UNIVERSIDAD POLITECNICA DE MADRID

DOCTORAL THESIS

**Towards reliability in collaborative
filtering recommender systems**

Author:

Bo ZHU

Supervisor:

Dr. Jesús BOBADILLA
SANCHO

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor*

in the

Intelligent Systems

Escuela Técnica Superior de Ingeniería de SISTEMAS INFORMÁTICOS

September 20, 2018

Declaration of Authorship

I, Bo ZHU, declare that this thesis titled, “Towards reliability in collaborative filtering recommender systems” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signed:

Date:

*"A journey of a thousand miles may not be achieved without accumulating each single step,
just as the enormous ocean may not be formed without gathering every stream."*

Xun Kuang

Abstract

Recommender systems have experienced a rapid growth in both academical research and industrial applications in latest years. Recommendations are provided to us when we choose items such as music, movies, restaurants, news, etc. As the most promising recommendation techniques, collaborative filtering recommender systems play an important role in current applications. The emergence of novel information sources requires to combine recommender systems with several advanced techniques coming from other fields like social network analysis, natural language processing, deep learning etc. This combination improves the quality of recommendations.

Previous recommender systems research mainly focuses on the improvements of recommendation accuracy. Recently a new trend arises for the performance evaluation of recommender systems that takes many novel properties into consideration. Reliability is one of the most remarkable aspects, which will be deeply studied in this thesis. Some measures have been proposed for measuring reliability of memory based collaborative filtering recommender systems. However, we observe a gap in the design of proper reliability measures for model based collaborative filtering recommender systems, especially those based on matrix factorization. We also found that there is a lack in proper reliability quality measures. This thesis concentrates on providing solutions to both lacks.

After a comprehensive study of state-of-the-art research works, we propose a novel reliability measure that overcomes the challenges of generality and abstraction caused by latent factors of matrix factorization methods. The proposed measure is tested against current baselines on public academic and commercial datasets; experiment results show that our novel measure achieves better improvements on recommendation quality. What's more, the proposed reliability measure also has a very positive impact against shilling attacks. In order to fill in the blank of reliability quality measures, we also propose, as far as we know, the first universal reliability quality measure that is able to provide numerical values for any type of recommender systems. This measure has also been tested on public academic and commercial datasets using current methods. Based on the experiment results we conclude that our quality measure outperforms state-of-the-art research works by providing more accurate and informative reliability quality values.

Resumen

Los sistemas de recomendación han tenido un rápido crecimiento tanto en investigación académica como en aplicaciones industriales en los últimos años. Las recomendaciones se nos proporcionan cuando elegimos elementos como música, películas, restaurantes, noticias, etc. Como parte de las técnicas de recomendación más prometedoras, los sistemas basados en recomendación de filtros colaborativos desempeñan un papel importante en las aplicaciones actuales. La aparición de nuevas fuentes de información requiere la combinación de sistemas de recomendación con técnicas avanzadas de otros campos, como el análisis de redes sociales, el procesamiento del lenguaje natural, el aprendizaje en profundidad, etc. Esta combinación mejora la calidad de las recomendaciones.

Los sistemas de recomendación anteriores se centran principalmente en las mejoras de la precisión de la recomendación. Recientemente, surge una tendencia para la evaluación del rendimiento de los sistemas de recomendación que toma muchas propiedades nuevas en consideración. La fiabilidad es uno de los aspectos más notables y será profundamente estudiado en esta tesis. Se han propuesto algunas medidas para medir la fiabilidad de los sistemas de recomendación de filtrado colaborativo basados en la memoria. Sin embargo, observamos un área sin cubrir en el diseño de las medidas de fiabilidad adecuadas para los sistemas de recomendación de filtros colaborativos basados en modelos, especialmente aquellos basados en la factorización matricial. También encontramos que hay una falta de medidas de calidad de fiabilidad adecuadas. Esta tesis se concentra en proporcionar soluciones a ambas ausencias.

Después de un estudio exhaustivo de trabajos de investigación del estado del arte, proponemos una nueva medida de fiabilidad que supera los desafíos de generalización y abstracción causados por los factores latentes de los métodos de factorización matricial. La medida propuesta se compara con métodos y algoritmos representativos, usando bases de datos públicas, tanto académicas como comerciales. Los resultados experimentales muestran que nuestra nueva medida logra mejoras en la calidad de la recomendación. Además, la medida de fiabilidad propuesta también tiene un impacto muy positivo contra los "Shilling Attacks". Para obtener una mejora respecto a las medidas existentes de calidad de la fiabilidad, también proponemos, hasta donde sabemos, la primera medida de calidad de fiabilidad universal que es capaz de proporcionar valores numéricos para cualquier tipo de sistema de recomendación. Esta medida también se ha probado usando conjuntos de datos públicos reales y ha sido comparada con los métodos de calidad de la fiabilidad existentes. Basándonos en los resultados de los experimentos, concluimos que nuestra medida supera los últimos trabajos de investigación, proporcionando valores de calidad de fiabilidad más precisos e informativos.

Acknowledgements

First of all, I would like to give my sincere gratitude to my PhD supervisor Jesús Bobadilla Sancho. This PhD thesis would not have been possible without his meticulous help. It is from him that I learned what characteristics a professional researcher should have: rigour, enthusiasm, patience and extraordinary intelligence, as well as a great sense of humour. He has continuously motivated me during my entire PhD period. It is an honour and fortune to conduct PhD research under his supervision.

My special thanks also go to Lecturers Abraham Gutiérrez Rodríguez, Fernando Ortega Requena, Xin Li, and my colleagues for their kind help. I really appreciate their fruitful discussions and beneficial advices.

I also want to thank all my friends here in Madrid. It would have been really boring without their accompanying.

Finally, I want to give my deepest gratefulness to my beloved family. Without their support, I would not have succeeded in going through those sleepless nights. Their concern, persuasion and insightful advices have always warmed my heart and kept me up. It is their encouragement that makes me through the darkness before dawn. My dearest wife deserves my forever gratefulness for her unconditional love and understanding, and accompanying me along the whole journey.

Once again, many thanks to all that have offered me their precious and warm help to make this PhD study an unforgettable experience.

