by crawling the collaborative tagging delicious site). Weng, Xu, Li, and Nayak (2008) combine the implicit relations between users' items preferences and the additional taxonomic preferences to make better quality recommendations as well as alleviate the cold-start problem. Loh, Lorenzi, Granada, Lichtnow, Wives, and Oliveira (2009) represent user's profiles with information extracted from their scientific publications. Martinez, Perez, and Barranco (2009) present a hybrid RS which combines a CF algorithm with a knowledge-based one. Chen and He (2009) propose a *number of common terms / term frequency* (NCT/TF) CF algorithm based on demographic vector. Maneeroj and Takasu (2009) propose a hybrid RS that utilizes latent features extracted from items represented by a multi-attributed record using a probabilistic model. Park, Pennock, Madani, Good, and Coste (2006) propose a new approach: they use filterbots, and surrogate users that rate items based only on user or item attributes.

1.2.3. Recommender System Evaluation

1.2.3.1. Evaluation Methods Overview

Research in RS requires using a representative set of public databases to facilitate investigations on the techniques, methods and algorithms developed by researchers in the field. Through these databases, the scientific community can replicate experiments to validate and improve their techniques. **Table 1.1** lists the current public databases referenced most often in the literature. Last.Fm and Delicious incorporate implicit ratings and social information; their data were generated from the versions released in the HetRec, 2011 data sets, hosted by the GroupLens research Group.

	Database	Ratings	Users	Items	Range	Tags	Tags assigns	Friend relations	Items
Without social information	**MovieLens 1M**	1 million	6,040	3,592	[1..5]	N/A	N/A	N/A	Movies
	MovieLens 10M	10 million	71,567	10,681	[1..5]	N/A	N/A	N/A	Movies
	Netflix	100 million	480,189	17,770	[1..5]	N/A	N/A	N/A	Movies
	Jester	4.1 million	73,421	100	[-10,10]	N/A	N/A	N/A	Jokes
	EachMovie	2.8 million	72,916	1,628	[0,1]	N/A	N/A	N/A	Movies
	BookCrossing	1.1 million	278,858	271,379	[1..10]	N/A	N/A	N/A	Books
With social information	**MovieLens**	855,898	2,113	10,153	[1..5]	13,222	47,957	N/A	Movies
	Lasf.fm	92,834	4,892	17,632	Implicit	11,946	186,479	25,434	Music
	Delicious	104,833	1,867	69,226	Impicit	53,388	437,593	15,328	Webs

Table 1.1. Most often used memory-based recommender systems public databases.

Since RS research began, evaluation of predictions and recommendations has become important (Sarwar, Karypis, Konstan, & Riedl, 2000a; Herlocker, Konstan, Terveen, & Riedl, 2004). Research in the RS field requires quality measures and evaluation metrics (Gunawardana & Shani, 2009) to know the quality of the techniques, methods, and algorithms for predictions and recommendations. Evaluation metrics (Herlocker, Konstan, Terveen, & Riedl, 2004; Hernández & Gaudioso, 2008) and evaluation frameworks (Herlocker, Konstan, Borchers, & Riedl, 1999; Bobadilla, Hernando, Ortega, & Bernal, 2011) facilitate comparisons of several solutions for the same problem and selection from different promising lines of research that generate better results.

Because of evaluation measures, RS recommendations have gradually been tested and improved (Cacheda, Carneiro, Fernández, & Formoso, 2011). A representative set of existing evaluation measures has standard formulations, and a group of open RS public databases has been generated. These two advances have facilitated quality comparisons for new proposed recommendation methods and previously published methods; thus, RS methods and algorithms research has progressed continuously.

The most commonly used evalutaion metrics (Antunes, Herskovic, Ochoa, & Pino, 2012) can be classified as (Herlocker, Konstan, Terveen, & Riedl, 2004; Hernández & Gaudioso, 2008; Gunawardana & Shani, 2009):

(a) *Prediction metrics*: such as the accuracy ones: Mean Absolute Error (MAE), Root of Mean Square Error (RMSE), and Normalized Mean Average Error (NMAE); and the coverage.

(b) *Recommendation set metrics*: such as precision, recall and Receiver Operating Characteristic (ROC) (Schein, Popescul, Ungar, & Pennock, 2002).

(c) *Rank recommendation metrics*: such as half-life (Breese, Heckerman, & Kadie, 1998) and discounted cumulative gain (Baltrunas, Makcinskas, & Ricci, 2010).

(d) *Diversity metrics*: such as the diversity and the novelty of the recommended items (Hurley & Zhang, 2011).

Hernández and Gaudioso (2008) propose an evaluation process based on the distinction between interactive and non-interactive subsystems. General publications and reviews also exist, which include the most commonly accepted evaluation measures: mean absolute error, coverage, precision, recall and derivatives of these: mean squared error, normalized mean absolute error, ROC and fallout. Goldberg, Roeder, Gupta, and Perkins (2001) focus on the aspects not related to the evaluation. Breese, Heckerman, and Kadie (1998) compare the predictive accuracy of various methods in a set of representative problem domains.

The majority of articles discuss attempted improvements to the accuracy of RS results (RMSE, MAE, etc.). It is also common to attempt an improvement in recommendations (precision, recall, ROC, etc.). However, additional objectives should be considered for generating greater user satisfaction (Ziegler, McNee, Konstan, & Lausen, 2005), such as topic diversification and coverage serendipity.

Currently, the field has a growing interest in generating algorithms with diverse and innovative recommendations, even at the expense of accuracy and precision. To evaluate these aspects, various metrics have been proposed to measure recommendation novelty and diversity (Hurley & Zhang, 2011; Vargas& Castells, 2011).

The frameworks aid in defining and standardizing the methods and algorithms employed by RS as well as the mechanisms to evaluate the quality of the results. Some authors have proposed different CF frameworks. Herlocker, Konstan, Borchers, and Riedl (1999) evaluate the following: similarity weight, significance weighting, variance weighting, selecting neighborhood and rating normalization. Hernández and Gaudioso (2008) propose a framework in which any RS is formed by two different subsystems, one of them to guide the user and the other to provide useful/interesting items. Koutrika, Bercovitz, and Garcia (2009) define a framework that introduces levels of abstraction in CF process, making the modifications in the RS more flexible. Antunes, Herskovic, Ochoa, and Pino (2012) presents an evaluation framework assuming that evaluation is an evolving process during the system lifecicle.

Most RS evaluation frameworks proposed so far present two deficiencies:

- Lack of formalization. Although the evaluation metrics are well defined, there are a variety of details about the methods' implementation, which can lead to different results in the same experiments if they are not correctly specified.
- Lack of standardization of the evaluation measures in aspects such as novelty and trust of the recommendations.

Bobadilla, Hernando, Ortega, and Bernal (2011) provide a complete series of mathematical formalizations based on sets theory. Authors provide a set of evaluation measures, which include the quality analysis of the following aspects: predictions, recommendations, novelty and trust.

The validation process is performed by ussing the most common cross validation techniques (random sub-sampling and k-fold cross validation) (Bengio & Grandvalet, 2004). For cold-start situations, due to the limited number of ratings, the usual method chosen to carry out the experiments is leave-one-out cross validation (Bobadilla, Ortega, Hernando, & Bernal, 2012a).

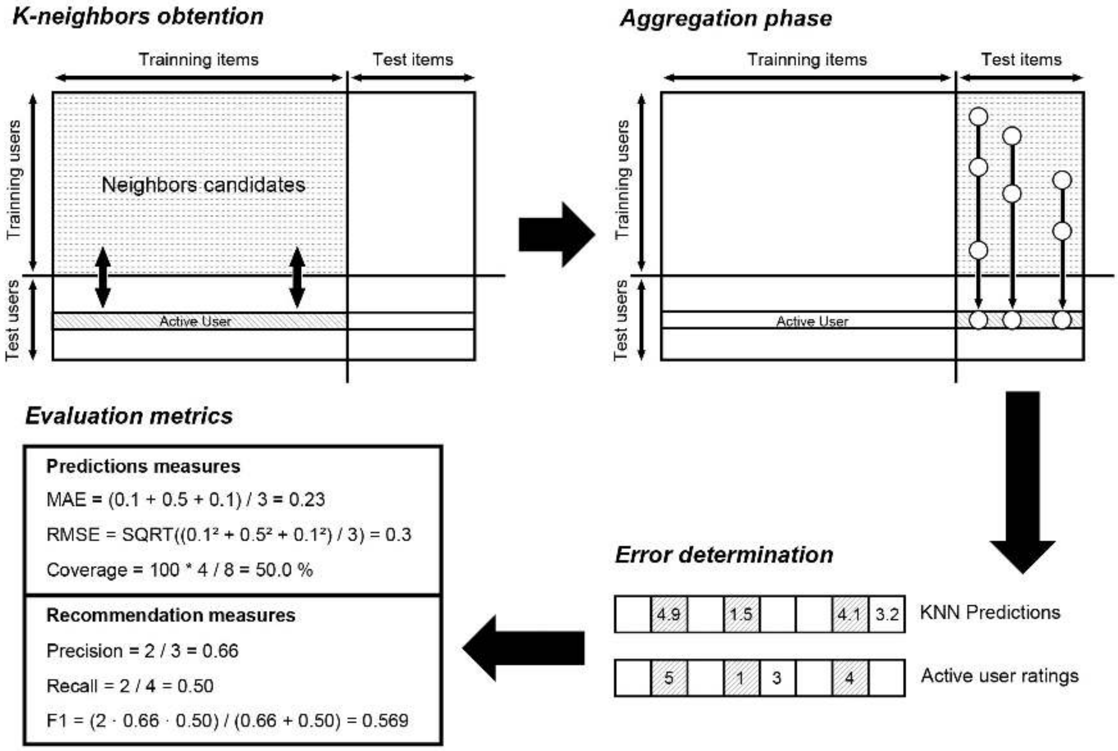


Figure 1.6. Collaborative filtering evaluation process.

Figure 1.6 shows the general mechanism for cross validation used to generate quality results form the evaluation measures. The database is divided in training and test areas for both users and items. In the first phase (top on the left side), k-neighbors are calculated for the active user (while the active user is selected from the set of test users, the k-neighbors are selected from the set of training users). In the aggregation phase (top on the right side), predictions are calculated for the active user (from the set of test items). Finally, evaluation metrics are used to compare the predictions and recommendations obtained with the real ratings of the user; the more accurate the predictions and recommendations, better quality of the proposed recommendation algorithm.

1.2.3.2. Quality of Predictions

In order to measure the accuracy of the results of an RS, different measures of prediction error are used like the Mean Absolute Error (MAE) and its related metrics: mean squared error, root mean squared error, and normalized mean absolute error stand out.

We define *U* as the set of the RS users, *I* as the set of the RS items, $r_{u,i}$ the rating of user *u* on item *i*, • the lack of rating ($r_{u,i} = \bullet$ means user *u* has not rated item *i*), $p_{u,i}$ the prediction of item *i* on user *u*.

Let $O_u = \left\{ i \in I \mid p_{u,i} \neq \bullet \wedge r_{u,i} \neq \bullet \right\}$, set of items rated by user *u* having prediction values. We define the MAE (1.1) and RMSE (1.2) of the system as the average of the user's MAE. We remark that the absolute difference between prediction and real value, $\left| p_{u,i} - r_{u,i} \right|$, informs about the error in the prediction.

$$MAE = \frac{1}{\#U} \sum_{u \in U} \left(\frac{1}{\#O_u} \sum_{i \in O_u} \left| p_{u,i} - r_{u,i} \right| \right) \tag{1.1}$$

$$RMSE = \frac{1}{\#U} \sum_{u \in U} \sqrt{ \frac{1}{\#O_u} \sum_{i \in O_u} \left(p_{u,i} - r_{u,i} \right)^2 } \tag{1.2}$$

The coverage could be defined as the capacity of predicting from a metric applied to a specific RS. In short, it calculates the percentage of situations in which at least one k-neighbor of each active user can rate an item that has not been rated yet by that active user. We defined $K_{u,i}$ as the set of neighbors of *u* which have rated the item *i*. We define the coverage of the system as the average of the user's coverage:

$$C_u = \left\{ i \in I \mid r_{u,i} = \bullet \wedge K_{u,i} \neq \varnothing \right\} \tag{1.3}$$

$$D_u = \left\{ i \in I \mid r_{u,i} = \bullet \right\} \tag{1.4}$$

$$coverage = \frac{1}{\#U} \sum_{u \in U} \left(100 \times \frac{\#C_u}{\#D_u} \right) \tag{1.5}$$

1.2.3.3. Quality of Recommendations

The confidence of users for a certain RS does not depend directly on the accuracy for the set of possible predictions. A user gains confidence on the RS when this user agrees with a reduced set of recommendations made by the RS.

In this section, we define the following three most widely used recommendation quality measures:

(a) *Precision*: which indicates the proportion of relevant recommended items from the total number of recommended items.

(b) *Recall*: which indicates the proportion of relevant recommended items from the number of relevant items.

(c) *F1*: which is a combination of precision and recall.

Let X_u as the set of recommendations to user *u*, and Z_u as the set of *n* recommendations to user *u*. We will represent the evaluation precision, recall and F1 measures for recommendations obtained by making *n* test recommendations to the user *u*, taking a θ relevancy threshold. Assuming that all users accept *n* test recommendations:

$$precision = \frac{1}{\#U}\sum_{u\in U}\frac{\#\left\{i \in Z_u \mid r_{u,i} \geq \theta\right\}}{n} \tag{1.6}$$

$$recall = \frac{1}{\#U}\sum_{u\in U}\frac{\#\left\{i \in Z_u \mid r_{u,i} \geq \theta\right\}}{\#\left\{i \in Z_u \mid r_{u,i} \geq \theta\right\} + \#\left\{i \in Z_u^C \mid r_{u,i} \geq \theta\right\}} \tag{1.7}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{1.8}$$

1.2.3.4. Other Quality Measures

1.2.3.4.1. Quality of the List of Recommendations: Rank Measures

When the number *n* of recommended items is not small, users give greater importance to the first items on the list of recommendations. The mistakes made in these items are more serious errors than those in the last items on the list. The ranking measures consider this situation. Among the ranking measures most often used are the following standard information retrieval measures: (a) half-life (1.9) (Breese, Heckerman, & Kadie, 1998), which assumes an exponential decrease in the interest of users as they move away from the recommendations at the top, and (b) discounted cumulative gain (1.10) (Baltrunas, Makcinskas, & Ricci, 2010), wherein decay is logarithmic.

$$HL = \frac{1}{\#U}\sum_{u\in U}\sum_{i=1}^{N}\frac{\max\left(r_{u,p_i}-d,0\right)}{2^{(i-1)/(\alpha-1)}} \tag{1.9}$$

$$DCG^k = \frac{1}{\#U}\sum_{u\in U}\left(r_{u,p_1}+\sum_{i=2}^{k}\frac{r_{u,p_i}}{\log_2(i)}\right) \tag{1.10}$$

p₁, … , pₙ represents the recommendation list, $r_{u,pi}$ represents the true rating of the user *u* for the item p_i, *k* is the rank of the evaluated item, *d* is the default rating, *α* is the number of the item on the list such that there is a 50% chance the user will review that item.

1.2.3.4.2. Novelty and Diversity

A novelty metric measures how acquainted the users are with the items the RS recommend them. A diversity metric measures how different the items recommended are.

Currently, novelty and diversity measures do not have a standard definition; therefore, different authors propose different metrics (Nehring & Puppe, 2002; Vargas & Castells, 2011). Certain authors have (Hurley & Zhang, 2011) used the following formalization:

$$diversity_{Z_u} = \frac{1}{\#Z_u\left(\#Z_u-1\right)}\sum_{i\in Z_u}\sum_{j\in Z_u}^{j\neq i}\left[1-sim(i,j)\right] \tag{1.11}$$

$$novelty_i = \frac{1}{\#Z_u-1}\sum_{j\in Z_u}\left[1-sim(i,j)\right],\ i\in Z_u \tag{1.12}$$

Here, *sim(i,j)* indicates item to item memory-based CF similarity metrics. Z_u indicates the set of *n* recommendations to user *u*.

1.2.3.4.3. Stability

The stability in the predictions and recommendations influences on the users' trust towards the RS. A RS is stable if the predicitions it provides do not change strongly over a short period of time. Adomavicius and Zhang (2010) propose a quality measure of stability, MAS (Mean Absolute Shift). This measure is defined through a set of known ratings R_1 and a set of predictions of all unknown ratings, P_1. For an interval of time, users of the RS will have rated a subset *S* of these unknown ratings and the RS can now make new predictions, P_2. MAS is defined as follows:

$$stability = MAS = \frac{1}{|P_2|} \sum_{(u,i)\in P_2} \left| P_2(u,i) - P_1(u,i) \right| \quad (1.13)$$

1.2.3.4.4. Reliability

The reliability of a prediction or a recommendation informs about how seriously we may consider this prediction. When RS recommends an item to a user with prediction 4.5 in a scale *{1, ..., 5}*, this user hopes to be satisfied by this item. However, this value of prediction (4.5 over 5) does not reflect with which certain degree the RS has concluded that the user will like this item (with value 4.5 over 5). Indeed, this prediction of 4.5 is much more reliable if it has obtained by means of 200 similar users than if it has obtained by only two similar users.

In Hernando, Bobadilla, Ortega, and Tejedor (2013), a reliability measure is proposed according the usual notion that the more reliable a prediction, the less liable to be wrong. Although this reliability measure is not a quality measure used for comparing different techniques of RS through cross validation, this can be regarded as a quality measure associated to a prediction and a recommendation. In this way, the RS provides a pair of values (prediction value, reliability value) through which users may balance its preference: for example users would probably prefer the option (4, 0.9) to the option (4.5, 0.1). Consequently, the reliability measure proposed in Hernando, Bobadilla, Ortega, and Tejedor (2013) provides a new understandable factor, which users may consider for taking its decisions. Nevertheless, the use of this reliability measure is just constrained to those RS based on the kNN algorithm.

The definition of reliability on the prediction, $p_{u,i}$, is based on two numeric factors: $s_{u,i}$ and $v_{u,i}$. $s_{u,i}$ measures the similarity of the neighbors used for making the prediction $p_{u,i}$; $v_{u,i}$ measures the degree of disagreement between these neighbors rating the item *i*. Finally, the reliablity measure is defined as follows:

$$s_{u,i} = \sum_{x \in K_{u,i}} sim(u,x) \tag{1.14}$$

$$f_s(s_{u,i}) = 1 - \frac{\overline{s}}{\overline{s} + s_{u,i}} \tag{1.15}$$

$$v_{u,i} = \frac{\sum_{x \in K_{u,i}} sim(u,x) \cdot \left(r_{x,i} - \overline{r}_x - p_{u,i} + \overline{r}_u\right)^2}{\sum_{x \in K_{u,i}} sim(u,x)} \tag{1.16}$$

$$f_v(v_{u,i}) = \left(\frac{\max - \min - v_{u,i}}{\max - \min}\right)^{\gamma} \tag{1.17}$$

$$\gamma = \frac{\ln(0,5)}{\ln\left(\frac{\max - \min - \overline{v}}{\max - \min}\right)} \tag{1.18}$$

$$reliability_{u,i} = \left(f_s(s_{u,i}) \cdot f_v(v_{u,i})^{f_s(s_{u,i})}\right)^{\frac{1}{1+f_s(s_{u,i})}} \tag{1.19}$$

Where $\overline{s}$ and $\overline{v}$ are respectively the median of the values of $s_{u,i}$ and $v_{u,i}$ in the specific RS. $K_{u,i}$ is the set of neighbors of *u* which have rated the item *i*. *[min..max]* is the discrete range of rating values.

1.2.4. Recommendations Explanation

An important research subject in the RS field focuses on providing explanations that justify the recommendations the user has received. This is an important aspect of an RS because it aids in maintaining a higher degree of user confidence in the results generated by the system.

The type of explanations used can be classified as follows (Papadimitriou, Symeonidid, & Manolopoulus, 2012; Gedikli, Jannach, & Ge, 2014):

- *Human style explanations (user to user approach).* For example, we recommend movie *i* because it was liked by the users who rated movies *j*, *k*, *m*, … very positively (*j*, *k*, *m*, … are movies well rated by the active user).

- *Item style explanations (item to item approach).* For example, we recommend the vacation destination *i* because you liked the vacation destinations *g*, *c*, *r*, ... (*g*, *c*, *r*, ... are vacation destinations similar to *i* and rated well by the active user).
- *Feature style explanations (it is recommended based on items' features).* For example, we recommend movie *i* because it was directed by director *d*, it features actors *a*, *b*, and it belongs to genre *g* (*d*, *a*, *b*, *g* are features the active user is interested in).
- *Hybrid methods*. This category primarily includes the following combinations: human/item, human/feature, feature/item, and human/feature/ item.

Additionally, in geo-social RS (Foursquare, Yelp, TripAdvisor, etc.), location information exists that must be used in the recommendation explanation mechanism (Yang, Cheng, & Dia, 2008). Geo-social RS typically adopt a hybrid human/item explanation method based on social, location and memory-based information.

There is a reference publication (Herlocker, Konstan, & Riedl, 2000) that is a helpful introduction to the RS explanations research field. They explore the utility of explanations in CF RS, and they stated three key research questions: (1) What models and techniques are effective in supporting explanations? (2) Can explanation facilities increase the acceptance of CF RS? (3) Can explanation facilities increase the filtering performance of the CF RS users? To answer to the first question, they propose using rating histograms, indications of past performance, comparisons to similar rated items, and use of domain specific content features. The results from the experiments conducted with RS users support an affirmative response to the second question. The third question is unanswered because users perform filtering based on many different channels of input.

A dynamic approach that favors the mechanisms for RS explanations includes using conversational techniques, such as the CCBR (conversational case-base reasoning), explained into McSherry (2005). As CCBR they use an incremental nearest neighbor process based on the Pareto case dominance approach. In a different study (McCarthy, Reilly, McGinty, & Smyth, 2004), a dynamic approach is also adopted, but it employs a different perspective. Instead of attempting to justify a particular recommendation they focus on how explanations can help users to understand the recommendation opportunities that remain if the current recommendation should not meet their requirements. They generate compound critiques as explanations: users have the opportunity to accept or critique recommendations. If they critique a recommendation, the critique acts as a filter over the remaining recommendations.

In a separate study, Bilgic and Mooney (2005) differentiate between the concepts promotion (increasing of the acceptance of the recommended item) and satisfaction (user satisfaction with the recommended item). They also produced better results by using the keyword style explanation (based on content data) compared with the neighbor style explanation (human style explanation). Authors propose a new classification of the recommendation justifications: keyword style explanation (for content-based RS), neighbor style explanation (for collaborative filtering RS) and influence style explanation (tells the user how their interactions with the RS influences the recommendation). Tintarev and Masthoff (2007) describe the advantages of making justifications in recommendations: transparency, scutability, trustworthiness, effectiveness, persuasiveness, efficiency and satisfaction.

Billus and Pazzani (1999) propose a recommendation system on news, which provides keyword style justifications of the recommendations through the weights used for obtaining these recommendations. Wang, De Vries, and Reinders (2006) describe a system of justifications based on the features of users' preference. Tintarnev and Masthoff (2007) design a recommedation system on films whose recommendations are justified through the features. Vig, Sen, and Riedle (2009) propose a mechanism for justifying recommendations called *tagsplanations*, which is based on community tags. *Trangsplanations* have two key components: tag relevance, the degree to which a tag describes an item; and tag preference, the user's sentiment toward a tag.

Yetim (2008) provides a framework for organizing justifications, used to categorize explanations; they propose the categorization of the discourse: explicative, theoretical, pragmatic, ethical, moral, legal, aesthetic, and personal. Although this theoretical framework has not been used into the research literature, it can be used to design new types of explanations.

Hernando, Moya, Ortega, and Bobadilla (2013) explain item to item recommendations using a novel technique based on the visualization of trees of items. Hernando, Bobadilla, Ortega, and Gutiérrez (2013) use tree visualization technique to provide valuable information about the reliability of recommendations and the importance of the ratings the user has made. Ortega, Bobadilla, Hernando, and

Rodríguez (2014) provide a method that facilitates the explanation of large numbers of recommended items by the generation of tree graphs containing: the recommended items, the items that have appeared most often in the recommendation process, the relative importance of the items, and the relationships that exist among the items.

The most relevant researches related to provide justifications in recommender systems include a study (Pu & Chen, 2007) on a new organization interface where results are grouped according to their tradeoff properties. They have developed a trust model for recommender agents based on the Pareto algorithm (excluding dominated categories). Symeonidis, Nanopoulos, and Manolopoulos (2008) first construct a feature profile for the users to reveal their favorite features, later they group users into biclusters to exploit partial matching between de preferences of groups of users over groups of items. Additionally they propose a metric to measure the quality of justifications: the explain coverage ratio. In Symeonidis, Nanopoulos, and Manolopoulos (2009) they use a prototype "MoviExplain" to put into the test the research showed into Symeonidis, Nanopoulos, and Manolopoulos (2008). In Hu, Koren, and Volinsky (2008) they use implicit feedback to derive an estimate of the user preference (like or dislike an item) and user confidence for each user-item pair.

1.2.5. Other Type of Recommender Systems

1.2.5.1. Geospatial RS

Due to the increasing use of mobile devices, location-aware systems are becoming more widespread. These systems show a tendency towards their consolidation as web 3.0 services and this naturally leads to location-aware CF and location-aware RS, which can be called geographic CF and geographic RS.

We introduce a classification for geographic CF RS and focus on the most relevant section of the classification obtained. The cases identified are:

- *RS*: Traditional RS, in which ratings and recommendations are made without using geographical information.
- *RS + G*: Traditional RS, which also contributes the item's geographical position. These RS cannot be regarded as geographic RS, as the geographic information does not play a part in the recommendation process. Geographic information is used afer the recommendation process.

- *GRS*: This group of Geographic RS is most likely to become popular in the near future. In these, whereas ratings are calculated in a traditional way, the recommendations are obtained by considering the geographical position of the user to whom the recommendation is to be made. A representative example is that of a RS for restaurants; the users rate a restaurant using very diverse concepts, which do not include the distance at the time of voting between the user and the restaurant. However, users of a Geographic RS expects a restaurant to be recommended to them not only because of good ratings from similar users (k-neighbors), but also according to the distance between their current position and that of the restaurant.
- *GRS+*: In this case, users establish ratings on items by weighting the distance between them and the items rated. In this type of geographic RS two possibilities can be established:
  - (a) Each item accepts only one rating for each user.
  - (b) Each item accepts more than one rating for each user, taking into account the geographical position of the user when rating.

From a theoretical point of view, GRS+(b) are the most complete; however, from a practical point of view, they involve a semantic difficulty in the item rating process, which makes their use very difficult. Rating items in this GRS+ involves that each user can rate items according to the relative distances between the user and the items. In this way, a user can rate a restaurant from their home differently to how they would rate it from their workplace; and when the distances are very different, the ratings are also likely to be so. The mental process would be something like this: I am 1 km far away from the restaurant and I rate very positively travelling 1 km to go to that restaurant which I think is good; but after some time, the same user, who is at work, 24 km far away from the restaurant, could rate it indicating that they do not consider it to be positive to travel 24 km to go to the restaurant even if they think it is good.

In summary, GRS+ have the advantage that they accept a wider variety of ratings and that these also contain the relative importance that each user gives to the items according to the distance required to access them. The disadvantage is that it is difficult to involve users in a particularly complex and demanding ratings process.

At present, there are few publications regarding geospatial RS. This is due, to a great extent, to the lack of public databases that include ratings and geographic positions capable of being combined in a RS. Some of the publications that focus more closely on the field are as follows:

Martinez, Rodriguez, and Espinilla (2009) and Biuk-Aghai, Fong, and Tain-Whar (2008) are examples of the RS + G group. In Schlieder (2007), they propose a novel approach for modeling the collaborative semantics of geographic folksonomies. This approach is based on multi-object tagging, that is, the analysis of tags that users assign to composite objects. This paper is based on the concept of groups of people who share a common geospatial feature data dictionary (including definitions of feature relationships) and a common metadata schema.

Yang, Chen, and Dia (2008) can be considered as a hybrid content based/geographic RS. The core of the system is a hybrid content based/geographic recommendation mechanism that analyzes a customer's history and position so that vendor information can be ranked according to the match with the preferences of a customer.

Matyas and Schlieder (2009) show a collaborative system that lies between a RS and a GRS. In this case, the users' ratings are taken based on the photos they have downloaded from a Web 2.0 and the photos they have uploaded to the same Web (the photos have a GPS address associated to them). After this, a search of k-neighborhoods based on this data is carried out. The recommendation process does not take into account the user's position.

Colombo-Mendoza, Valencia-García, Rodríguez-González, Alor-Hernández, and Samper-Zapater (2014) propose a context-aware mobile recommender system in the movie showtimes domain. The system proposed is called RecomMetz, and it considers three different kinds of contextual information: location, crowd and time. In a nutshell, RecomMetz has unique features: (a) the items to be recommended have a composite structure (movie theater + movie + showtime), (b) the integration of the time and crowd factors into a context-aware model, and (c) the implementation of an ontology-based context modeling approach. This paper can be clasfied as GRS+.

It is possible to collect travel GPS traces from users and use the database to generate recommendations (Zheng & Xie, 2011). The travel GPS traces can be reinforced with social information based on friends (Zheng, Zhang, Ma, Xie, & Ma, 2011). Both papers can be classified as GRS+.

1.2.5.2. Social RS

As the web 2.0 has developed, RS have increasingly incorporated social information (Lerman, 2013) (e.g., trusted and untrusted users, followed and followers, friends lists, posts, blogs, and tags). This new contextual information (Ma, Zhou, Lyu, & King, 2011; Tan, Bu, Chen, & He, 2011) improves the RS. Social information reduces the sparsity problem inherent in memory-based RS because social information reinforces traditional memory-based information (users ratings): users connected by a network of trust exhibit significantly higher similarity on items and meta-data that non-connected users (Lee & Brusilovsky, 2009). This additional information increase the computational complexity required to obtain recommendations, so it is necessary to find an equilibrium between performance and accuracy (Kang & Lerman, 2013b).

Social information is used by researchers with three primary objectives: (a) to improve the quality of predictions and recommendations (Arazy, Kumar, & Shapira, 2009; Carrer-Neto, Hernández-Alcaraz, Valencia-García, & García-Sánchez, 2012), (b) propose or generate new RS (Siersdorfer &Sizov, 2009; Li, Liao, & Lai, 2011; Zhu & Lerman, 2014), and (c) elucidate the most significant relationships between social information and collaborative processes (Perugini, Gonçalves, & Fox, 2004; Hossain & Fazio, 2009).

Trust and reputation is an important area of research in RS (O'donovan & Smyth, 2005); this area is closely related to the social information currently included in RS (Josang, Ismail, & Boynd, 2007). The most common approachs to generating trust and reputation measurements are the following:

(a) *User trust*: to calculate the credibility of users through explicit information of the rest of users (Li & Kao, 2009; Yuan, Guan, Lee, lee, & Hur, 2010) or to calculate the credibility of users through implicit information obtained in a social network (Massa & Avesani, 2004; Park, Hong, & Cho, 2007).

(b) *Item trust*: to calculate the reputation of items through a feedback of users (Josang, Ismail, & Boynd, 2007) or to calculate the reputation of items studying how users work with these items (Cho, Kwon, & Park, 2009; Kitisin & Neuman, 2009).

In the social RS field, users can introduce labels associated with items. The set of triples *<user, item, tag>* form information spaces referred to as folksonomies. Fundamentally, folksonomies are used in the following two ways: (a) to create tag recommendation systems (RS based only on tags) (Marinho & Schmidt-thieme, 2008) and (b) to enrich the recommendation processes using tags (Gedikli & Jannach, 2010).

Content-based filtering has recently become more important due to the surge in social networks. RS show a clear trend to allow users to introduce content (Perugini, Gonçalves, & Fox, 2004; Arazy, Kumar, & Shapira, 2009), such as comments, critiques, ratings, opinions and labels as well as to establish social relationship links (e.g., followed, followers, like user and dislike user). This additional information increases the accuracy of predictions and recommendations, which has generated a variety of research articles: Kim, Alkhaldi, Saddik, and Jo (2011), Zheng and Li (2011) and Carrer-Neto, Hernández-Alcaraz, Valencia-García, and García-Sánchez (2012).

1.2.5.3. Bio-inspired RS

Much of the proposed model-based RS are based on bio-inspired approaches, which primarily use Genetic Algorithms (GAs) and Neural Networks (NNs). Models have also been proposed based on Artificial Immune Networks (AINs).

Methods based on GA are heuristic approaches based on evolutionary principles such as natural selection and survival of the fitest. GA have mainly been used in two aspects of RS: clustering (Kim & Ahn, 2005; Zhang & Chang, 2006) and hybrid user models (Ho, Fong, & Yan, 2007; Al-Shamri & Bharadwaj, 2008; Gao & Li, 2008). A common technique to improve the features of RS consists of initially carrying out a clustering on all of the users, in such a way that a group of classes of similar users is obtained, after this, the desired CF techniques can be applied to each of the clusters, obtaining similar results but in much shorter calculation times. It is usual to use common genetic clustering algorithms such as GA-based K-means (Kim & Ahn, 2008).