Contents

Declaration of Authorship	iii
Abstract	vi
Resumen	vii
Acknowledgements	ix
1 Introduction	1
1.1 Preliminary concepts	2
1.2 Motivations	5
1.3 Objectives and hypothesis	8
1.4 Structure of the thesis	9
2 Related work	11
2.1 Content based RS	12
2.1.1 Advantages and defects of content based methods	12
2.1.2 State of the art of content based methods	14
2.2 Collaborative filtering RS	15
2.2.1 Advantages and drawbacks of CF based methods	16
2.2.2 Memory based CF methods	18
2.2.3 Model based CF methods	24
2.3 Demographic RS	37
2.4 Domain-based taxonomy	38
2.4.1 Location based RS	38
2.4.2 Social RS	40
2.4.3 Review based RS	40
2.4.4 Time aware RS	41
2.5 RS evaluation	42
2.5.1 Accuracy	42
2.5.2 Properties beyond accuracy	46
2.5.3 Reliability	50
2.6 Summary	55
3 Reliability measures for recommender systems	57
3.1 Introduction	58

3.2	A novel reliability quality measure for recommender systems	60
3.3	Experiments	65
3.3.1	Reliability value distribution	66
3.3.2	Accuracy vs. reliability	67
3.3.3	Precision vs. reliability	69
3.3.4	Reliability vs. number of recommendations	72
3.3.5	Comparative results using a state of the art baseline	74
3.3.6	Introducing additional information	75
3.3.7	Testing shilling attacks impact	79
3.4	Summary	81
4	Evaluation of the quality of reliability measures	83
4.1	Introduction	84
4.2	Proposed methods	87
4.3	Formulation of the proposed models	90
4.4	Experimental results and discussions	93
4.4.1	Introduction	93
4.4.2	Tested reliability measures	95
4.4.3	Mazurovsky's method	96
4.4.4	Prediction reliability quality results	98
4.4.5	Recommendation reliability quality results	99
4.4.6	Experiment discussion	101
4.5	Summary	102
5	Conclusions and future works	105
A	List of author's publications	109
	Bibliography	111

List of Figures

1.1	A brief introduction to RS	2
1.2	An illustration of RS	4
1.3	New trend of RS	5
1.4	An example of reliability	6
2.1	High level architecture of Content-based RS	13
2.2	A detailed explanation of the heuristics	21
2.3	Fuzzy model based CF methods	26
2.4	neural network based CF methods	34
2.5	Loosely and tightly coupled methods	34
2.6	Youtube video recommendation architecture	36
2.7	Deep neural network architecture for the first stage	36
2.8	Deep neural network architecture for the second stage	37
2.9	An illustration of review based RS	41
2.10	RS evaluation process	43
3.1	Hierarchical architecture of the proposed method	62
3.2	Hierarchical architecture of the proposed method: Details	63
3.3	Distributions of the reliability values obtained for users and items	67
3.4	MAE values obtained for each possible reliability value	68
3.5	Averaged reliabilities values obtained for 15 groups of predictions errors	70
3.6	Recommendation quality (precision) improvements	71
3.7	Coverage of the proposed method	73
3.8	Inverse relation between reliability and Mean Absolute Error	76
3.9	Number of rating probabilities	77
3.10	Prediction quality (Mean Absolute Error) values obtained using several reliability measures	79
3.11	Recommendation quality (Precision) values obtained using the “Proposed” and the “Mixed” reliability measures	80
3.12	Shilling attack: Reliability values and MAE results	81
4.1	Reliability measures and reliability quality measures	85
4.2	Historical evolution on the reliability quality field and the new proposal	86
4.3	Experimentation process stages	94
4.4	Cross validation process to obtain reliability quality prediction values	95

4.5	Baseline method scheme	97
4.6	Reliability quality results (referring to predictions) obtained when using several RM, reliability quality methods and recommender systems datasets	98
4.7	Reliability quality results (referring to recommendations) obtained when using several reliability measures and recommender system datasets .	100

List of Tables

1.1	Examples of different kinds of information sources.	6
2.1	Main advantages and disadvantages of the memory-based and model-based approaches.	17
2.2	A summary on number of publications using different neural network models.	35
3.1	Datasets used to test the proposed method.	66
3.2	Experimental settings and parameters.	66
3.3	Datasets used to test the proposed method, baseline parameters, proposed method parameters and tested quality measure.	75
3.4	Reliability versus Mean Absolute Error correlation results. The highest (negative) correlation values correspond to the best experiment results.	77
4.1	Reliability quality measure: penalty and reward situations.	88
4.2	Parameters.	91
4.3	Reliability quality measure: Penalty and reward situations.	91
4.4	Tested reliability measures.	96

List of Abbreviations

RS	Recommender Systems
ML	Machine Learning
AI	Artificial Intelligence
CF	Collaborative Filtering
PSO	Particle Swarm Optimization
GA	Genetic Algorithm
MF	Matrix Factorization
SVD	Singular Vector Decomposition
MMMF	Maximum Margin Matrix Factorization
NMF	Non-negative Matrix Factorization
SGD	Stochastic Gradient Descent
ALS	Alternating Least Squares
PMF	Probabilistic Matrix Factorization
MCMC	Markov Chain Monte Carlo
BPMF	Bayesian Probabilistic Matrix Factorization
PLSA	Probabilistic Latent Semantic Analysis
LDA	Latent Dirichlet Allocation
FM	Factorization Machine
SVM	Support Vector Machine
MLP	MultiLayer Perceptron
AE	AutoEncoder
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
DSSM	Deep Semantic Similarity Model
RBM	Restricted Boltzmann Machine
NADE	Neural Autoregressive Distribution Estimation
GAN	Generative Adversarial Network
POI	Point Of Interest
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
ROC	Receiver Operating Characteristic
AUC	Area Under ROC Curve
NDCG	Normalized Discounted Cumulative Gain
CF4J	Collaborative Filtering for Java
KNN	K Nearest Neighbors

RM	Reliability Measure
RQM	Reliability Quality Measure

Dedicated to my dear family

Chapter 1

Introduction

In recent years, research works and industrial applications of recommender systems have been explosively increasing in many domains such as E-commerce, traveling, online news, music etc. Thousands of recommendation models have been proposed to continuously update the upper bounder of the recommendation performance. Apart from the explicit rating information, adequate side information of all types is being integrated into the source data for personalized recommendations of better quality.