The RS hybrid user models commonly use a combination of CF with demographic filtering or CF with content based filtering, to exploit merits of each one of these techniques. In these cases, the chromosome structure can easily contain the demographic characteristics and/or those related to content-based filtering.

In order to tackle location-based advertisement, Dao, Jeong, and Ahn (2012) propose a model-based CF using GA. They combine both user's preferences and interaction context. Bobadilla, Ortega, Hernando, and Alcalá (2011) use GA to create a similarity metric weighting a set of very simple similarity measures. Hwang, Su, and Tseng (2010) employ a GA to learn personal preferences of customers.

NN is a model based on the observed behavior of biological neurons. This model, intended to simulate the way the brain processes information, enables the computer to "learn" to a certain degree. A NN typically consists of a number of interconnected nodes. Each handles a designated sphere of knowledge, and has several inputs from the network. Based on the inputs it gets, a node can "learn" about the relationships between sets of data, pattern, and, based upon operational feedback, are molded into the pattern required to generate the required results.

The RS most relevant research available in which NN usually focuses is hybrid RS, in which NN are used for learn users profiles; NN have also been used in the clustering processes of some RS.

The hybrid approaches enable NN to act on the additional information to the ratings. In Ren, He, Gu, Xia, and Wu (2008) they propose a hybrid recommender approach that employs Widrow-Hoff (Widrow & Hoff, 1988) algorithm to learn each user's profile from the contents of rated items. This improves the granularity of the user profiling. In Christakou and Stafylopatis (2005) they use a combination of content-based and CF in order to construct a system that provides more precise recommendations concerning movies. In Lee, Choi, and Woo (2002) first, all users are segmented by demographic characteristics and users in each segment are clustered according to the preference of items using the Self-Organizing Map (SOM) NN. Kohonen's SOMs are a type of unsupervised learning; their goal is to discover some underlying structure of the data.

Two alternative NN uses are presented in Huang, Chuang, Ke, and Sandnes (2008) and Roh, Oh, & Han (2003). In the first case paper, authors use a training backpropagation NN for generating association rules that are mined from a transactional database. In the second paper, authors propose a model that combines a CF algorithm with two machine learning processes: SOM and Case Based Reasoning (CBR) by changing an unsupervised clustering problem into a supervised user preference reasoning problem.

Neuro-fuzzy inference has been used in Sevarac, Devedzic, and Jovanovic (2012) to create pedagogical rules in e-learning. A new cold-start similarity measure has been perfected in Bobadilla, Ortega, Hernando, and Bernal (2012a) using optimization based on neural learning.

Artificial immune systems are distributed and adaptive systems using the models and principles derived form the human immune system. They model the defence system, which can protect our body against infections. In order to tackle the RS sparsity problem and to make algorithms more scalable, Acilar and Arslan (2008) present a new CF model based on the AIN Algorithm (aiNet). AIN were previously proposed to general recommendations (Caizer & Aickelin, 2005) and to recommend web sites (Morrison & Aickelin, 2002).

1.3. Group Recommendations

RS that consider groups of users (Jameson & Smyth, 2007; Boratto & Carta, 2011) are starting to expand and to be used in different areas: tourism (Ardissono, Goy, Petrone, Segnan, & Torasso, 2003; McCarthy, Salamó, Coyle, McGinty, Smyth, & Nixon, 2006), music (Chao, Balthrop, & Forrest, 2005), TV (Yu, Zhou, Hao, & Gu, 2006) and web (Pazzani & Billsus, 2007).

Given the specific characteristics of the recommendation to groups, Xie and Lui (2014) present a formal mathematical model of a group recommendation system. The model analyzes how the rating schemes affect the recommender system. They answer questions such us: what is the desired number of ratings per product so to guarantee an accurate recommendation?; what are some effective voting rules in summarizing ratings?; or how users' misbehavior such as cheating, in product rating may affect the recommendation accuracy?. Other authors (Amer-Yahia, Roy, Chawlat, Das, & Yu, 2009; Roy, Amer-Yahia, Chala, Das, & Yu, 2010) analyze the core of group recommendation and propose a formal semantics that accounts for both item relevance to a group and disagreements among group members.

Different model based techniques has been applied to group recommendation problem. Alonso, Cabrerizo, Chiclana, Herrera, and Herrera-Viedma (2009) propose, as a front-end, the incorporation of a process of estimation of missing information when dealing with incomplete fuzzy linguistic preference relations. Boratto & Carta (2014a) evaluate group-modeling strategies in a group recommendation scenario in which groups are detected by clustering users. Chen, Cheng, and Chuang (2008) propose a novel group recommendation system based on the framework of collaborative filtering. The novelty of this method is that they use genetic algorithm to predict the possible interactions among group members. Recently studies (Christensen & Schiaffino, 2013; Christensen & Schiaffino, 2014) combine a matrix factorization model to estimate unknown ratings with social network analysis that express social influence factors in a group, such us, cohesion, social similarity and social centrality. Sotelo, Blanco-Fernández, López-Nores, Gil-Solla, and Pazos-Arias (2009) present an approach to content recommendation for groups of people, based on TV contents descriptions and semantic reasoning techniques.

To measure the quality of the recommendations performed to a group of users, some authors has proposed different quality measures. Boratto and Carta (2014b) study the impact of content novelty on the accuracy of a group recommender system, by introducing a constraint on the percentage of a group for which the recommended content has to be novel. Rao, Pujari, and Padmanabhan (2014) introduce monotonicity, which tries to measure the number of items continued to be recommended when a recommendation technique is used incrementally.

Group recommendations trends are focused on include social factors to the recommendation process. Quijano-Sánchez, Díaz-Agudo, and Recio-García (2014) improve the overall quality of group recommendations through the addition of social knowledge to existing recommendation strategies. They use the information stored in social networks to elicit social factors following two approaches: the cognitive modeling approach and the social approach. Garcia and Sebastia (2014) argue that the aggregation of user preferences to create a group is not a realistic situation, because not all the members in a group act in the same manner. They have opted for building a multi-agent system, where an agent acts on behalf of one group member. Then, different *UserAgents* negotiate with the aim of building a group profile that satisfies their particular minimum requirements. Gartrell, Xing, Lv, Beach, Han, Mishra, and Seada (2010) propose a group recommendation method that utilizes both social and content interests of group members. They study the key characteristics of groups and propose a group consensus function that captures the social, expertise, and interest dissimilarity among multiple group members.

Collaborative filtering has been the most studied method to perform group recommendations. Some authors (Berkovsky & Freyne, 2010; García, Sebastia, & Onaindia, 2011; Christensen & Schiaffino, 2011) propose different alternatives to include group recommendation capabilities into collaborative filtering combining the predictions generated to each individual member of the group to compute a prediction to the group of users. Baltrunas, Makcinskas, and Ricci (2010) use rank aggregation of individual lists of recommendations to merge the individual recommendation list into one recommendation list for the group of users. Bobadilla, Ortega, Hernando, and Bernal (2012b) present a group recommendation method inspired on memory-based CF. The method computes the intersection of a large number of neighbors of each user of the group to create a new set of neighbors for the entire group. Kim, Kim, Oh, and Ryu (2010) propose a recommendation procedure for group activities in online communities. It consists of two phases: (1) generate a recommendation set for a group using collaborative filtering method, and (2) remove irrelevant items from the recommendation set in order to improve satisfaction of individual members' preferences.

The paper on which this PhD thesis is based on (Ortega, Bobadilla, Hernando, & Gutiérrez, 2013) studies the performance of group recommendation techniques in order to propose a new method. The proposed method highly improve the performance of the recommendation process without deteriorate the accuracy of it. The key factor of this improvement is focused on determining when the data should be unified from the users of the group to group of users.

2. HYPOTHESIS AND MOTIVATION

2.1. Introduction

The social networks have broadly increased the collaborative use of the Internet. Now, users are not isolated, they interact with other users through communities, followers and friends. This fact forces us to rethink all bases for with the web was designed. We need to develop new technologies and algorithms that satisfy the user necessities.

In the field of recommender systems, the new trends in social networks can bring us new user recommendation types. The earliest recommender systems were focused on compute a set of MANY recommendations to ONE user. Nowadays, we can use the social information about groups of users to generate a set of MANY recommendations to a group of MANY users.

There are certain contexts in which the group recommendations are more important than the individual recommendations. See a movie, go out to dinner at a restaurant, or going on vacation, are actions that we usually do in group. Therefore, in these contexts, it is more interesting to provide a recommendation for a group of user that several recommendations for individual users.

2.2. Motivation

Group of users recommendations present a new challenge to recommender systems. These has always been focused on individual recommendations, so if we want to add the capability of compute group of users recommendations, we will need to make major changes in the original algorithms in order to achieve an efficient and precise group recommendations results. The notion of generating a set of recommendations that will satisfy a group of users with potentially competing interest is callenging in itself.

The group recommendation problem opens a new research area on the recommender systems' field. This area has been slightly studied and not too many publications about it can be found. The capabilities of the group recommendation research are endless. Adapt current recommender systems methods to work with group of users, design new algorithms for group recommendations, and include social information generated by the web 2.0 to the group recommendations process, are some of the highlights that this new research area can offer us.

This PhD thesis is focused on adjust the traditional collaborative filtering algorithm to the group of users recommendations problem. Different alternatives to perform this adjustment can be found. This PhD thesis will analyze them and will propose an accurate, precise and efficient solution that improves the studied methods.

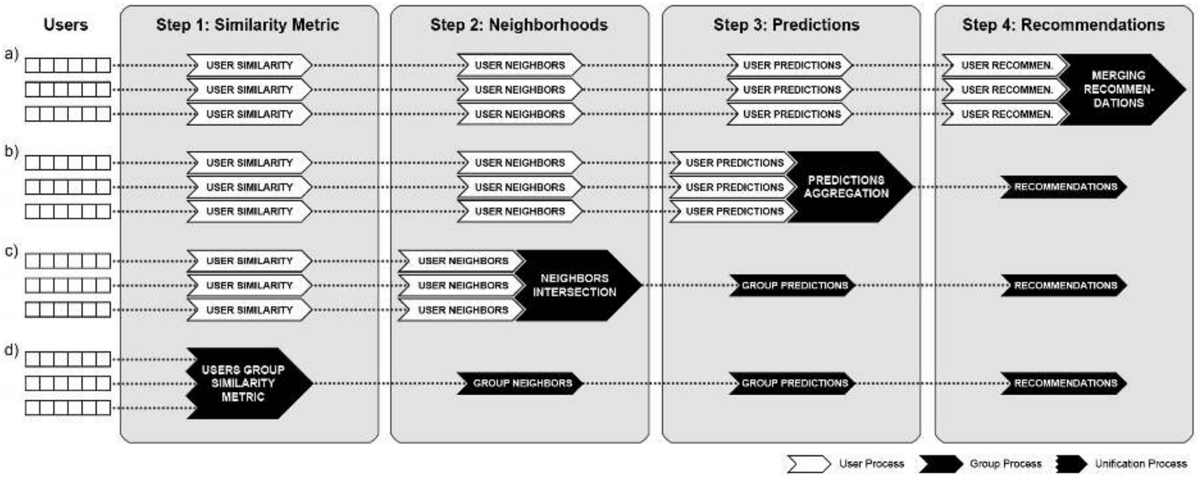


Figure 2.1. State of the art of the CF recommendations to a group of users.

The **Figure 2.1** represents the state of the art of the CF recommendation to a group of users. The individual members of a group are represented on the left; each square represents the user rating to an item. The white boxes represent processes carried out for one user and the black boxes represent processes carried out for a group of users. The biggest black boxes show key information: they indicate the CF process phase where the unification is performed: "*n users* → *1 group*".

The graph shows the four representative cases of tackling the solution to recommendation by groups. In the case (a), the recommendations obtained for each individual user of the group are merged into one recommendation for the group. Baltrunas, Makcinskas and, Ricci (2010) use rank aggregation of individual lists of recommendations.

In the case (b), the data unification is performed in the prediction phase of the CF process: n individual predictions of n users of the group are combined into one prediction of the group (predictions aggregation). This approach has been followed by

Berkovsky and Freyne (2010), García, Sebastia, and Onaindia (2011) and Christensen and Schiaffino (2011).

In the case (c), the set of neighbors of the group of users are obtained by unifying the sets of neighbors of the individual users. Bobadilla, Ortega, Hernando and Bernal (2012b) followed this approach proposing the intersection of a large number of neighbors of each user of the group.

The case (d) shows the approach proposed in this PhD thesis: design of a similarity metric that acts directly on the set of votes of the group's users. This solution is the only one that directly provides a set of neighbors for the group of users.

2.3. Hypothesis

There are different alternatives to include group recommendations based on traditional collaborative filtering algorithm. All of them try to unify the data from each user of the group into the group of users in one of the main stages of the collaborative filtering process: k-neighbors search, rating prediction or item recommendation. However, none of these previous works have studied the impact on the system performance of the stage where this unification process is made.

The hyphotesis proposed in this PhD thesis is the following:

The earlier unification of the users of the group into a group of users
improves the overall performance of the group recommendation process
without deteriorate the quality of the generated recommendations.

This hypothesis states that if the users' group information is unified as soon as possible, less computational cost will be required to carry out the recommendations. Furthermore, this should not adversely affect the quality of recommendations that the group of users receive.

3. FORMALIZATION

3.1. Preliminary Definitions

We define a CF RS with *n* users and *m* items. The following sets are defined:

Let $U = \{u \in U \mid 1 \le u \le n\}$ be the set of users. (3.1)

Let $I = \{i \in I \mid 1 \le i \le m\}$ be the set of items. (3.2)

In the CF approach is necessary that the users rate the items, so we define the following set:

Let $V = \{v \in \mathbb{N} \mid \min \le v \le \max\} \cup \{\bullet\}$ be the set of ratings. (3.3)

Where *min* is the minimun value of a rating, *max* is the maximum value of a rating and • represents the absence of rating.

Let $r_{u,i} \in V$ be the rating of the user *u* to the item *i.* (3.4)

The main difference between the proposed method and the traditional CF RS is that the recommendations are made to a group of users instead of a single user. It is necessary to make a formal definition of the group that receives the recommendations:

Let $G \subset U$ be a group of users. (3.5)

The number of users into a group is arbitrary. The proposed method works for any group size. However, the size of the group must be always smaller that the number of users of the RS. Furthermore, the proposed method is more accurate when the size of the group is small than when the size of the group is big.

We define the cardinality of a set *C* as its number of valid elements and we represent it with $\#C = \#\{x \in C \mid x \ne \bullet\}$.

3.2. Overview

Figure 3.1 summarizes the group recommendation method shown in this PhD thesis. Step (1) selects the users of the group. Step (2) interchanges the information between the group of users and each user of the RS in order to compute the KNN.

Step (3) performs recommendations to the group of users using the information of the KNN.

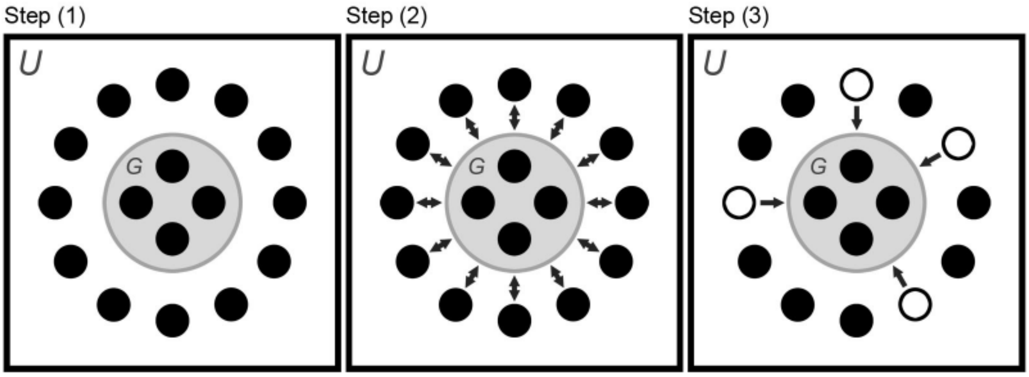


Figure 3.1. Proposed method overview.

3.3. Computing KNN

The first step to get recommendations is to find those users who have a high similarity with the group of users. We define the *Users Group Similarity Metric* (UGSM) that aims to quantify the similarity between any user *u* of the RS and the set of users belonging to group *G*.

Let *UGSM (G, u)* be the similarity metric between a user *u* of the RS and the group *G*. This similarity metric returns a value directly proportional to the number of cases in which user *u's* rating coincides with that of any of the members of group *G*. It is reasonable to believe that if user *u's* opinion coincides with that of the majority of the members of the group, user *u* will be capable of proposing new items that the majority of the users belonging to group *G* will like. Formally:

$$UGSM(G,u) = \begin{cases} \dfrac{1}{\# I_{G,u}} \displaystyle\sum_{i \in I_{G,u}} \dfrac{\#\{g \in G \mid r_{g,i} = r_{u,i}\}}{\#\{g \in G \mid r_{g,i} \neq \bullet\}} & I_{G,u} \neq \varnothing \wedge u \notin G \\ \bullet & otherwise \end{cases} \tag{3.6}$$

$$UGSM(G,u) \in [0,1] \tag{3.7}$$

Where:

$I_u = \#\{i \in I \mid r_{u,i} \neq \bullet\}$ is the set of items rated by user *u*.

$I_G = \bigcup_{g \in G} I_g$ is the set of items rated by at least one of the users of the group *G.*

$I_{G,u} = I_G \cap I_u$ is the set of items rated both user *u*and at least one user of *G*.

High values of UGSM represent a high similarity between the group and the user. Low values represent low similarity.

Using the similarity between the group of users and each user that do not belong to the group, we compute *k* nearest neighbors of the group *G*. These *k* users are the most similar to the group of users.

We define K_G as the *k* users who have a higher similarity with the group. The following expressions must hold:

$$K_G \subset U \tag{3.8}$$

$$\# K_G = k \tag{3.9}$$

$$\forall x \in K_G, \forall y \in (U - K_G) : UGSM(G,x) \geq UGSM(G,y) \tag{3.10}$$

3.4. Predictions and Recommendations

We can now compute the predictions to the items not rated by any members of the group: we find the rating average of the *k* neighbors on these items. Formally, we calculate the prediction on an item *i* for group *G* ($p_{G,i}$) using the weighted average aggregation approach:

Let $P_{G,i} = \{u \in K_G \mid r_{u,i} \neq \bullet\}$ be the set of neighbors which have rated item *i*.

$$p_{G,i} = \frac{\sum_{u \in P_{G,i}} UGSM(G,u) \cdot r_{u,i}}{\sum_{u \in P_{G,i}} UGSM(G,u)} \Leftrightarrow P_{G,i} \neq \varnothing \tag{3.11}$$

$$p_{G,i} = \bullet \Leftrightarrow P_{G,i} \neq \varnothing \tag{3.12}$$

The items recommended to the group will be those that have not been rated by any of the group's users and which have received a higher prediction. We $X_G$define as the set of items subject to being recommended to group *G*. The following expressions must be verified:

$$X_G \subset \left(I - I_G\right) \tag{3.13}$$

$$\forall i \in X_G : p_{G,i} \neq \bullet \tag{3.14}$$

We define *Z_G* as the set of *N* items recommended to group *G*. The following expressions must be verified:

$$Z_G \subseteq X_G \tag{3.15}$$

$$\# Z_G = \min\left(N, \# X_G\right) \tag{3.16}$$

$$\forall x \in Z_G, \forall y \in \left(X_G - Z_G\right) : p_{G,x} \geq p_{G,y} \tag{3.17}$$

3.5. Running Example

The following running example tries to clarify how the proposed method computes the recommendations to the group of users.

We define a CF RS with 8 users and 10 items. **Table 3.1** contains the ratings matrix. The users can rate the items in a scale from 1 (do not like it) to 5 (love it). The symbol • represents the absence of rating.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	•	•	1	2	•	3	•	5	•	4
u_2	5	•	2	•	•	3	•	5	•	4
u_3	5	•	5	1	•	4	2	3	•	•
u_4	4	•	3	2	3	3	•	4	5	4
u_5	5	5	5	•	•	3	•	5	3	•
u_6	•	3	•	1	•	4	1	4	•	3
u_7	3	4	•	2	•	3	2	5	4	4
u_8	4	3	•	•	2	•	1	•	5	2

Table 3.1. Running example ratings matrix.

We define G as the group of users. For this running example, this group will be composed of u_1, u_2 and u_3.

$$G=\{u_1,u_2,u_3\} \tag{3.18}$$

The first step is to compute the similarity between the group *G* and each user of the RS using *UGSM(G, u)*. **Table 3.2** shows the computed similarity values. High values represent a high similarity between the group and the user. Low values represent low similarity between the group and the user. The symbol • represents the absence of similarity.

	u₁	u₂	u₃	u₄	u₅	u₆	u₇	u₈
UGSM(G,u)	•	•	•	.361	.667	.167	.639	0

Table 3.2. Similarity between G and each user of the RS.

The following example shows how the similarity between group *G* and user u_4 has been computed.

$$I_{G,u_4}=\{i_1,i_3,i_4,i_6,i_8,i_{10}\} \tag{3.19}$$

$$UGSM\left(G,u_4\right)=\frac{1}{6}\left(\frac{0}{2}+\frac{0}{3}+\frac{1}{2}+\frac{2}{3}+\frac{0}{3}+\frac{2}{2}\right)=\frac{13}{36}=0.361 \tag{3.20}$$

The next step is to get the KNN of the group *G*. For this running example we have selected k = 3. Equation (3.21) shows the 3 users with higher similarity to the group *G*.

$$K_G=\{u_5,u_7,u_4\} \tag{3.21}$$

Table 3.3 contains predictions generated by the KNN of the items not rated by any user of the group G.

	i₁	i₂	i₃	i₄	i₅	i₆	i₇	i₈	i₉	i₁₀
$p_{G,i}$	•	4.512	•	•	3.0	•	•	•	3,818	•

Table 3.3. Predictions generated to the group of users.

The following example shows how the prediction of the item *i₂* to the group *G* has been computed.

$$P_{G,i_2} = \{u_5, u_7\} \tag{3.22}$$

$$p_{G,i_2} = \frac{UGSM(G,u_5)\cdot r_{u_5,i_2} + UGSM(G,u_7)\cdot r_{u_7,i_2}}{UGSM(G,u_5) + UGSM(G,u_7)} = \frac{0.667\cdot 5 + 0.639\cdot 4}{0.667 + 0.639} = 4.512 \tag{3.23}$$

Finally, we establish the set of items recommended to the group. In this example we will recommend only one item to the group (*N = 1*). Therefore, we will select the item with the highest prediction value.

$$X_G = \{i_2, i_5, i_9\} \tag{3.24}$$

$$Z_G = \{i_2\} \tag{3.25}$$

4. EXPERIMENTS

4.1. The RecSys API

Experiments presented in this PhD thesis have been performed using RecSys API. The RecSys API has been developed at Escuela Técnica Superior de Ingeniería de Sistemas Informáticos (Universidad Politecnica de Madrid) by Fernando Ortega and Jesus Bernal. This API, written in Java, allows us to conduct experiments quickly, efficiently and safely.

The main objective of this API is to facilitate the experiments encoding. The RecSys API has two main highlights: simplify the data usage and expedite the validation of the experimental results. To simplify the data usage, we have defined an object hierarchy that allows the programmer an easily access and manipulation of the users, items and ratings sets. We have also encode parsers for the most commonly used RS datasets: MovieLens, Netflix, FilmAffinity and BookCrossing. To expedite the validation of the experimental results, the RecSys API automatically generates the training and test sets used by the experiments. Furthermore, it has an interface that allows us to easily encode experiments. Using this interface the experiment execution will be parallelized.

A wide variety of similarity metrics are currently implemented for the RecSys API. Most of the traditional metrics are implemented: Pearson correlation, MSD, cosine, adjusted cosine, Jaccard and Spearman rank. Some recently published similarity metrics are also implemented: JSMD (Bobadilla, Serradilla, & Bernal, 2010), CJSMD (Bobadilla, Ortega, Hernando, & Arroyo, 2013), PIP (Ahn, 2008) and Singularities (Bobadilla, Ortega, & Hernando, 2012).

The RecSys API can be use to generate user-to-user recommendations and item-to-item recommendations. The most common aggregation approaches are available to be used: average, weighted average, deviation from mean y weighted deviation from mean.

Finally, some quality measures are accessible to check the quality of the recommendations generated. Both quality of predictions and quality of recommendations has been implemented. To test quality of predictions we can use:

MAE and coverage. To test quality of recommendations we can use: precision, recall and F1.

4.2. Quality Measures

With the purpose of evaluating the proposed CF RS for groups of users, we will use the following quality measures: 1) accuracy and coverage, to measure the quality of the predictions; 2) precision and recall, to measure the quality of the recommendations.

The experiments have been performed using test ratings. The ratings have been removed from the training process and have not been used either to obtaine the KNN or to compute the predictions.

We define $\tilde{r}_{u,i}$ as the test rating of the user *u* to the item *i*.

Let T_G the set of items with a test rating of each user of the group *G* ($\tilde{r}_{u,i} \neq \bullet$) and with a not null prediction value ($p_{G,i} \neq \bullet$):

$$T_G = \left\{ i \in I \mid p_{G,i} \neq \bullet \wedge \forall u \in G : \tilde{r}_{u,i} \neq \bullet \right\} \tag{4.1}$$

Let MAE_G the mean absolute error of the predictions performed to the group *G*:

$$MAE_G = \frac{1}{\# T_G} \sum_{i \in T_G} \left(\frac{1}{\# G} \sum_{u \in G} \left| \tilde{r}_{u,i} - p_{G,i} \right| \right) \tag{4.2}$$

Let $accuracy_G$ be the inverse of the MAE_G:

$$accuracy_G = \frac{1}{MAE_G} \tag{4.3}$$

Let $coverage_G$ be the percentage of items for which a prediction for the group can be calculated over all items not rated by any user in the group *G*.

$$coverage_G = \frac{\# \left\{ i \notin I_G \mid p_{G,i} \neq \bullet \right\}}{\# \left(I - I_G \right)} \tag{4.4}$$

To compute the precision and recall, we must redefine the set of *N* items recommended to group *G*. We define $\tilde{Z}_G$ as the set of items with a test rating of each member of the group G that has been recommended to the group G. The following expressions must be verified:

$$\tilde{Z}_G \subseteq T_G \tag{4.5}$$

$$\# \tilde{Z}_G = \min\left(N, \# T_G\right) \tag{4.6}$$

$$\forall x \in \tilde{Z}_G, \forall y \notin \tilde{Z}_G : p_{G,x} \geq p_{G,y} \tag{4.7}$$

Let *precision_u* be the proportion of items that have been recommended and have been rated positively by the user *u* with respect the total number of items recommended.

$$precision_u = \frac{\# \left\{ i \in \tilde{Z}_G \mid \tilde{r}_{u,i} \in P \right\}}{\# \tilde{Z}_G} \tag{4.8}$$

Where *P* is the set of positive ratings. For example, if the worst rating is 1 and the best rating is 5, we can define *P* as *P* = [4..5].

Let $precision_G$ be the average precision of the group *G*:

$$precision_G = \frac{1}{\# G} \sum_{u \in G} precision_u \tag{4.9}$$

Let $recall_u$ be the proportion of items that have been recommended and have been rated positively by the user *u* with respect the total number of items that the user have rated positively.

$$recall_u = \frac{\# \left\{ i \in \tilde{Z}_G \mid \tilde{r}_{u,i} \in P \right\}}{\# \left\{ i \in T_G \mid \tilde{r}_{u,i} \in P \right\}} \tag{4.10}$$

Let $recall_G$ be the average recall of the group *G*:

$$recall_G = \frac{1}{\# G} \sum_{u \in G} recall_u \tag{4.11}$$

4.3. Experiments Overview

Our experiments will be carried out using the MovieLens and Netflix databases. The main information about the data of these databases is shown in **Table 4.1**.

	MovieLens	Netflix
Number of users	6,040	480,189
Number of items	3,706	17,770
Number of ratings	1,000,209	100,480,507
Rating scale	1 - 5	1 – 5

Table 4.1. Main information about the databases used on the experiments.

		Experiment 1	Experiment 2a	Experiment 2b	Experiment 3
Database		MovieLens	MovieLens	Netflix	MovieLens
Test items percent		35%	35%	25%	35%
Number of tested groups		1200	1200	48000	1200
Groups size		2-8	2-8	2-8	2-8
Number of neighbors (precision and recall)		200	200	300	300
Relevant threshold (precision and recall)		5	5	5	5
Relevant rates / Not relevant rates (SING)		4, 5 / 1, 2, 3	4, 5 / 1, 2, 3	4, 5 / 1, 2, 3	4, 5 / 1, 2, 3
Number of recommendations		1-10	1-10	1-10	10
Graphs	Prediction quality	Figure 4.2	Figure 4.4	Figure 4.6	Figure 4.8
	Recomm. quality	Figure 4.3	Figure 4.5	Figure 4.7	

Table 4.2. Main parameters used in the experiments.

In the experiments, the groups of users are generated randomly based on the users of the database. We also establish sets of test and training items on which to calculate the quality measures of the RS. **Table 4.2** includes the main parameters used in the experiments, as well as the figures and graphs containing the results. **Figure 4.1** contains the experiments will be carried out.

The main experiment (experiment 2) compares the method presented in this paper (UGSM) with the current state of the art alternatives set out in **Figure 2.1**. The baseline alternatives are showed into **Figure 4.1**. For the first alternative we will use the method proposed by Batrunas, Makcinskas, and Ricci (2010) (from now on RANK); however, method RANK is conceived solely to obtain recommendations and not to obtain predictions, therefore, two of the proposed quality measures, accuracy and coverage, cannot be used on it. For the second alternative we will use the method proposed by Berkovsky and Freyne (2010) (from now on PER). For the third alternative we will use the best result obtained with ERRS, Bobadilla, Ortega, Hernando, and Bernal (2012b), in experiment 1.

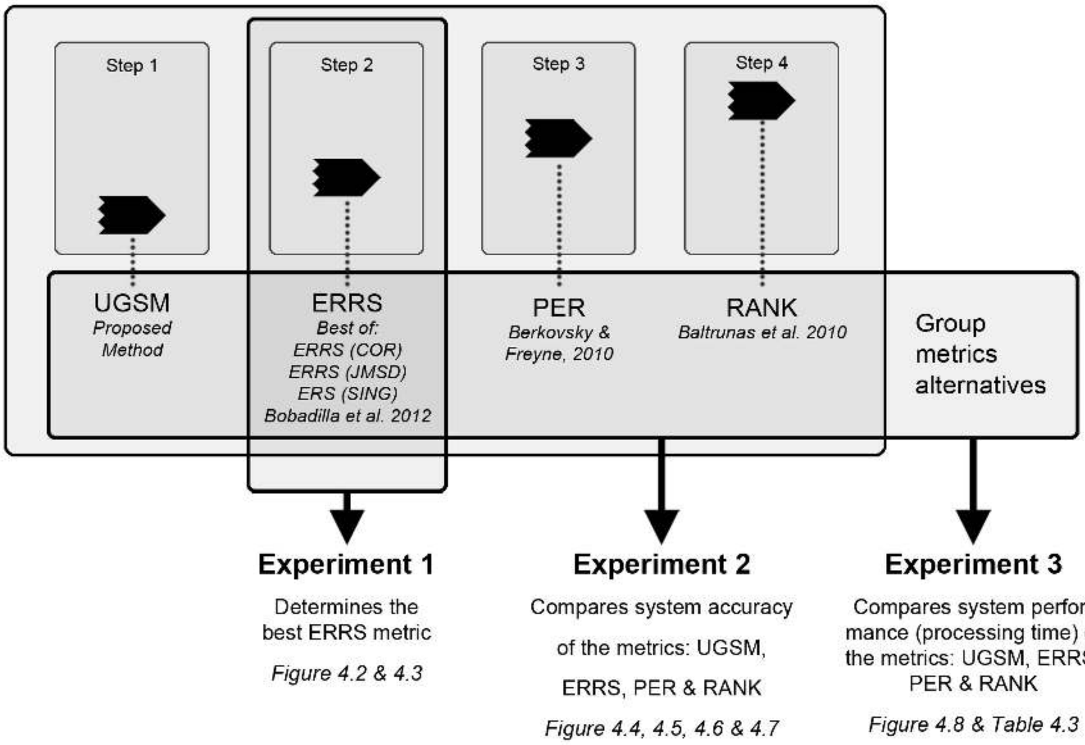


Figure 4.1. Experiments overview.