In this chapter, some preliminary concepts of this thesis are first introduced, followed with the motivations, objectives and hypothesis. Our contributions to the recommender system community are then listed, and finally the structure of this thesis is given.

1.1 Preliminary concepts

Nowadays we are living in a world crowded by so many information sources that it is impossible for us to absorb even a very small portion. This phenomenon is usually called "information overload". Since people are not able to track all this information and manually select what they are interested in, recommender systems are proposed and developed to help us filter out the uninteresting majority, and discover the few with high relevance to our personal preferences. These intelligent systems appear in many activities of our daily life, such as selecting movies to watch, reserving hotels for vacations, purchasing interesting products at E-commerce websites, finding friends with similar interests, and even searching for novel scientific publications that fit one's research lines.

As shown in Figure 1.1, recommender systems (RS) are a branch of machine learning (ML), which is an extremely popular academic and industrial sub-field of artificial intelligence (AI). RS can be roughly divided into content based RS, collaborative filtering RS and demographic RS. Any combination of these three types generates hybrid models. This traditional taxonomy will be comprehensively introduced in next chapter.

FIGURE 1.1: A brief introduction to RS. AI stands for artificial intelligence, and ML refers to machine learning.

Collaborative filtering (CF) RS is one of the most popular and successful family of existing recommendation methods. It is based on the assumption that "users will rate items similarly or have similar behaviors in the future if they rated or acted similarly in the past" (Goldberg et al., 1992). (Wang et al., 2016) divide CF into rating-based CF and ranking-based CF, while (Shi, Larson, and Hanjalic, 2014; Bobadilla et al., 2013b) classify CF into memory-based CF and model-based CF. Pure CF RS have difficulty

in dealing with cold-start, sparsity, synonymy and grey sheep problems. The widely accepted solution is to combine CF RS with content based RS or demographic RS to leverage their advantages while avoiding the shortcomings. Some popular enhanced CF recommendation algorithms have been demonstrated in Figure 1.2.

RS collect users' preferences over different items, which are usually represented as a User-Item matrix, in an explicit or implicit way. Explicit ratings such as 1-5 stars, like/dislike or implicit feedbacks such as favorite song playlists, product purchase history, checked-in places and fully watched videos etc, are the most important information source for traditional RS. As shown in Figure 1.2, this information lays the foundation for the basic RS.

Together with the prosperity of a wide variety of RS scenarios, novel information sources are now becoming available for the enhancement of recommendation algorithms to further improve the performance of RS. In Figure 1.3 we summarize different information sources into user-item interaction which is the previously mentioned User-Item matrix, item side information, user side information and user-item interaction side information (Shi, Larson, and Hanjalic, 2014). Figure 1.2 also organizes some examples of these new sources into four levels according to the continuous Web evolution. Here we reorganize them based on the novel category and add some supplementaries in Table 1.1. We further associate different sources to some RS scenarios in Figure 1.3. Note that cross domain RS (Tang et al., 2012) is a very promising research topic that intends to transfer user preferences in a certain single domain to other crossing domains. Take the e-commerce platforms like Amazon as an obvious instance where the user consumption histories of different types of product are tracked. When a user purchases a laptop, accessories like mouses and headsets will be recommended to him. While movies like "The butterfly" and "Little miss sunshine" are rated both high by many users, then the music "The winter is" should be recommended to a user who have listened to "Le papillon", but not aware of former song.

The rapid growth of recommendation algorithms, as well as the emergence of more and more information sources, have encouraged the great development of evaluation of RS. Accuracy is the first thing users pay attention to since inaccurate or even contrary recommendations will greatly damage their trust in the RS. This quality measure is highly valued by users and somehow correlated to their satisfactions.

At the top of Figure 1.2, some commonly used recommendation quality measures are listed. Besides the accuracy of recommendation results, there are more aspects that should draw our attention, among which we insist on the importance of reliability. Reliability and confidence (in prediction) are synonyms and usually used alternatively. We choose reliability as the term name in this thesis because the word confidence is ambiguous and can refer to either the confidence RS have in their predictions or the confidence users have in RS (Swearingen and Sinha, 2001; Shani and

FIGURE 1.2: An illustration of RS (Bobadilla et al., 2013b)

FIGURE 1.3: New trend of RS by incorporation of more data sources

Gunawardana, 2011). Let us explain reliability by taking the hotel recommendation in Booking as an example, Figure 1.4 shows two recommended hotels at the same price for the same date. If we only consider the previous review scores, we shall definitely choose the hotel below. However, if we take the number of reviews into consideration, we may change to choose Catalonia Gran Via hotel due to the fact 5292 visitors have agreed it is a very good choice, while the other hotel is only recommended by 33 visitors. The high recommendation score may be the consequence of some irresponsible exaggerated comments. Such review score outliers from possible gray sheep users are supposed to be rare and will be averaged out by a larger population of reviewers, and in this way, the recommendation score of the upper hotel can be viewed more reliable than that of the lower one.

1.2 Motivations

Despite the importance of reliability, we observe a lack of relevant research works conducted on this RS property, in recent years, comparing to other non-accuracy RS properties like novelty or diversity. (Herlocker, Konstan, and Riedl, 2000; Swearingen and Sinha, 2001) first point out that the users' satisfaction level can be improved by jointly considering accuracy and confidence of the predicted ratings. (McNee et al., 2003) propose to use the number of ratings an item receives as the reliability measure (RM), just like what we choose in the above hotel recommendation example. (Shani and Gunawardana, 2011) summarize two different RMs: the probability that the rating prediction is correct, and the confidence interval of the rating prediction at a predefined level, e.g. 95%. Some papers (Hernando et al., 2013b; Zhang,

TABLE 1.1: Examples of different kinds of information sources.