Before running experiment 2, we must determine the similarity metric that achieves the best result when applied to the ERRS method. Experiment 1 tests ERRS in combination with: Pearson Correlation (COR), the main reference in the CF field; JMSD (Bobadilla, Serradilla, & Bernal, 2010), a metric that provides great accuracy;

and SING (Bobadilla, Ortega, & Hernando, 2012), a metric that provides a good balance between accuracy and coverage.

Finally, experiment 3 runs a system performance (time consuming) comparison. We measure the average time to provide a set of recommendations to a group of users running different group recommendation algorithms. The problem of obtaining recommendations for groups of users is computationally more expensive than obtaining recommendations for a single user; therefore, as the size of the group increases, the calculation time required to achieve a recommendation for the group will be greater. The hypothesis of this experiment is that the sooner the users are unified (n users → 1 group) the less time is required to obtain the recommendations. We will use **Figure 4.1** as a base to check the times: for the first case we will test RANK using as the rank aggregation Borda count, for the second case we will use PER, for the third case we will use ERRS, and for the fourth case we will use UGSM.

4.4. Experiment 1: ERRS Baseline

Figure 4.2 and **Figure 4.3** shows the results of experiment 1. We can observe that: ERRS (JMSD) is the method that gives the best results as regards accuracy; ERRS (SING) is the one that provides the best results as regards coverage, precision and recall; in addition, it obtains good accuracy results. ERRS (COR) offers significantly worse results than the other alternatives.

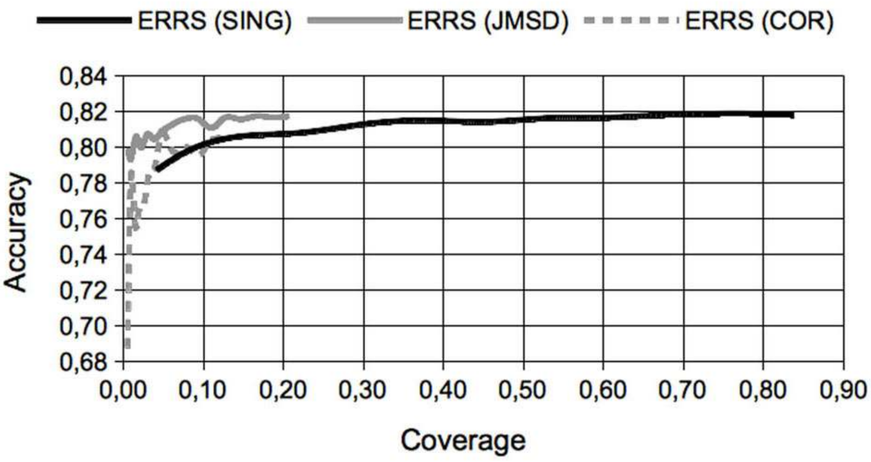


Figure 4.2. System accuracy and coverage comparisons using different CF similarity metrics with the ERRS method in MovieLens.

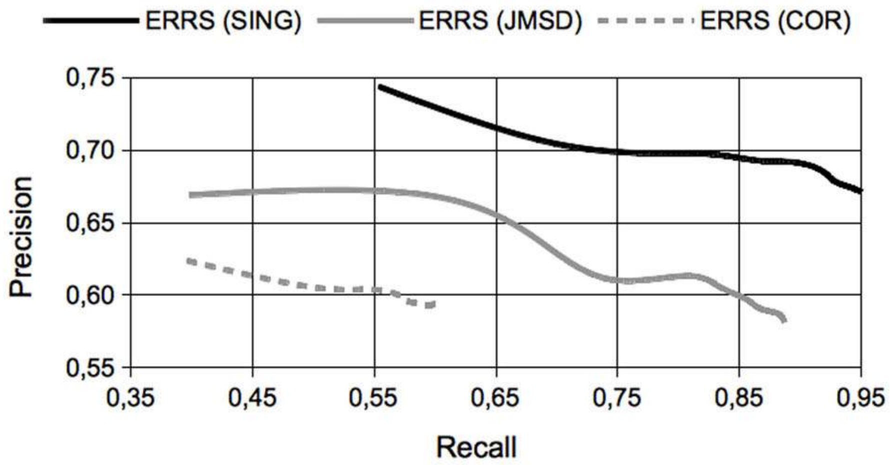


Figure 4.3. System precision and recall comparisons using different CF similarity metrics with the ERRS method in MovieLens.

In general, the method that provides the best average quality is ERRS (SING), and therefore, it will be the one we select as the baseline for the following experiments.

4.5. Experiment 2: Accuracy Test

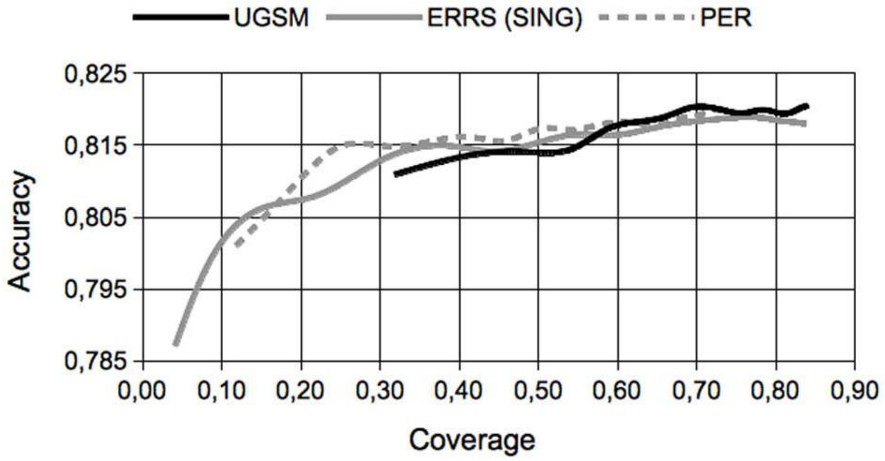


Figure 4.4. System accuracy and coverage comparisons using as recommendation methods: UGSM, ERRS (SING) and PER. MovieLens dataset used.

Figure 4.4 and **Figure 4.5** show the results of experiment 2a, in which I compare UGSM, PER, RANK and ERRS (SING) in MovieLens. We can observe that, as regards accuracy/coverage, the proposed method offers the best balance between quality and quantity of recommendations. As regards precision/recall, the results obtained by UGSM are similar to those obtained using PER and RANK, and higher than those obtained when we use ERRS (SING).

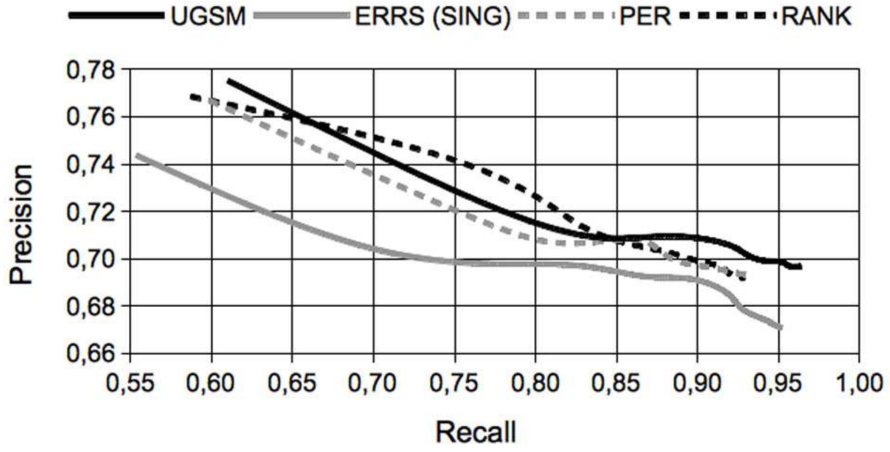


Figure 4.5. System precision and recall comparisons using as recommendation methods: UGSM, ERRS (SING), PER and RANK. MovieLens dataset used.

Figure 4.6 and **Figure 4.7** show the results of experiment 2b, in which I compare UGSM, PER, RANK and ERRS (SING) in Netflix. We can observe that, as regards the accuracy/coverage, the proposed method offers lower accuracy in the predictions but much more coverage than the two reference methods. As regards the precision/recall, the proposed method is slightly worse than PER and RANK and significantly better than ERRS (SING).

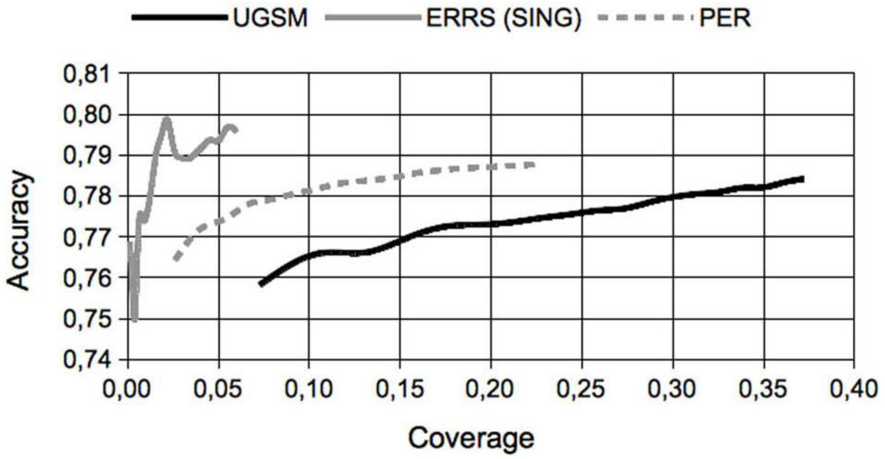


Figure 4.6. System accuracy and coverage comparisons using as recommendation methods: UGSM, ERRS (SING) and PER. Netflix dataset used.

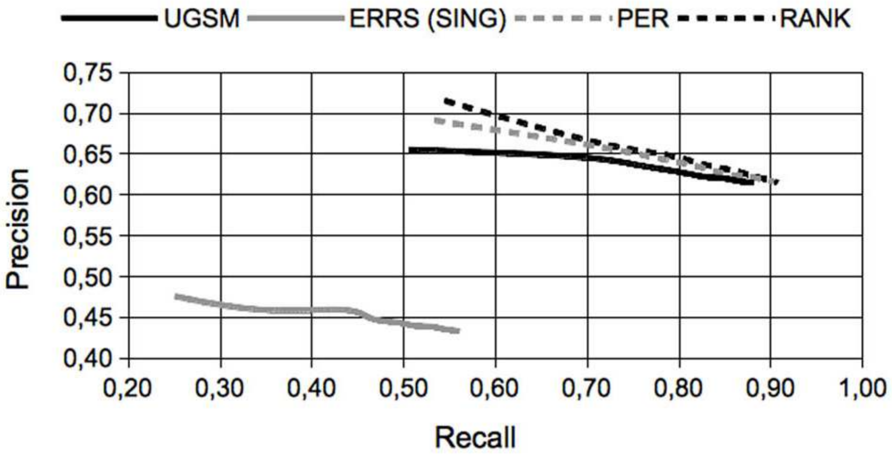


Figure 4.7. System precision and recall comparisons using as recommendation methods: UGSM, ERRS (SING), PER and RANK. Neflix dataset used.

4.6. Experiment 3: Performance Test

Figure 4.8 shows the results of experiment 3. By placing the results in order of calculation time (from fastest to slowest) we obtain: UGSM < ERRS < PER < RANK. These results coincide with the hypothesis set out in this paper: the unification of the group in early stages of the CF favors the speed in obtaining recommendations. Furthermore, we can observe that the recommendation obtaining time for the group increases lineally with the size of the group, which favors the scalability of the method.

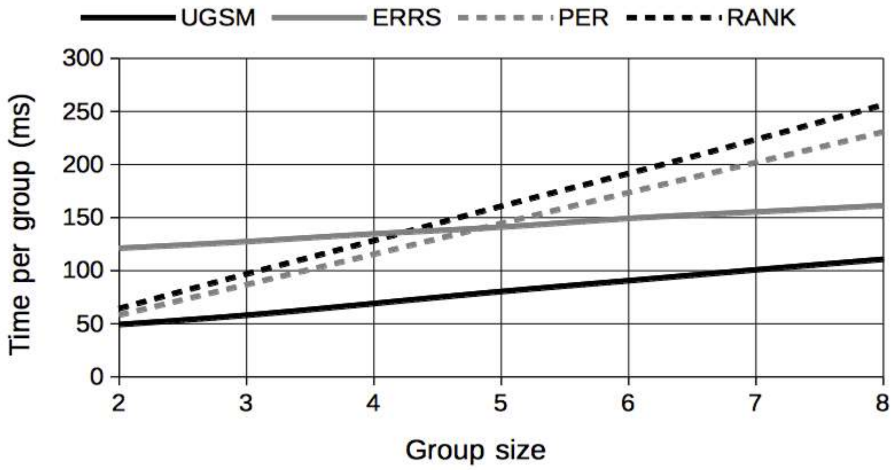


Figure 4.8. System performance comparison using different group recommendation algorithms based on CF in MovieLens.

I have performed T-tests that confirm that our method improves the speed in obtaining recommendations. Indeed, for each size of users group, I will compare the speed of my method with the speed of each method proposed by other authors. In my experiments, the null hypothesis considers that the speed of my method is the same than the speed of the method proposed by other authors, while my alternative hypothesis considers that the speed of my method is better than the speed of the method described by other authors. As we will see, my experiments prove that there is statistic evidence for rejecting this null hypothesis and accepting the alternative hypothesis. The experiments performed consist of the following steps:

(1) For each group size, #G, I have randomly obtained 100 groups of #G users.

(2) For each of these groups of users, I calculate the time that both my method and the methods described by other authors take to make the recommendations.

(3) For each method described by other authors, I perform a paired t-test comparing its speed with the speed of my method (we calculate the p-value associated to this t-test).

Group Size	UGSM-ERRS	UGSM-PER	UGSM-RANK
2	1.01839E-79	1.55241E-12	1.94293E-33
3	8.5768E-142	1.63384E-72	3.158E-123
4	1.2557E-138	1.1418E-143	3.5824E-151
5	4.3187E-102	3.5304E-117	1.3759E-130
6	3.3477E-101	8.676E-148	9.7014E-153
7	3.4037E-101	2.6485E-174	3.3329E-197
8	1.8211E-39	6.53943E-94	1.04222E-90

Table 4.3. p-values of all the t-tests carried out to compare the performance of the different group recommendation algorithms in MovieLens.

In **Table 4.3**, we present the p-values of all the t-tests performed. As may be seen, the p-values obtained are very low. Indeed they are all below the typical significance levels 0.05 and 0.01. Consequently, we can conclude that there is statistic evidence for rejecting the null hypothesis and accepting the alternative hypothesis, which means, my method improves significantly the speed of the methods presented by other authors.

5. FUTURE WORK

This PhD thesis are focus on understanding, analyzing and improving the different alternatives to perform group of user recommendations using collaborative filtering techniques. Although the experimental results are positive and the proposed method has significantly reduced the time required to compute the recommendations to groups of users, different research lines can improve the experimental results presented here.

The proposed method presents a new way to perform group recommendations using a similarity metric that is applied on the first step of the collaborative filtering algorithm. Similarity metrics has been widely studied on traditional memory based collaborative filtering. Developing an alternative group-based similarity metric could increase the accuracy of the results without deteriorate the performance. Add properties as the non-numeric information of the ratings (Bobadilla, Serradilla, & Bernal, 2010) or the singularity of them (Bobadilla, Ortega, & Hernando, 2012) have reported great benefits in previous works.

Cold start problem also affects to group recommender systems. Include users with few ratings in a group of users could decrease the accuracy of the recommendations, because it is more difficult to get the group's preferences if the sparsity of group’s data is high. Novel items recommendation is also a challenge of group recommender systems: if only a few users have rated the novel item, it is highly probable that the item will not be recommended. Study the impact of cold start situation in group recommender systems is an interesting research work.

Current trends in RS area include social factors to the recommendations process (followers, trust networks, implicit ratings, etc.). This fact usually increases the accuracy of the recommendations but decreases the overall performance of the system, because more data are needed to compute the recommendations. Furthermore, include additional information to the RS requieres more specific methods than traditional RS which only works with ratings information. Some authors (Christensen & Schiaffino, 2013; Christensen & Schiaffino, 2014) have studied how to include social factors to model based collaborative filtering. However, these factors have not been included on memory based collaborative filtering yet. A new powerful research line is to analyze how to incorporate these factors to memory based CF.

These factors could be included in one (or many) of the main four stages of the algorithm: similarity computation, neighborhood construction, rating aggregation and recommendation phase. Furthermore, the impact of incorporate different type of sociological factors could be analyzed.

The proposed method uses weighted average aggregation to compute predictions over the items that the group unknowns. Nevertheless, deviation from mean aggregation has presented better results on traditional memory based collaborative filtering. This aggregation method requires the user rating average to work. We have replaced the active user for a group of users, so we cannot use this method to compute the predictions. Moreover, we cannot use the rating average of the group because this value is not too representative for large size groups. One possible solution is to replace the user rating average for the group rating average of similar items to the active one. It is necessary that we precompute the items similarities in order to calculate the predictions efficiently. This similarity can be computed using the well known item to item collaborative filtering methods.

Nowadays, collaborative filtering researches are focused on the usage probabilistic models based on matrix factorization. These methods report better experimental results: they decrease the sparsity problem and they speed up the recommendation process if the models are built. We can incorporate group recommendations for these models in two different stages: before matrix factorization or after matrix factorization.

If we want to add group recommendation capabilities before matrix factorization process, it is necessary to build a virtual user that represents the group. This virtual user will be composed of the rating information of each user of the group. To build this virtual user, or the rating average can be used or a weighted model depending on the influence of each user on the group can be proposed. When the virtual user is built, the factorization process can be carried out using the resulting matrix of merging the virtual user and each user of the recommender system that does not belong to the group. After that, we can compute recommendations in a similar way to the case of individual user. The main problem of these methods is that is very expensive computationally, because it is necessary a continuous update of the model. Using techniques as folding-in this trouble can be minimized.

If we want to add group recommendation capabilities after matrix factorization process, we can presuppose that we have factorized the rating matrix into two matrices: items-factor matrix and users-factors matrix. Using the users-factors matrix we can perform the recommendations with the method proposed in this PhD thesis. We compute the similarity between the group of users and each user of the recommender system using the users-factors matrix instead of the original ratings matrix. Once the similarities are computed, the kNN set can be built and the recommendations can be computed. How to determine the similarity between the group and the users is an interesting research line because different methods can be applied.

Finally, we can modify the probabilistic model that represents the factorization process. The most used model is PMF (Mnih & Salakhutdinov, 2007). To extend the model it is necessary to add it new probabilistic variables that represents the group of users and the interaction of the group with the items. The output of the model would be a set of probabilistic distributions that allow us to compute the item predictions of the group without aggregate the individual users information.

6. CONCLUSIONS

In collaborative filtering based recommender systems, when we wish to make recommendations to groups of users, there are four basic levels on which we can act to unify data of the users in the group with the objective of obtaining the data of the group of users: similarity metric, establishing the neighborhood, prediction phase, and determining recommended items.

Each of the four previous levels shows an aggregation approach of individual users in a group of users. The experimental results show fairly unrepresentative differences in the quality of the recommendations obtained on applying the different aggregation approaches; however, the system performance notably deteriorates as the moment the aggregation approach is applied is delayed.

The group recommendation system performance is related to the moment at which the aggregation phase is performed. This is due to the fact that until that moment there are as many differentiated processes for the execution of collaborative filtering algorithms as users that belong to the group. From the aggregation, the different collaborative filtering processes associated to each of the users disappear and give rise to a single collaborative filtering process for the group.

In short, it is advisable to include the aggregation of the users of the group in the actual similarity metric used in the collaborative filtering: the system performance significantly improves and the system accuracy undergoes barely appreciable changes. In this work I have provided a similarity metric that implements this concept.

REFERENCES

Acilar, A. M., & Arslan, A. (2008). A collaborative filtering method based on artificial immune network. *Expert Systems with Applications, 36*(4), 8324– 8332.

Adomavicius, G., & 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), 734–749.

Adomavicius, G., & Zhang, J. (2010). On the stability of recommendations algorithms. In *ACM Conference on Recommender Systems* (pp. 47–54).

Ahn, H.J. (2008). A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences, 178*, 37–51.

Al-Shamri, M.Y.H., & Bharadwaj, K.K. (2008). Fuzzy-genetic approach to recommender systems based on a novel hybrid user model. *Expert Systems with Applications, 35*(3), 1386–1399.

Alonso, S., Cabrerizo, F.J., Chiclana, F., Herrera, F., & Herrera-Viedma, E. (2009). Group decision making with incomplete fuzzy linguistic preference relations. *International Journal of Intelligent Systems, 24*, 201–222.

Amer-Yahia, S., Roy, S. B., Chawlat, A., Das, G., & Yu, C. (2009). Group recommendation: Semantics and efficiency. *Proceedings of the VLDB Endowment, 2*(1), 754-765.

Ansari, A., Essegaier, S., & Kohli, R. (2000). Internet recommendation systems. *Journal of Marketing Research, 37*(3), 363–375.

Antunes, P., Herskovic, V., Ochoa, S.F., & Pino, J.A. (2012). Structuring dimensions for collaborative systems evaluation. *ACM Computing Surveys, 44*(2), article 8.

Arazy, O., Kumar, N., & Shapira, B. (2009). Improving Social Recommender Systems. *Journal IT Professional, 11*(4), 31–37.

Ardissono, L., Goy, A., Petrone, G., Segnan, M., & Torasso, P. (2003). INTRIGUE: Personalized recommendation of tourist attractions for desktop and handset devices. *Applied Artificial Intelligence, 17*(8-9), 687–714.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval* (Vol. 463). New York: ACM press.

Balabanovic, M., & Shoham, Y. (1997). Content-based, collaborative recommendation. *Communications of the ACM, 40*(3), 66–72.

Baltrunas, L., Makcinskas, T., & Ricci, F. (2010, September). Group recommendations with rank aggregation and collaborative filtering. In*Proceedings of the fourth ACM conference on Recommender systems* (pp. 119-126). ACM.

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

Basu, C., Hirsh, H., & Cohen, W. (1998, July). Recommendation as classification: Using social and content-based information in recommendation. In *AAAI/IAAI* (pp. 714-720).

Bengio, Y., & Grandvalet, Y. (2004). No unbiased estimator of the variance of k-fold cross-validation. *The Journal of Machine Learning Research*, *5*, 1089-1105.

Berkovsky, S., & Freyne, J. (2010, September). Group-based recipe recommendations: analysis of data aggregation strategies. In *Proceedings of the fourth ACM conference on Recommender systems* (pp. 111-118). ACM.

Bilge, A., & Polat, H. (2012). An improved privacy-preserving DWT-based collaborative filtering scheme. *Experts Systems with Applications, 39*(3), 3654–3841.

Bilgic, M., & Mooney, R. J. (2005, January). Explaining recommendations: Satisfaction vs. promotion. In Beyond Personalization Workshop, IUI (Vol. 5).

Billsus, D., & Pazzani, M. J. (1999, April). A personal news agent that talks, learns and explains. In Proceedings of the third annual conference on Autonomous Agents (pp. 268-275). ACM.

Billsus, D., & Pazzani, M. J. (2000). User modeling for adaptive news access.User modeling and user-adapted interaction, 10(2-3), 147-180.

Billsus, D., Pazzani, M., & Chen, J. (2002). A learning agent for wireless news access. In Proceedings of the International Conference on Intelligent User Interfaces (pp. 33–36).

Biuk-Aghai, R. P., Fong, S., & Si, Y. W. (2008, December). Design of a recommender system for mobile tourism multimedia selection. In Internet Multimedia Services Architecture and Applications, 2008. IMSAA 2008. 2nd International Conference on (pp. 1-6). IEEE.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, *3*, 993-1022.

Bobadilla, J., & Serradilla, F. (2009, January). The effect of sparsity on collaborative filtering metrics. In Proceedings of the Twentieth Australasian Conference on Australasian Database-Volume 92 (pp. 9-18). Australian Computer Society, Inc..

Bobadilla, J., Hernando, A., Ortega, F., & Bernal, J. (2011). A framework for collaborative filtering recommender systems. Expert Systems with Applications,38(12), 14609-14623.

Bobadilla, J., Hernando, A., Ortega, F., & Gutiérrez, A. (2012). Collaborative filtering based on significances. Information Sciences, 185(1), 1-17.

Bobadilla, J., Ortega, F., & Hernando, A. (2012). A collaborative filtering similarity measure based on singularities. Information Processing & Management, 48(2), 204-217.

Bobadilla, J., Ortega, F., Hernando, A., & Alcalá, J. (2011). Improving collaborative filtering recommender system results and performance using genetic algorithms. Knowledge-based systems, 24(8), 1310-1316.

Bobadilla, J., Ortega, F., Hernando, A., & Arroyo, A. (2013). A balanced memory-based collaborative filtering similarity measure. International Journal of Intelligent Systems, 27(10), 939–946.

Bobadilla, J., Ortega, F., Hernando, A., & Bernal, J. (2012a). A collaborative filtering approach to mitigate the new user cold start problem. Knowledge-Based Systems, 26, 225-238.

Bobadilla, J., Ortega, F., Hernando, A., & Bernal, J. (2012b). Generalization of recommender systems: Collaborative filtering extended to groups of users and restricted to groups of items. Expert Systems with Applications, 39(1), 172-186.

Bobadilla, J., Ortega, F., Hernando, A., & Glez-de-Rivera, G. (2013). A similarity metric designed to speed up, using hardware, the recommender systems k-nearest neighbors algorithm. *Knowledge-Based Systems*, *51*, 27-34.

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, *46*, 109-132.

Bobadilla, J., Serradilla, F., & Bernal, J. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. Knowledge-Based Systems, 23(6), 520-528.

Bobadilla, J., Serradilla, F., & Hernando, A. (2009). Collaborative filtering adapted to recommender systems of e-learning. Knowledge-Based Systems, 22(4), 261-265.

Bogers, T., & Van den Bosch, A. (2009). Collaborative and content-based filtering for item recommendation on social bookmarking websites. Submitted to CIKM, 9.

Boratto, L., & Carta, S. (2011). State-of-the-art in group recommendation and new approaches for automatic identification of groups. In *Information Retrieval and Mining in Distributed Environments* (pp. 1-20). Springer Berlin Heidelberg.

Boratto, L., & Carta, S. (2014a, June). Modeling the Preferences of a Group of Users Detected by Clustering: a Group Recommendation Case-Study. In*Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)* (p. 16). ACM.

Boratto, L., & Carta, S. (2014b). Impact of Content Novelty on the Accuracy of a Group Recommender System. In *Data Warehousing and Knowledge Discovery*(pp. 159-170). Springer International Publishing.

Borràs, J., Moreno, A., & Valls, A. (2014). Intelligent tourism recommender systems: A survey. *Expert Systems with Applications, 41*(16), 7370-7389.

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

Bridge, D., Göker, M. H., McGinty, L., & Smyth, B. (2005). Case-based recommender systems. *The Knowledge Engineering Review*, *20*(03), 315-320.

Burke, R. (2000a). A case-based reasoning approach to collaborative filtering. In*Advances in Case-Based Reasoning* (pp. 370-379). Springer Berlin Heidelberg.

Burke, R. (2000b). Knowledge-based recommender systems. *Encyclopedia of library and information systems*, *69*(Supplement 32), 175-186.

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

Cacheda, F., Carneiro, V., Fernández, D., & Formoso, V. (2011). Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web (TWEB)*, *5*(1), 2.

Caizer, S., & Aickelin, U. (2005). A recommender system based on idiotypic artificial immune networks. *Journal of Mathematics, Models and Algorithms. v4 i2*, 181-198.

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

Candillier, L., Meyer, F., & Boullé, M. (2007). Comparing state-of-the-art collaborative filtering systems. In *Machine Learning and Data Mining in Pattern Recognition* (pp. 548-562). Springer Berlin Heidelberg.

Carrer-Neto, W., Hernández-Alcaraz, M. L., Valencia-García, R., & García-Sánchez, F. (2012). Social knowledge-based recommender system. Application to the movies domain. *Expert Systems with Applications*, *39*(12), 10990-11000.

Castro-Schez, J. J., Miguel, R., Vallejo, D., & López-López, L. M. (2011). A highly adaptive recommender system based on fuzzy logic for B2C e-commerce portals. *Expert Systems with Applications*, *38*(3), 2441-2454.

Chao, D. L., Balthrop, J., & Forrest, S. (2005, November). Adaptive radio: achieving consensus using negative preferences. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work* (pp. 120-123). ACM.

Chen, T., & He, L. (2009, June). Collaborative filtering based on demographic attribute vector. In *Future Computer and Communication, 2009. FCC'09. International Conference on* (pp. 225-229). IEEE.

Chen, Y. L., Cheng, L. C., & Chuang, C. N. (2008). A group recommendation system with consideration of interactions among group members*. Expert systems with applications, 34*(3), 2082-2090.

Chirita, P. A., Nejdl, W., & Zamfir, C. (2005, November). Preventing shilling attacks in online recommender systems. In *Proceedings of the 7th annual ACM international workshop on Web information and data management* (pp. 67-74). ACM.

Cho, J., Kwon, K., & Park, Y. (2009). Q-rater: A collaborative reputation system based on source credibility theory. *Expert Systems with Applications*,*36*(2), 3751-3760.

Choi, K., & Suh, Y. (2013). A new similarity function for selecting neighbors for each target item in collaborative filtering. *Knowledge-Based Systems*, *37*, 146-153.

Choi, K., Yoo, D., Kim, G., & Suh, Y. (2012). A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electronic Commerce Research and Applications*, *11*(4), 309-317.

Christakou, C., Vrettos, S., & Stafylopatis, A. (2007). A hybrid movie recommender system based on neural networks. *International Journal on Artificial Intelligence Tools*, *16*(05), 771-792.

Christensen, I. A., & Schiaffino, S. (2011). Entertainment recommender systems for group of users. *Expert Systems with Applications*, *38*(11), 14127-14135.

Christensen, I. A., & Schiaffino, S. (2013, November). Matrix Factorization in Social Group Recommender Systems. In *Artificial Intelligence (MICAI), 2013 12th Mexican International Conference on* (pp. 10-16). IEEE.

Christensen, I. A., & Schiaffino, S. (2014). Social influence in group recommender systems. *Online Information Review*, *38*(4), 5-5.

Chulyadyo, R., & Leray, P. (2014). A Personalized Recommender System from Probabilistic Relational Model and Users' Preferences. *Procedia Computer Science, 35*, 1063-1072.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. (1999, August). Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR workshop on recommender systems*(Vol. 60).

Cohen, W. (1995). Fast effective rule induction. In Proceedings of the Twelfth International Conference on Machine Learning (pp. 115–123).

Colombo-Mendoza, L. O., Valencia-García, R., Rodríguez-González, A., Alor-Hernández, G., & Samper-Zapater, J. J. (2014). RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes. *Expert Systems with Applications, 42*(3), 1202-1222.

Condliff, M. K., Lewis, D. D., Madigan, D., & Posse, C. (1999, August). Bayesian mixed-effects models for recommender systems. In ACM SIGIR (Vol. 99, pp. 23-30).

Costa-Montenegro, E., Barragáns-Martínez, A. B., & Rey-López, M. (2012). Which App? A recommender system of applications in markets: Implementation of the service for monitoring users' interaction. Expert systems with applications, 39(10), 9367-9375.

Dao, T. H., Jeong, S. R., & Ahn, H. (2012). A novel recommendation model of location-based advertising: Context-Aware Collaborative Filtering using GA approach. Expert Systems with Applications, 39(3), 3731-3739.

De Gemmis, M., Lops, P., Semeraro, G., & Basile, P. (2008, October). Integrating tags in a semantic content-based recommender. In *Proceedings of the 2008 ACM conference on Recommender systems* (pp. 163-170). ACM.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *JASIS*, *41*(6), 391-407.

DuBois, T., Golbeck, J., Kleint, J., & Srinivasan, A. (2009). Improving recommendation accuracy by clustering social networks with trust.Recommender Systems & the Social Web, 532, 1-8.

Ekstrom, M. A., Bjornsson, H. C., & Nass, C. I. (2005). A reputation mechanism for business-to-business electronic commerce that accounts for rater credibility.Journal of Organizational Computing and Electronic Commerce, 15(1), 1-18.

Felfernig, A., & Burke, R. (2008, August). Constraint-based recommender systems: technologies and research issues. In *Proceedings of the 10th international conference on Electronic commerce* (p. 3). ACM.

Gao, L. Q., & Li, C. (2008). Hybrid personalizad recommended model based on genetic algorithm. In *Int. conf. on wireless commun. netw. and mob. computing*(pp. 9215-9218).

Gao, M., Wu, Z., & Jiang, F. (2011). Userrank for item-based collaborative filtering recommendation. *Information Processing Letters*, *111*(9), 440-446.

Garcia, I., & Sebastia, L. (2014). A negotiation framework for heterogeneous group recommendation. *Expert Systems with Applications, 41*(4), 1245-1261.