	Explicit	Implicit
User-item interaction	Like/dislike, numerical or ordinal ratings etc.	Purchase history, query log, click history, retweets etc.
Item side information	Item general descriptions etc.	Tags (folksonomies), context information like RFID etc.
User side information	Conventional demographic info such as age, gender, nationality etc. Social connections like friends, followers, followees, trusters, trustees and context information like health sensor data	Reputation, location info like routes, check-in history etc.
User-item interaction side information	Sentiment tags etc.	Images, videos, text reviews, timestamps and context information like companion, GeoTags, season, weather etc.

Catalonia Gran Via

★★★★★

Madrid City Centre, Madrid – [Show on map](#) (500 m from centre) – Metro access

Very good 8.4
5,292 reviews
Staff 8.3

Popular now! 8 people are looking at this moment
In high demand! Booked 53 times in the last 24 hours

Twin/Double Room € 180

In high demand!
Risk free: You can cancel later, so lock in this great price today.

[See our last available rooms >](#)

Apartamento La Famosa Gran Via

Apartment

Madrid City Centre, Madrid – [Show on map](#) (500 m from centre) – Metro access

Superb 9.2
33 reviews
Staff 9.5

Apartment – 40 m²
In high demand!
Risk free: You can cancel later, so lock in this great price today.

€ 180

[Choose your apartment >](#)

FIGURE 1.4: An example of reliability of recommendation results

Guo, and Chen, (2016b) establish RMs in a heuristic way. For example, (Hernando et al., 2013b) propose a RM that introduces both appropriate positive and negative factors to fit different RS scenarios. Several papers (Moradi and Ahmadian, 2015; Mahdi et al., 2017) focus on the RMs when additional trust information is available in social RS scenario. The advantage of introducing such information is obvious: if

a close friend recommends you some product, then this recommendation is more reliable to you because it is probably based on your friend's personal experience. If a professional reviewer you follow on Twitter criticizes some movie, you may consider this as a reliable warning as well. The problem of these RMs is that they only perform well when certain types of additional information are available.

We observe the fact that nearly all published researches concentrate on incorporating reliability information into memory based CF RS. As far as we know, there is only one publication (Chang and Hsiao, 2011) that partially discusses the reliability property in model-based CF RS. However, the focus of (Chang and Hsiao, 2011) is to improve the reliability of the input rating matrix with missing values, rather than to measure the reliability of rating predictions. Comparing to memory based CF RS, incorporating reliability into model based CF RS is more challenging because the prediction is conducted on basis of latent factors that can not be explicitly explained. However, the investigation on RMs for model based CF RS should definitely be carried out in spite of the great difficulties. It is our duty to contribute to such field because model based CF RS are showing their unique potentiality in RS applications. Studies on beyond accuracy properties will significantly further empower model based methods.

Another gap we have discovered is about the reliability quality measures (RQM). As RMs are drawing more attentions, we expect more novel RMs be proposed in the near future. It is important to first distinguish RQM from RM. A RM provides evidence about how reliable a predicted rating value is, while a RQM applies a testing strategy to acquire the quality of assigned reliability values. Direct comparisons of confidence intervals provided by different RMs are meaningless, as indicated in (Shani and Gunawardana, 2011), because "some RMs are weaker and easier to estimate. In such a case a more accurate estimate of a weaker RM does not imply a better recommender."

A proper RQM is of great significance and usefulness to select the best RM from many candidates. Although the reliability of RS is becoming a popular research field, there is still a surprising total lack of numerical universal RQMs. Till now very few researches have been conducted on measuring the quality of RMs. Some early publications on RMs (Swearingen and Sinha, 2001; Herlocker et al., 2004) do not provide concrete numerical RQMs. The evaluation of RMs is constructed on the concept that the reliability values and the corresponding accuracy values should be positively correlated. (Hernando et al., 2013b) provide two heuristic ways to measure the quality of RMs. One way is to establish some threshold first and by means of graphs, it proves their hypothesis that "the more reliable a prediction, the less liable it is to be wrong" (Hernando et al., 2013b). The other way is based on the use of "confidence curve" where prediction errors is graphically compared to reliability values. (Mazurowski, 2013) propose an analysis technique using the "confidence curve" as well. In summary, either heuristic thresholds or graphs are leveraged in

previous RQMs. No universal numerical RQM has been properly proposed yet. It is extremely important to fill in the blank of investigations on such RQMs.

1.3 Objectives and hypothesis

The lack of related research works on reliability measures for model based collaborative filtering recommender systems and numerical reliability quality measures have motivated us to carry out some pioneering researches on such topics, having a comprehensive overview on recommender system research. The main objective of this thesis is to fill in the previously mentioned research gaps; this general purpose is achieved by several sub-objectives:

- To identify the state-of-the-art research works in recommender systems, with an emphasis on collaborative filtering ones and the reliability property.
- To design a reliability measure suitable for matrix factorization model based collaborative filtering recommender systems that overcomes the challenges of generality and abstraction brought by the latent factors.
- To propose a novel and universal reliability quality measure that is able to numerically evaluate the quality of different reliability measures.

To achieve these objectives, we set out a hypothesis for this doctoral thesis as follows:

Given the fact that several reliability measures that work for memory based collaborative filtering recommender systems have been proposed, it is possible to design reliability measures that work for matrix factorization model based collaborative filtering recommender systems; it is also possible to design reliability quality measures that work universally for both memory based and model based collaborative filtering recommender systems. Such measures are able to provide numerical values, in a similar way as the existing quality measures.

Based on this hypothesis, we provide some scientific contributions on the collaborative filtering recommender systems.

- An insightful review on research topics on recommender systems is given to facilitate future researches, especially collaborative filtering ones. Some promising trends are also introduced to help readers to catch up with the research frontiers.
- We propose a novel reliability measure for matrix factorization based collaborative filtering recommender systems in order to assign reliability values to the corresponding predictions. By incorporating this additional information we managed to significantly improve the recommendation quality. It also shows potential capability to handle shilling attacks.

- We further propose reliability quality measures for prediction and recommendation that intend to measure how suitable each reliability measure is.

1.4 Structure of the thesis

For the convenience of readers, we list the structure of this thesis as follows.