Garcia, I., Sebastia, L., & Onaindia, E. (2011). On the design of individual and group recommender systems for tourism. *Expert Systems with Applications*,*38*(6), 7683-7692.

Garcia, R., & Amatriain, X. (2010, October). Weighted content based methods for recommending connections in online social networks. In *Workshop on Recommender Systems and the Social Web* (pp. 68-71).

Gartrell, M., Xing, X., Lv, Q., Beach, A., Han, R., Mishra, S., & Seada, K. (2010, November). Enhancing group recommendation by incorporating social relationship interactions. In *Proceedings of the 16th ACM international conference on Supporting group work* (pp. 97-106). ACM.

Gavalas, D., & Kenteris, M. (2011). A web-based pervasive recommendation system for mobile tourist guides. *Personal and Ubiquitous Computing*, *15*(7), 759-770.

Gedikli, F., & Jannach, D. (2010, September). Rating items by rating tags. In*Proceedings of 2nd Workshop on Recommender Systems and the Social Web at ACM RecSys 2010* (pp. 25-32).

Gedikli, F., Jannach, D., & Ge, M. (2014). How should I explain? A comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies, 72*(4), 367-382.

Gemmell, J., Schimoler, T., Ramezani, M., Christiansen, L., & Mobasher, B. (2009). Improving folkrank with item-based collaborative filtering.*Recommender Systems & the Social Web*.

Gemmell, J., Schimoler, T., Ramezani, M., Christiansen, L., & Mobasher, B. (2010). Resource recommendation for social tagging: A multi-channel hybrid approach. *Recommender Systems & the Social Web, Barcelona, Spain*.

George, T., & Merugu, S. (2005, November). A scalable collaborative filtering framework based on co-clustering. In *Data Mining, Fifth IEEE International Conference on* (pp. 4-pp). IEEE.

Golbeck, J., & Kuter, U. (2009). The ripple effect: change in trust and its impact over a social network. In *Computing with Social Trust* (pp. 169-181). Springer London.

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, *4*(2), 133-151.

Good, N., Schafer, J. B., Konstan, J. A., Borchers, A., Sarwar, B., Herlocker, J., & Riedl, J. (1999, July). Combining collaborative filtering with personal agents for better recommendations. In *AAAI/IAAI* (pp. 439-446).

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, *101*(Suppl 1), 5228-5235.

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

Herlocker, J. L., Konstan, J. A., & Riedl, J. (2000, December). Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work* (pp. 241-250). ACM.

Herlocker, J. L., Konstan, J. A., & Riedl, J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information retrieval*, *5*(4), 287-310.

Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999, August). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 230-237). ACM.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, *22*(1), 5-53.

Hernández, F., & Gaudioso, E. (2008). Evaluation of recommender systems: A new approach. *Expert Systems with Applications*, *35*(3), 790-804.

Hernando, A., Bobadilla, J., Ortega, F., & Gutiérrez, A. (2013). Trees for explaining recommendations made through collaborative filtering. *Information Sciences*, *239*, 1-17.

Hernando, A., Bobadilla, J., Ortega, F., & Tejedor, J. (2013). Incorporating reliability measurements into the predictions of a recommender system.*Information Sciences*, *218*, 1-16.

Hernando, A., Moya, R., Ortega, F., & Bobadilla, J. (2014). Hierarchical graph maps for visualization of collaborative recommender systems. *Journal of Information Science*, *40*(1), 97-106.

Ho, Y., Fong, S., & Yan, Z. (2007). A Hybrid GA-based Collaborative Filtering Model for Online Recommenders. In *International Conference on e-Business* (pp. 200-203).

Hofmann, T. (1999, July). Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence* (pp. 289-296). Morgan Kaufmann Publishers Inc..

Hossain, L., & Fazio, D. (2009). The social networks of collaborative process.*The Journal of High Technology Management Research*, *20*(2), 119-130.

Hu, R., & Pu, P. (2011, October). Enhancing collaborative filtering systems with personality information. In *Proceedings of the fifth ACM conference on Recommender systems* (pp. 197-204). ACM.

Hu, Y., Koren, Y., & Volinsky, C. (2008, December). Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on* (pp. 263-272). IEEE.

Huang, Y. P., Chuang, W. P., Ke, Y. H., & Sandnes, F. E. (2008). Using back-propagation to learn association rules for service personalization. *Expert Systems with Applications*, *35*(1), 245-253.

Huang, Z., Zeng, D., & Chen, H. (2007). A comparison of collaborative-filtering recommendation algorithms for e-commerce. *IEEE Intelligent Systems*, *22*(5), 68-78.

Hurley, N., & Zhang, M. (2011). Novelty and diversity in top-n recommendation--analysis and evaluation. *ACM Transactions on Internet Technology (TOIT)*,*10*(4), 14.

Hwang, C. S., Su, Y. C., & Tseng, K. C. (2010). Using genetic algorithms for personalized recommendation. In *Computational Collective Intelligence. Technologies and Applications* (pp. 104-112). Springer Berlin Heidelberg.

Jameson, A., & Smyth, B. (2007). Recommendation to groups. In *The adaptive web* (pp. 596-627). Springer Berlin Heidelberg.

Jannach, D. (2009). Fast computation of query relaxations for knowledge-based recommenders. *Ai Communications*, *22*(4), 235-248.

Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., & Stumme, G. (2007). Tag recommendations in folksonomies. In *Knowledge Discovery in Databases: PKDD 2007* (pp. 506-514). Springer Berlin Heidelberg.

Jeong, B., Lee, J., & Cho, H. (2009). User credit-based collaborative filtering.*Expert Systems with Applications*, *36*(3), 7309-7312.

Joachims, T. (1998). *Text categorization with support vector machines: Learning with many relevant features* (pp. 137-142). Springer Berlin Heidelberg.

Jøsang, A., Ismail, R., & Boyd, C. (2007). A survey of trust and reputation systems for online service provision. *Decision support systems*, *43*(2), 618-644.

Kaleli, C., & Polat, H. (2012). Privacy-preserving SOM-based recommendations on horizontally distributed data. *Knowledge-Based Systems*, *33*, 124-135.

Kang, J. H., & Lerman, K. (2013a, November). LA-CTR: A Limited Attention Collaborative Topic Regression for Social Media. In *AAAI*.

Kang, J., & Lerman, K. (2013b). Scalable Mining of Social Data using Stochastic Gradient Fisher Scoring. In *Proceedings of the CIKM workshop on Data-driven User Behavioral Modelling and Mining from Social Media (DUBMOD)*.

Kim, H. N., Alkhaldi, A., El Saddik, A., & Jo, G. S. (2011). Collaborative user modeling with user-generated tags for social recommender systems. *Expert Systems with Applications*, *38*(7), 8488-8496.

Kim, H. N., El-Saddik, A., & Jo, G. S. (2011). Collaborative error-reflected models for cold-start recommender systems. *Decision Support Systems*, *51*(3), 519-531.

Kim, H. N., Ji, A. T., Ha, I., & Jo, G. S. (2010). Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. *Electronic Commerce Research and Applications*, *9*(1), 73-83.

Kim, J. K., Kim, H. K., Oh, H. Y., & Ryu, Y. U. (2010). A group recommendation system for online communities. *International Journal of Information Management, 30*(3), 212-219.

Kim, J. W., Lee, B. H., Shaw, M. J., Chang, H. L., & Nelson, M. (2001). Application of decision-tree induction techniques to personalized advertisements on internet storefronts. *International Journal of Electronic Commerce*, *5*, 45-62.

Kim, K. J., & Ahn, H. (2005). Using a clustering genetic algorithm to support customer segmentation for personalized recommender systems. In *Artificial Intelligence and Simulation* (pp. 409-415). Springer Berlin Heidelberg.

Kim, K. J., & Ahn, H. (2008). A recommender system using GA K-means clustering in an online shopping market. *Expert systems with applications*, *34*(2), 1200-1209.

Kitisin, S., & Neuman, C. (2006, August). Reputation-based trust-aware recommender system. In *Securecomm and Workshops, 2006* (pp. 1-7). IEEE.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30-37.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30-37.

Koutrika, G., Bercovitz, B., & Garcia, H. (2009, June). FlexRecs: expressing and combining flexible recommendations. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data* (pp. 745-758). ACM.

Krulwich, B. (1997). Lifestyle finder: Intelligent user profiling using large-scale demographic data. *AI magazine*, *18*(2), 37.

Kwon, K., Cho, J., & Park, Y. (2009). Multidimensional credibility model for neighbor selection in collaborative recommendation. *Expert Systems with Applications*, *36*(3), 7114-7122.

Lam, S. K., & Riedl, J. (2004, May). Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web*(pp. 393-402). ACM.

Lam, X. N., Vu, T., Le, T. D., & Duong, A. D. (2008, January). Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication* (pp. 208-211). ACM.

Landia, N., & Anand, S. (2009). Personalised tag recommendation.*Recommender Systems & the Social Web, New York, NY, USA*, 83-86.

Lang, K. (1995). Newsweeder: Learning to filter netnews. In *In Proceedings of the Twelfth International Conference on Machine Learning*.

Lee, D. H., & Brusilovsky, P. (2009). Does Trust Influence Information Similarity?. *Recommender Systems & the Social Web*, 10.

Lee, M., Choi, P., & Woo, Y. (2002, January). A hybrid recommender system combining collaborative filtering with neural network. In *Adaptive Hypermedia and Adaptive Web-Based Systems* (pp. 531-534). Springer Berlin Heidelberg.

Lee, S. K., Cho, Y. H., & Kim, S. H. (2010). Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations. *Information Sciences*, *180*(11), 2142-2155.

Lerman, K. (2013). Social Informatics: Using Big Data to Understand Social Behavior. In *Handbook of Social Computation*. Springer.

Leung, C. W. K., Chan, S. C. F., & Chung, F. L. (2008). An empirical study of a cross-level association rule mining approach to cold-start recommendations.*Knowledge-Based Systems*, *21*(7), 515-529.

Li, Q., & Kim, B. M. (2003, October). Clustering approach for hybrid recommender system. In *Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on* (pp. 33-38). IEEE.

Li, Y. M., & Kao, C. P. (2009). TREPPS: A trust-based recommender system for peer production services. *Expert systems with applications*, *36*(2), 3263-3277.

Li, Y. M., Liao, T. F., & Lai, C. Y. (2012). A social recommender mechanism for improving knowledge sharing in online forums. *Information Processing & Management*, *48*(5), 978-994.

Loh, S., Lorenzi, F., Granada, R., Lichtnow, D., Wives, L. K., & de Oliveira, J. P. M. (2009). Identifying Similar Users by their Scientific Publications to Reduce Cold Start in Recommender Systems. In *Proceedings of the 5th International Conference on Web Information Systems and Technologies (WEBIST)* (pp. 593-600).

Luo, X., Xia, Y., & Zhu, Q. (2012). Incremental collaborative filtering recommender based on regularized matrix factorization. *Knowledge-Based Systems*, *27*, 271-280.

Luo, X., Xia, Y., & Zhu, Q. (2013). Applying the learning rate adaptation to the matrix factorization based collaborative filtering. *Knowledge-Based Systems*,*37*, 154-164.

Ma, H., Zhou, T. C., Lyu, M. R., & King, I. (2011). Improving recommender systems by incorporating social contextual information. *ACM Transactions on Information Systems (TOIS)*, *29*(2), 9.

Machanavajjhala, A., Korolova, A., & Sarma, A. D. (2011). Personalized social recommendations: accurate or private. *Proceedings of the VLDB Endowment*,*4*(7), 440-450.

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

Marinho, L. B., & Schmidt-Thieme, L. (2008). Collaborative tag recommendations. In *Data Analysis, Machine Learning and Applications* (pp. 533-540). Springer Berlin Heidelberg.

Martinez, L., Perez, L. G., & Barranco, M. J. (2009, June). Incomplete preference relations to smooth out the cold-start in collaborative recommender systems. In *Fuzzy Information Processing Society, 2009. NAFIPS 2009. Annual Meeting of the North American* (pp. 1-6). IEEE.

Martinez, L., Rodriguez, R. M., & Espinilla, M. (2009, September). REJA: A georeferenced hybrid recommender system for restaurants. In *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on* (Vol. 3, pp. 187-190). IET.

Massa, P., & Avesani, P. (2004). Trust-aware collaborative filtering for recommender systems. In *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE* (pp. 492-508). Springer Berlin Heidelberg.

Matyas, C., & Schlieder, C. (2009). A spatial user similarity measure for geographic recommender systems. In *GeoSpatial Semantics* (pp. 122-139). Springer Berlin Heidelberg.

McCarthy, K., Reilly, J., McGinty, L., & Smyth, B. (2004, August). Thinking positively-explanatory feedback for conversational recommender systems. In*Proceedings of the European Conference on Case-Based Reasoning (ECCBR-04) Explanation Workshop* (pp. 115-124).

McCarthy, K., Salamó, M., Coyle, L., McGinty, L., Smyth, B., & Nixon, P. (2006, May). CATS: A Synchronous Approach to Collaborative Group Recommendation. In *FLAIRS Conference* (Vol. 2006, pp. 86-91).

McNally, K., O'Mahony, M. P., Coyle, M., Briggs, P., & Smyth, B. (2011). A case study of collaboration and reputation in social web search. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *3*(1), 4.

McSherry, D. (2005). Explanation in recommender systems. *Artificial Intelligence Review*, *24*(2), 179-197.

McSherry, F., & Mironov, I. (2009, June). Differentially private recommender systems: building privacy into the net. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 627-636). ACM.

Melville, P., Mooney, R. J., & Nagarajan, R. (2002, July). Content-boosted collaborative filtering for improved recommendations. In *AAAI/IAAI* (pp. 187-192).

Merve Acilar, A., & Arslan, A. (2009). A collaborative filtering method based on artificial immune network. *Expert Systems with Applications*, *36*(4), 8324-8332.

Middleton, S. E., Shadbolt, N. R., & De Roure, D. C. (2004). Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, *22*(1), 54-88.

Mnih, A., & Salakhutdinov, R. (2007). Probabilistic matrix factorization. In*Advances in neural information processing systems* (pp. 1257-1264).

Mooney, R. J., & Roy, L. (2000, June). Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries* (pp. 195-204). ACM.

Morrison, T., & Aickelin, U. (2008). An artificial immune system as a recommender system for web sites. *arXiv preprint arXiv:0804.0573*.

Nanopoulos, A., Rafailidis, D., Symeonidis, P., & Manolopoulos, Y. (2010). Musicbox: Personalized music recommendation based on cubic analysis of social tags. *Audio, Speech, and Language Processing, IEEE Transactions on*,*18*(2), 407-412.

Nehring, K., & Puppe, C. (2002). A theory of diversity. *Econometrica*, *70*(3), 1155-1198.

Nilashi, M., Ibrahim, O. B., & Ithnin, N. (2014). Hybrid recommendation approaches for multi-criteria collaborative filtering. *Expert Systems with Applications*, *41*(8), 3879-3900.

Núñez-Valdéz, E. R., Cueva-Lovelle, J. M., Sanjuán-Martínez, O., García-Díaz, V., Ordoñez, P., & Montenegro-Marín, C. E. (2012). Implicit feedback techniques on recommender systems applied to electronic books. *Computers in Human Behavior*, *28*(4), 1186-1193.

O'Donovan, J., & Smyth, B. (2005, January). Trust in recommender systems. In*Proceedings of the 10th international conference on Intelligent user interfaces*(pp. 167-174). ACM.

Ortega, F., Bobadilla, J., Hernando, A., & Gutiérrez, A. (2013). Incorporating group recommendations to recommender systems: Alternatives and performance. *Information Processing & Management*, *49*(4), 895-901.

Ortega, F., Bobadilla, J., Hernando, A., & Rodríguez, F. (2014). Using Hierarchical Graph Maps to Explain Collaborative Filtering Recommendations. *International Journal of Intelligent Systems*, *29*(5), 462-477.

Ortega, F., Sánchez, J. L., Bobadilla, J., & Gutiérrez, A. (2013). Improving collaborative filtering-based recommender systems results using Pareto dominance. *Information Sciences: an International Journal*, *239*, 50-61.

Papadimitriou, A., Symeonidis, P., & Manolopoulos, Y. (2012). A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Mining and Knowledge Discovery*, *24*(3), 555-583.

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

Park, M. H., Hong, J. H., & Cho, S. B. (2007). Location-based recommendation system using bayesian user's preference model in mobile devices. In*Ubiquitous Intelligence and Computing* (pp. 1130-1139). Springer Berlin Heidelberg.

Park, S. T., & Chu, W. (2009, October). Pairwise preference regression for cold-start recommendation. In *Proceedings of the third ACM conference on Recommender systems* (pp. 21-28). ACM.