1. Chapter 2 introduces the state-of-the-art research works on the foundations of RS. Both the traditional and domain-based taxonomies, as well as important evaluation metrics for different system properties are surveyed. A brief discussion of new trends in RS is also presented.
2. Chapter 3 proposes a novel reliability measure for RS using matrix factorization.
3. Chapter 4 conducts a further research on reliability, and proposes a reliability quality metric that measures how reliable the reliability measures are.
4. Chapter 5 draws the conclusions of this thesis, and points out the plan of future work.

Chapter 2

Related work

Recommender systems (RS) have been intensively studied in the last forty years. Thousands of recommendation algorithms have been proposed, and even more real-world applications have emerged in many scenarios of our daily life. In this chapter, a comprehensive introduction on the fundamental knowledge about RS is given to prepare the readers for further discussions in later chapters.

At the beginning of this chapter, we will first introduce different types of RS according to traditional taxonomy which divides them into content based, collaborative filtering and demographic RS. Some hybrid models are introduced as their enhanced versions. The current methods are formally presented in their corresponding subsections. Later a domain-based taxonomy is given considering the rich novel side information available in different RS scenarios. Then various quality measures are presented for the evaluation of RS from different aspects in order to better fit users' interests. Finally, the state-of-the-art research works on reliability are comprehensively introduced.

2.1 Content based RS

Content based RS assume that users' preferences will remain the same during a relatively long period. In general, the objective of this type of RS is to recommend items that are sufficiently similar to those the user showed his interest in an explicit way (such as giving a high rating to a movie etc.) or in an implicit way (such as saving a product to the shopping cart in Amazon etc.).

RS that are based on content information consist of three principal steps (Ricci, Rokach, and Shapira, 2011):

1. The first step is the extraction of features from the content information in order to properly represent items. This is a crucial step for content based RS because the item representations affect the performance of the following steps (Pazzani and Billsus, 2007). Proper feature extraction techniques should be applied for different domain, together with corresponding domain knowledge. In the high level architecture in Figure 2.1, this step is carried out in the component CONTENT ANALYZER.
2. The second step is the construction of user model/profile which learns the generalized user preferences via training user examples. The PROFILE LEARNER component in Figure 2.1 takes in charge of modeling the user interests reflected by previous positive and negative rating or interacting history for different items. Different learning algorithms are applied depending on the application domain. For recommending products, rule-based or probabilistic classification models etc. can be properly trained by sufficient positive and negative training samples; while in fields like news recommendation, traditional information retrieval techniques like Relevance feedback (Rocchio, 1971) can be applied on news documents representations. Such representations are usually obtained by the vector space model with the classic Term Frequency-Inverse Document Frequency (TF-IDF) weighting method.
3. Leveraging the item representations and user profiles obtained in the above two steps, this step intends to recommend the most relevant item(s) to each user. The relevance between each item and user pair is either assigned by a binary classifier model or calculated using a certain relevance (similarity) measure. This process is conducted in the FILTERING COMPONENT in Figure 2.1.

2.1.1 Advantages and defects of content based methods

Comparing to Collaborative filtering RS, content based systems have several advantages (Lops, Gemmis, and Semeraro, 2011):

FIGURE 2.1: High level architecture of Content-based RS (Lops, Gemmis, and Semeraro, 2011).

1. The learning process of each user's preference is solely dependent on his feedbacks on the set of item representations. No interaction with other users, such as searching for neighbors, is required. This is named "User Independence" in (Lops, Gemmis, and Semeraro, 2011). This property makes content based approaches more adaptive to sparse datasets.
2. The recommendation results provided by content based RS have better interpretability than those obtained by collaborative ones, which refers to "Transparency" in (Lops, Gemmis, and Semeraro, 2011). This is because that the automatically extracted or manually designed features are comprehensible to human users. However, the recommendation list given by collaborative filtering methods is vaguely explained by some similarities between users that are not directly interpretable, or "transparent" to humans.
3. Content based methods are especially competent to "cold start" problem when fresh new items with no ratings are involved. Thanks to extra features suggested by domain experts, the representation generation of items is independent of users' feedback (Balabanović and Shoham, 1997). The recommendation can be conducted as long as the item representations and user profiles are generated, even in the absence of any feedback for a certain item. In contrary, collaborative filtering methods are severely affected by "cold start" problem and can not achieve any information about this new item since no interaction among users is available.

Content based RS have three drawbacks (Pazzani and Billsus, 2007; Lops, Gemmis, and Semeraro, 2011; Su and Khoshgoftaar, 2009; Balabanović and Shoham, 1997):

1. The performance of this type of approaches is severely affected if the content information is not sufficient to generate enough meaningful representations. This is called "Limited Content Analysis" in (Lops, Gemmis, and Semeraro, 2011; G. Adomavicius, 2005). Automated feature extraction is practically difficult in multimedia domains such as video, music or fields like poetry and domain knowledge is crucial for the extension of the reliable feature set. Take news recommendation as an example, features extracted without manual supervision, such as keyword frequencies, can only capture the topic information of news to be recommended. Other important aspects like publisher impact, or the writing quality of news text are out of consideration, and should be suggested by domain experts.
2. Another shortcoming of content based RS is "Over Specialization" (Lops, Gemmis, and Semeraro, 2011; G. Adomavicius, 2005), which refers to the fact that this kind of approaches inherently suffers from a lack of novelty (Bobadilla et al., 2011; Hurley and Zhang, 2011) or serendipity (Ge, D-Battenfeld, and Jannach, 2010; Kotkov, Wang, and Veijalainen, 2016) in its recommendation results. Since user profiles are constructed uniquely upon previous user preferences, these systems are capable to recommend only those items that match the captured preferences, while novel and unexpected items that the user may like are totally out of consideration. This is a vital drawback if RS keep recommending just jazz music to a jazz fan, and limits his exploration to some classic music that he might like as well. Ignoring the diversity and dynamics nature of user preferences will lead to users' ultimate fatigue.
3. To learn the user profile (Step 2) a considerable quantity of user feedbacks are required by the supervised machine learning module. When a "New User" (Lops, Gemmis, and Semeraro, 2011; G. Adomavicius, 2005) enters the system, little information is available to precisely acquire his preference. As a result, the quality of recommendations can not be guaranteed.

2.1.2 State of the art of content based methods

Content based RS have been intensively studied in the latest years. A plenty of methods were proposed regarding each of the three principal steps.

For generating item representations, different techniques are applied based on the item data types. If the input data is well structured, i.e. all items share the same finite set of features, the representations of item can be automatically generated. However, in domains like news, video or music recommendation, text information retrieval and natural language processing techniques must be conducted for the generation

of item representations. (Ahn et al., 2007; Mooney and Roy, 2000; Mak, Koprinska, and Poon, 2003) adopt keyword-based vector space models to learn the representations of items based on their statistical properties. (Magnini and Strapparva, 2000; Eirinaki, Vazirgiannis, and Varlamis, 2003; Taieb, Aouicha, and Hamadou, 2013; Gabrilovich and Markovitch, 2007) introduce knowledge graph based semantic analysis techniques that use lexical ontologies like WordNet (University, 2018), or encyclopedias like Wikipedia(wikipedia, 2018) etc. as external knowledge. Dense low-dimensional representations of items are generated in (Vasile, Smirnova, and Conneau, 2016; McFee, Barrington, and Lanckriet, 2012; Barkan and Koenigstein, 2016; Oord, Dieleman, and Schrauwen, 2013) using neural network models to better capture the semantic similarities based on content information like audios, text descriptions and tags etc. In our opinion, the knowledge based RS category in (Aggarwal, 2016b) can be viewed as a subcategory of content based methods.

The learning process of users' profiles can be treated as a binary classification task that classifies items into interesting ones and uninteresting ones. Different classifier models can be trained on basis of plenty of positive and negative item examples. (Garcia and Amatriain, 2010) follows a heuristic way to model the user preferences by weighting item features based on popularity. (Billsus and Pazzani, 2000; Gemmis et al., 2008) applied probabilistic classifier to efficiently infer user interests. (Ahn et al., 2007; Balabanović and Shoham, 1997) adopted the Rocchio's algorithm to progressively refine the model based on users' relevance feedback on previous recommended items. Nearest neighbors methods are used in (Billsus and Pazzani, 2000; Middleton, Shadbolt, and Roure, 2004) in the user profiling process. Rule-based classification model like decision tree is applied in (Kim et al., 2001) to induct the user preference and recommend personalized advertisements.

2.2 Collaborative filtering RS

The term collaborative filtering was first introduced by the Tapestry project (Goldberg et al., 1992) in year 1992. Instead of using well structured features extracted from content information like content based methods, collaborative filtering techniques utilize explicit user-item interaction such as user ratings on movies or implicit ones like user purchase history and click frequencies, to make recommendations. This interaction information is usually encoded in a user-item matrix where each element represents the explicit rating (or the implicit values) of a user on an item. The general idea of CF techniques is that if user A and B have rated similarly on a same group of items, then A's rating on unknown items can be inferred by user B's known rating on such items; or if item C and D have been rated in a similar way by many users, then the unknown rating of a user on item C can be roughly predicted using his own rating on item D.

2.2.1 Advantages and drawbacks of CF based methods

Comparing to pure content based techniques, CF methods have several advantages:

- The user-item interaction information for CF methods is much easier to obtain. Even when some data is missing, it can still be inferred by known ratings of other users. However, if the content information of some items is not available, content based algorithms will keep these items out of consideration for recommendations.
- Usually it is necessary to introduce knowledge of domain experts to content based methods to get item representations. In case of CF techniques, the recommendation is based on the similarity analysis between users and items, which does not involve any expert supervision.
- The recommendation quality of CF methods outperform pure content based ones regarding serendipity, novelty and diversity (Desrosiers and Karypis, 2011). As mentioned previously, content based methods can not provide any unexpected recommendation for a user. However, CF algorithms do not suffer from this limitation since the various preferences of neighbors are used for the inference of possible recommendations for the active user.

In spite of the popularity of CF techniques in academic researches and real-world applications, this family of methods face several severe problems (Fakhfakh, Ammar, and Amar, 2017).

- Besides the New User problem mentioned in the Content based method section, CF techniques also suffer from the New Item problem. New items that should be appropriately recommended to users are regularly provided to RS. CF methods are not able to recommend these new items until they have been rated by sufficient users. These two problems are also named "Cold-start" problem.
- The sparsity of explicit ratings is another severe problem for CF methods. It is normal that only a few users rate many items and the majority of users only rate several few items. CF methods should be able to make accurate recommendations only on basis of the available data.
- Synonymy problem when several items have the same or very similar names of entries.
- Gray sheep problem refers to the phenomenon that the preference of some users deviate significantly from the rest. This type of users can hardly benefit from CF methods.

A trivial solution to these drawbacks is to combine content based methods with CF ones to leverage the advantages of both and avoid their drawbacks. These hybrid methods will be introduced as complementary improved CF methods enhanced with side content information in the following related works instead of using an individual section.

There are many taxonomies of CF algorithms, the widely accepted way is to classify these CF techniques into two groups: Memory based CF methods and Model based CF methods (Desrosiers and Karypis, 2011; Shi, Larson, and Hanjalic, 2014; Aggarwal, 2016a). Besides this rough binarized categorization, a more detailed classification is given in (Zhu et al., 2018a) to illustrate the progressive differences among various CF methods in a spectral diagram. The traditional taxonomy is used to categorize all CF methods in this thesis. (Zhu et al., 2018a) gives a comparison between memory based and model based CF methods as shown in Table 2.1. In the following subsections, each type of CF methods will be introduced.

TABLE 2.1: Main advantages and disadvantages of the memory-based and model-based approaches.

Memory-based approaches	
Advantages	Disadvantages
It does not require an initial time to create a model	Worse accuracy than model-based methods
Simplicity of algorithms	Worst recommendation times
Recommendations are always updated	
It does not require resources to store the model	
Ease to provide explanation of recommendations	
Model-based approaches	
Advantages	Disadvantages
Better accuracy than memory-based methods	Time needed to create the model
Better recommendation times	Time needed to periodically update the model
Additional possibilities from the model (clustering, analytics, etc.)	Recommendations are based on an outdated model
	It requires resources to store the model
	Complex algorithms.