Park, S. T., Pennock, D., Madani, O., Good, N., & DeCoste, D. (2006, August). Naïve filterbots for robust cold-start recommendations. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 699-705). ACM.

Park, Y. J., & Tuzhilin, A. (2008, October). The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 ACM conference on Recommender systems* (pp. 11-18). ACM.

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

Pazzani, M. J., & Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine learning*, *27*(3), 313-331.

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

Pereira, R., Lopes, H., Breitman, K., Mundim, V., & Peixoto, W. (2014). Cloud based real-time collaborative filtering for item–item recommendations. *Computers in Industry, 65*(2), 279-290.

Perugini, S., Gonçalves, M. A., & Fox, E. A. (2004). Recommender systems research: A connection-centric survey. *Journal of Intelligent Information Systems*, *23*(2), 107-143.

Pham, M. C., Cao, Y., Klamma, R., & Jarke, M. (2011). A Clustering Approach for Collaborative Filtering Recommendation Using Social Network Analysis. *J. UCS*, *17*(4), 583-604.

Pitsilis, G., Zhang, X., & Wang, W. (2011). Clustering recommenders in collaborative filtering using explicit trust information. In *Trust Management V*(pp. 82-97). Springer Berlin Heidelberg.

Popescul, A., Pennock, D. M., & Lawrence, S. (2001, August). Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence* (pp. 437-444). Morgan Kaufmann Publishers Inc..

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

Porcel, C., Moreno, J. M., & Herrera-Viedma, E. (2009). A multi-disciplinar recommender system to advice research resources in university digital libraries.*Expert Systems with Applications*, *36*(10), 12520-12528.

Porcel, C., Tejeda-Lorente, A., Martínez, M. A., & Herrera-Viedma, E. (2012). A hybrid recommender system for the selective dissemination of research resources in a technology transfer office. *Information Sciences*, *184*(1), 1-19.

Pu, P., & Chen, L. (2007). Trust-inspiring explanation interfaces for recommender systems. *Knowledge-Based Systems*, *20*(6), 542-556.

Qin, W., Xin, L., & Liang, H. (2011). Unifying user-based and item-based algorithm to improve collaborative filtering accuracy. *Energy Procedia*, *13*, 8231-8239.

Quijano-Sánchez, L., Díaz-Agudo, B., & Recio-García, J. A. (2014). Development of a group recommender application in a Social Network. *Knowledge-Based Systems, 71*, 72-85.

Rao, K. V., Pujari, A. K., & Padmanabhan, V. (2014). Virtual user approach for group recommender systems using precedence relations. *Information Sciences, 294*, 15-30.

Rashid, A. M., Karypis, G., & Riedl, J. (2008). Learning preferences of new users in recommender systems: an information theoretic approach. *ACM SIGKDD Explorations Newsletter*, *10*(2), 90-100.

Ray, S., & Mahanti, A. (2009). Strategies for effective shilling attacks against recommender systems. In *Privacy, Security, and Trust in KDD* (pp. 111-125). Springer Berlin Heidelberg.

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

Rodrigues Nt, J. A., Tomaz, L. F. C., De Souza, J. M., & Xexéo, G. (2013). Bringing knowledge into recommender systems. *Journal of Systems and Software*, *86*(7), 1751-1758.

Roh, T. H., Oh, K. J., & Han, I. (2003). The collaborative filtering recommendation based on SOM cluster-indexing CBR. *Expert Systems with Applications*, *25*(3), 413-423.

Roy, S. B., Amer-Yahia, S., Chawla, A., Das, G., & Yu, C. (2010). Space efficiency in group recommendation. *The VLDB Journal*, *19*(6), 877-900.

Ryan, P., & Bridge, D. (2006). Collaborative recommending using formal concept analysis. *Knowledge-Based Systems*, *19*(5), 309-315.

Salter, J., & Antonopoulos, N. (2006). CinemaScreen recommender agent: combining collaborative and content-based filtering. *Intelligent Systems, IEEE*,*21*(1), 35-41.

Salton, G. (1989). *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000a, October). Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM conference on Electronic commerce* (pp. 158-167). ACM.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000b). *Application of dimensionality reduction in recommender system-a case study* (No. TR-00-043). Minnesota Univ Minneapolis Dept of Computer Science.

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

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web* (pp. 291-324). Springer Berlin Heidelberg.

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

Schlieder, C. (2007). Modeling collaborative semantics with a geographic recommender. In *Advances in Conceptual Modeling–Foundations and Applications* (pp. 338-347). Springer Berlin Heidelberg.

Serrano-Guerrero, J., Herrera-Viedma, E., Olivas, J. A., Cerezo, A., & Romero, F. P. (2011). A google wave-based fuzzy recommender system to disseminate information in University Digital Libraries 2.0. *Information Sciences*, *181*(9), 1503-1516.

Sevarac, Z., Devedzic, V., & Jovanovic, J. (2012). Adaptive neuro-fuzzy pedagogical recommender. *Expert Systems with Applications*, *39*(10), 9797-9806.

Shepitsen, A., Gemmell, J., Mobasher, B., & Burke, R. (2008, October). Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 ACM conference on Recommender systems* (pp. 259-266). ACM.

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

Siersdorfer, S., & Sizov, S. (2009, June). Social recommender systems for web 2.0 folksonomies. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia* (pp. 261-270). ACM.

Soboroff, I., & Nicholas, C. (1999, August). Combining content and collaboration in text filtering. In *Proceedings of the IJCAI* (Vol. 99, pp. 86-91).

Son, L. H. (2014). HU-FCF: A hybrid user-based fuzzy collaborative filtering method in Recommender Systems. *Expert Systems with Applications, 41*(15), 6861-6870.

Sotelo, R., Blanco-Fernandez, Y., Lopez-Nores, M., Gil-Solla, A., & Pazos-Arias, J. J. (2009). TV program recommendation for groups based on muldimensional TV-anytime classifications. *Consumer Electronics, IEEE Transactions on*, *55*(1), 248-256.

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, *2009*, 4.

Sun, X., Kong, F., & Ye, S. (2005, March). A comparison of several algorithms for collaborative filtering in startup stage. In *Networking, Sensing and Control, 2005. Proceedings. 2005 IEEE* (pp. 25-28). IEEE.

Symeonidis, P., Nanopoulos, A., & Manolopoulos, Y. (2008). Providing justifications in recommender systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, *38*(6), 1262-1272.

Symeonidis, P., Nanopoulos, A., & Manolopoulos, Y. (2009, October). MoviExplain: a recommender system with explanations. In *Proceedings of the third ACM conference on Recommender systems* (pp. 317-320). ACM.

Takács, G., Pilászy, I., Németh, B., & Tikk, D. (2009). Scalable collaborative filtering approaches for large recommender systems. *The Journal of Machine Learning Research*, *10*, 623-656.

Tan, S., Bu, J., Chen, C., & He, X. (2011). Using rich social media information for music recommendation via hypergraph model. In *Social media modeling and computing* (pp. 213-237). Springer London.

Tintarev, N., & Masthoff, J. (2007, April). A survey of explanations in recommender systems. In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on* (pp. 801-810). IEEE.

Tran, T., & Cohen, R. (2000, July). Hybrid recommender systems for electronic commerce. In *Proc. Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, Technical Report WS-00-04, AAAI Press*.

Tso-Sutter, K. H., Marinho, L. B., & Schmidt-Thieme, L. (2008, March). Tag-aware recommender systems by fusion of collaborative filtering algorithms. In*Proceedings of the 2008 ACM symposium on Applied computing* (pp. 1995-1999). ACM.

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

Vargas, S., & Castells, P. (2011, October). Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems* (pp. 109-116). ACM.

Victor, P., Cornelis, C., & De Cock, M. (2011). *Trust networks for recommender systems* (Vol. 4). Springer.

Vig, J., Sen, S., & Riedl, J. (2009, February). Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th international conference on Intelligent user interfaces* (pp. 47-56). ACM.

Vozalis, M. G., & Margaritis, K. G. (2007). Using SVD and demographic data for the enhancement of generalized collaborative filtering. *Information Sciences*,*177*(15), 3017-3037.

Wang, C., & Blei, D. M. (2011, August). Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 448-456). ACM.

Wang, J., De Vries, A. P., & Reinders, M. J. (2006, August). 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* (pp. 501-508). ACM.

Wang, J., De Vries, A. P., & Reinders, M. J. (2008). Unified relevance models for rating prediction in collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, *26*(3), 16.

Wang, Z., Liu, Y., & Ma, P. (2014). A CUDA-enabled Parallel Implementation of Collaborative Filtering. *Procedia Computer Science, 30*, 66-74.

Weng, L. T., Xu, Y., Li, Y., & Nayak, R. (2008, November). Exploiting item taxonomy for solving cold-start problem in recommendation making. In *Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on* (Vol. 2, pp. 113-120). IEEE.

Widrow, B., & Hoff, M. E. (1988, January). Adaptive switching circuits. In *Neurocomputing: foundations of research* (pp. 123-134). MIT Press.

Winoto, P., & Tang, T. Y. (2010). The role of user mood in movie recommendations. *Expert Systems with Applications*, *37*(8), 6086-6092.

Wu, M. L., Chang, C. H., & Liu, R. Z. (2014). Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices. *Expert Systems with Applications*, *41*(6), 2754-2761.

Xie, B., Han, P., Yang, F., Shen, R. M., Zeng, H. J., & Chen, Z. (2007). DCFLA: A distributed collaborative-filtering neighbor-locating algorithm. *Information Sciences*, *177*(6), 1349-1363.

Xie, H., & Lui, J. (2014). Mathematical modeling of group product recommendation with partial information: How many ratings do we need?. *Performance Evaluation*, 77, 72-95.

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

Yang, W. S., Cheng, H. C., & Dia, J. B. (2008). A location-aware recommender system for mobile shopping environments. *Expert Systems with Applications*,*34*(1), 437-445.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization.*Information retrieval*, *1*(1-2), 69-90.

Yao, Z., & Zhang, Q. (2009, April). Item-based clustering collaborative filtering algorithm under high-dimensional sparse data. In *Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on* (Vol. 1, pp. 787-790). IEEE.

Yetim, F. (2008). A framework for organizing justifications for strategic use in adaptive interaction contexts.

Yu, Z., Zhou, X., Hao, Y., & Gu, J. (2006). TV program recommendation for multiple viewers based on user profile merging. *User Modeling and User-Adapted Interaction*, *16*(1), 63-82.

Yuan, W., Guan, D., Lee, Y. K., Lee, S., & Hur, S. J. (2010). Improved trust-aware recommender system using small-worldness of trust networks. *Knowledge-Based Systems*, *23*(3), 232-238.

Zaíane, O. R. (2002, December). Building a recommender agent for e-learning systems. In *Computers in Education, 2002. Proceedings. International Conference on* (pp. 55-59). IEEE.

Zhan, J., Hsieh, C. L., Wang, I. C., Hsu, T. S., Liau, C. J., & Wang, D. W. (2010). Privacy-preserving collaborative recommender systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*,*40*(4), 472-476.

Zhang, F., & Chang, H. Y. (2006, October). A collaborative filtering algorithm employing genetic clustering to ameliorate the scalability issue. In *e-Business Engineering, 2006. ICEBE'06. IEEE International Conference on* (pp. 331-338). IEEE.

Zhang, J., Peng, Q., Sun, S., & Liu, C. (2014). Collaborative filtering recommendation algorithm based on user preference derived from item domain features. *Physica A: Statistical Mechanics and its Applications, 396*, 66-76.

Zhang, Z. K., Yu, L., Fang, K., You, Z. Q., Liu, C., Liu, H., & Yan, X. Y. (2014). Website-oriented recommendation based on heat spreading and tag-aware collaborative filtering. *Physica A: Statistical Mechanics and its Applications, 399*, 82-88.

Zhen, L., Huang, G. Q., & Jiang, Z. (2009a). Collaborative filtering based on workflow space. *Expert Systems with Applications*, *36*(4), 7873-7881.

Zhen, L., Huang, G. Q., & Jiang, Z. (2009b). Recommender system based on workflow. *Decision Support Systems*, *48*(1), 237-245.

Zhen, L., Jiang, Z., & Song, H. (2010). Distributed recommender for peer-to-peer knowledge sharing. *Information Sciences*, *180*(18), 3546-3561.

Zheng, N., & Li, Q. (2011). A recommender system based on tag and time information for social tagging systems. *Expert Systems with Applications*, *38*(4), 4575-4587.

Zheng, Y., & Xie, X. (2011). Learning travel recommendations from user-generated GPS traces. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(1), 2.

Zheng, Y., Zhang, L., Ma, Z., Xie, X., & Ma, W. Y. (2011). Recommending friends and locations based on individual location history. *ACM Transactions on the Web (TWEB)*, *5*(1), 5.

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

Zhu, L., & Lerman, K. (2014). A Visibility-based Model for Link Prediction in Social Media. In *Proceedings of the ASE/IEEE Conference on Social Computing*.

Zhu, R., & Gong, S. (2009, August). Analyzing of collaborative filtering using clustering technology. In *Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on* (Vol. 4, pp. 57-59). IEEE.

Ziegler, C. N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005, May). Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web* (pp. 22-32). ACM.

INDEX OF FIGURES

FIGURE 1.1. RECOMMENDER SYSTEMS CLASIFICATION. 6

FIGURE 1.2. RECOMMENDER SYSTEMS TRENDS CLASSIFIED BY DATA SOURCES. 9

FIGURE 1.3. DIFFERENT ALTERNATIVES FOR COMBINING CONTENT BASED FILTERING (CBF) AND COLLABORATIVE FILTERING (CF). 13

FIGURE 1.4. USER TO USER COLLABORATIVE FILTERING PROCESS EXAMPLE. 17

FIGURE 1.5. ITEM TO ITEM COLLABORATIVE FILTERING PROCESS EXAMPLE. 18

FIGURE 1.6. COLLABORATIVE FILTERING EVALUATION PROCESS 26

FIGURE 2.1. STATE OF THE ART OF THE CF RECOMMENDATIONS TO A GROUP OF USERS. 44

FIGURE 3.1. PROPOSED METHOD OVERVIEW 48

FIGURE 4.1. EXPERIMENTS OVERVIEW. 57

FIGURE 4.2. SYSTEM ACCURACY AND COVERAGE COMPARISONS USING DIFFERENT CF SIMILARITY METRICS WITH THE ERRS METHOD IN MOVIELENS. 58

FIGURE 4.3. SYSTEM PRECISION AND RECALL COMPARISONS USING DIFFERENT CF SIMILARITY METRICS WITH THE ERRS METHOD IN MOVIELENS. 59

FIGURE 4.4. SYSTEM ACCURACY AND COVERAGE COMPARISONS USING AS RECOMMENDATION METHODS: UGSM, ERRS (SING) AND PER. MOVIELENS DATASET USED. 59

FIGURE 4.5. SYSTEM PRECISION AND RECALL COMPARISONS USING AS RECOMMENDATION METHODS: UGSM, ERRS (SING), PER AND RANK. MOVIELENS DATASET USED. 60

FIGURE 4.6. SYSTEM ACCURACY AND COVERAGE COMPARISONS USING AS RECOMMENDATION METHODS: UGSM, ERRS (SING) AND PER. NETFLIX DATASET USED. 60

FIGURE 4.7. SYSTEM PRECISION AND RECALL COMPARISONS USING AS RECOMMENDATION METHODS: UGSM, ERRS (SING), PER AND RANK. NEFLIX DATASET USED 61

FIGURE 4.8. SYSTEM PERFORMANCE COMPARISON USING DIFFERENT GROUP RECOMMENDATION ALGORITHMS BASED ON CF IN MOVIELENS. 61

INDEX OF TABLES

TABLE 1.1. MOST OFTEN USED MEMORY-BASED RECOMMENDER SYSTEMS PUBLIC DATABASES. 23

TABLE 3.1. RUNNING EXAMPLE RATINGS MATRIX. 50

TABLE 3.2. SIMILARITY BETWEEN G AND EACH USER OF THE RS. 51

TABLE 3.3. PREDICTIONS GENERATED TO THE GROUP OF USERS. 51

TABLE 4.1. MAIN INFORMATION ABOUT THE DATABASES USED ON THE EXPERIMENTS. 56

TABLE 4.2. MAIN PARAMETERS USED IN THE EXPERIMENTS. 56

TABLE 4.3. P-VALUES OF ALL THE T-TESTS CARRIED OUT TO COMPARE THE PERFORMANCE OF THE DIFFERENT GROUP RECOMMENDATION ALGORITHMS IN MOVIELENS. 62