2.2.2 Memory based CF methods

Memory based CF methods, also referred as neighborhood based Methods (Aggarwal, 2016a; Desrosiers and Karypis, 2011; G. Adomavicius, 2005), have been intensively studied in academic community and widely applied in industrial applications. These methods assume that similar users will give similar feedbacks to groups of related items (Aggarwal, 2016a), and they make predictions and recommendations to the active users based on his own explicit/implicit rating record and those of his nearest neighbors. Although memory based approaches are outperformed by recent state of the art model based approaches in terms of prediction accuracies Zhu et al., 2018a, the evolution of this type of collaborative filtering techniques never stops due to its unique advantages.

Memory based CF techniques are further divided as user based CF methods and item based CF methods. As introduced in (Ortega et al., 2018), both types follows four generic steps:

1. Compute the similarity among users/items considering some similarity metric.
2. Find the nearest neighbors of each active user/item.
3. Make prediction of the unknown ratings of the active user/item by aggregating the known rating information of nearest neighbors of active user/item.
4. Select those items with the most positive prediction to recommend to the active user.

There are also some researches (Wang, Vries, and Reinders, 2006; Ghazanfar, Prügell-Bennett, and Szedmak, 2012; Verstrepen and Goethals, 2014) that combine user based CF and item based CF to obtain ensemble approaches.

Similarity metrics for neighborhood search

The performance of memory based CF techniques is heavily dependent on the quality of encountered neighborhood of each specific user, and the adopted similarity metric severely affects the neighborhood search result. As a consequence, the selection of a proper similarity metric is crucial for memory based CF techniques.

There are a plenty of similarity metrics proposed in previous research works. These metrics are also classified as user based and item based according to the rating vectors used for the similarity calculation. A list of the most commonly used similarity metrics are implemented in our Java library CF4J (Ortega et al., 2018). If not particularly specified, the user based version of each similarity is introduced as examples. Please note that the corresponding item based versions are available in the CF4J library as well.

Pearson correlation similarity (Resnick et al., 1994) is one of the most popular benchmark similarity metrics that normalizes the effects of means and variances. It is computed according to Eq. 2.1:

$$Pearson(u, v) = \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)^2 \sum_{i \in I_u \cap I_v} (r_{v,i} - \bar{r}_v)^2}} \quad (2.1)$$

Given I_u is the set of items that user u has voted.

Constrained Pearson correlation similarity (Shardanand and Maes, 1995) uses median values in Eq. 2.2 instead of mean of the rating ranges to obtain a similarity measure that is less sensitive to sparse and extreme rating vectors:

$$ConstrainedPearson(u, v) = \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - \tilde{r}_{median})(r_{v,i} - \tilde{r}_{median})}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{u,i} - \tilde{r}_{median})^2 \sum_{i \in I_u \cap I_v} (r_{v,i} - \tilde{r}_{median})^2}} \quad (2.2)$$

Cosine Similarity (Bobadilla, Serradilla, and Bernal, 2010) is another popular similarity metric most used in previous researches. This similarity metric is typically used to measure the correlation between two user rating vectors, as calculated in Eq. 2.3:

$$Cosine(u, v) = \frac{\sum_{i \in I_u \cap I_v} r_{u,i} r_{v,i}}{\sqrt{\sum_{i \in I_u} r_{u,i}^2} \sqrt{\sum_{i \in I_v} r_{v,i}^2}} \quad (2.3)$$

One shortcoming of cosine similarity metric is that it does not take into account the difference of users' rating habits. When rating an item, some users intend to use extreme (highest/lowest) ratings, while some others prefer to rate more mildly. This rating difference largely affects the mean and variance of user rating vectors. Adjusted cosine similarity (Ahn, 2008) is proposed to overcome this drawback in the item based scenario. It compares ratings that are centered on the user means to normalize the rating deviations as in Eq. 2.4:

$$AdjustedCosine(i, j) = \frac{\sum_{u \in U_i \cap U_j} (r_{u,i} - \bar{r}_u)(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U_i \cap U_j} (r_{u,i} - \bar{r}_u)^2 \sum_{u \in U_i \cap U_j} (r_{u,j} - \bar{r}_u)^2}} \quad (2.4)$$

Instead of using ratings directly, Spearman rank similarity (Bobadilla, Serradilla, and Bernal, 2010) uses the rankings of these ratings to capture users' preference. Note that items with the same rating share the same ranking, this metric is calculated according to Eq. 2.5. Spearman rank does not need to normalize rating values since only rankings are required. However, this metric can not obtain satisfying results

when exist too many tied ratings:

$$SpearmanRank(u, v) = \frac{\sum_{i \in I_u \cap I_v} (rank_{u,i} - \overline{rank}_u)(rank_{v,i} - \overline{rank}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (rank_{u,i} - \overline{rank}_u)^2 \sum_{i \in I_u \cap I_v} (rank_{v,i} - \overline{rank}_v)^2}} \quad (2.5)$$

Mean squared difference (MSD) (Desrosiers and Karypis, 2011) is a simple similarity metric that first calculates the mean squared difference of two user rating vectors, and then subtracted by one as in Eq. 2.6. Sometimes the mean absolute difference is also used as an alternative:

$$MSD(u, v) = 1 - \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - r_{v,i})^2}{|I_u \cap I_v|} \quad (2.6)$$

Originally Jaccard Index is used to measure the similarity of two sets based on the division of the cardinality of the intersection and the cardinality of the union of two sets. Jaccard similarity (Bobadilla, Serradilla, and Bernal, 2010) adopted this idea and measure the similarity between two user rating vectors as following in Eq. 2.7:

$$Jaccard(u, v) = \frac{|I_u \cap I_v|}{|I_u \cup I_v|} \quad (2.7)$$

Jaccard metric discards numerical rating information by only considering intersection and union set operations. JMSD (Bobadilla, Serradilla, and Bernal, 2010) combines Jaccard and MSD in order to leverage both the numerical information of the ratings and the useful information extracted from proportions of the common and uncommon ratings between users. This similarity metric is calculated according to Eq. 2.8

$$JMSD(u, v) = Jaccard(u, v) \times MSD(u, v) \quad (2.8)$$

PIP measure (Ahn, 2008) focuses on improving recommendation performance under cold-start conditions where only a small number of ratings are available for similarity calculation for each user. It considers mainly three aspects of the rating information: Proximity, Impact and Popularity, and the final similarity is the multiplication of these three factors as in Eq. 2.9. Proximity considers the numerical difference between user ratings, as well as whether the ratings are in agreement. Impact evaluates the strength of user likeness/dislikeness towards a certain item. Popularity further adjusts the similarity between users based on the deviations of their ratings from all users' average rating. The details of these three heuristics are given in Figure 2.2.

$$PIP(u, v) = \sum_{i \in I_u \cap I_v} Proximity(r_{u,i}, r_{v,i}) \times Impact(r_{u,i}, r_{v,i}) \times Popularity(r_{u,i}, r_{v,i}) \quad (2.9)$$

FIGURE 2.2: A detailed explanation of the heuristics adopted in (Ahn, 2008).

Some contextual information can be obtained if ratings are further divided into positive ones R_{pos} and negative ones R_{neg} . In (Bobadilla, Ortega, and Hernando, 2012) these extra information sources named singularities are computed based on Eq. 2.10 and Eq. 2.11 for each item. When later calculating similarity between user u and v , a commonly rated item can face three situations: both ratings are positive, as case A; both ratings are negative, as case B; and one positive rating and one negative rating, as case C. As shown in Eq. 2.12, the Mean Squared Difference weighted by the corresponding singularity values for each case is first calculated, and the arithmetic average of the three cases is then assigned as the final similarity result.

$$s_{Pos}^i = 1 - \frac{|Pos_i|}{|U|} = 1 - \frac{|\{u \in U | r_{u,i} \in R_{pos}\}|}{|U|} \quad (2.10)$$

$$s_{Neg}^i = 1 - \frac{|Neg_i|}{|U|} = 1 - \frac{|\{u \in U | r_{u,i} \in R_{neg}\}|}{|U|} \quad (2.11)$$

$$Singularity(u, v) = \frac{\sum_{i \in A} [1 - (r_{u,i} - r_{v,i})^2] (s_{Pos}^i)^2}{|A|} + \frac{\sum_{i \in B} [1 - (r_{u,i} - r_{v,i})^2] (s_{Neg}^i)^2}{|B|} + \frac{\sum_{i \in C} [1 - (r_{u,i} - r_{v,i})^2] (s_{Pos}^i)(s_{Neg}^i)}{|C|} \quad (2.12)$$

A new similarity metric improved from PIP named New Heuristic Similarity Model (NHSM) was proposed by (Liu et al., 2014). They first change the three terms in the original PIP metric and replace them by Promixity (Ahn, 2008), Significance (Bobadilla et al., 2012) and Singularity (Bobadilla, Ortega, and Hernando, 2012) concepts to obtain the new PSS metric as in Eq. 2.13. As seen in Eq. 2.14, 2.15 and 2.16, the non-linear sigmod function is adopted to better weight the three terms in PSS. A modified version of Jaccard metric is given in Eq. 2.17. The rating preference of each user is also considered and quantified according to Eq. 2.18. Finally, the NHSM similarity metric is computed by multiplying PSS, modified Jaccard and user rating preference as in Eq. 2.19:

$$PSS(u, v) = \sum_{i \in I_u \cap I_v} Proximity(r_{u,i}, r_{v,i}) \times Significance(r_{u,i}, r_{v,i}) \times Singularity(r_{u,i}, r_{v,i}) \quad (2.13)$$

$$Proximity(r_{u,i}, r_{v,i}) = 1 - \frac{1}{1 + \exp(-|r_{u,i} - r_{v,i}|)} \quad (2.14)$$

$$Significance(r_{u,i}, r_{v,i}) = \frac{1}{1 + \exp(-|r_{u,i} - r_{median}| \times |r_{v,i} - r_{median}|)} \quad (2.15)$$

$$Singularity(r_{u,i}, r_{v,i}) = 1 - \frac{1}{1 + \exp(-|\frac{r_{u,i} + r_{v,i}}{2} - \mu_i|)} \quad (2.16)$$

$$JaccardModified(u, v) = \frac{|I_u \cap I_v|}{|I_u| \times |I_v|} \quad (2.17)$$

$$URP(u, v) = 1 - \frac{1}{1 + \exp(-|\mu_u - \mu_v| \times |\sigma_u - \sigma_v|)} \quad (2.18)$$

$$NHSM(u, v) = PSS(u, v) \times JaccardModified(u, v) \times URP(u, v) \quad (2.19)$$

Neighbor selection approaches

Once the similarities between user/item pairs have been correctly computed, a list of nearest neighbors need to be selected to predict and recommend. In general, there are three choices for this step (Desrosiers and Karypis, 2011).

- Top-N filtering: This is the most popular approach adopted in the majority of the state of the art research works. A proper N should be carefully predefined using cross validation since an improper N damages the performance of memory based CF methods with regard to accuracy and efficiency.

- **Threshold filtering:** Some methods (Herlocker et al., 1999; Zheng et al., 2016) pre-establish a threshold value to filter users/items that are not enough similar to the active one. This approach is more flexible than the top-N filtering one when the density of the distribution of users/items in the corresponding similarity space does not vary too much. Considering the sensibility of the threshold, this parameter is more difficult to set in advance.
- **Negative filtering:** Instead of selecting most similar neighbors based on positive correlations, this approach (Herlocker et al., 1999) uses negative correlations between different users/items to filter out candidates. The selection result of this approach is less accurate than the other two, but it can serve as a useful complement.

Aggregation methods for prediction

As the third generic step, the rating prediction is generated by adopting different aggregation techniques on the known values of the active user/item's neighborhood. (Desrosiers and Karypis, 2011; Levinas, 2014) introduced some commonly used aggregation methods that are also implemented in our CF4J library (Ortega et al., 2018). Both user based and item based equations will be introduced since different data are used to aggregate the ratings.

User based average of neighborhood is the simplest way to aggregate the known ratings of the active user's neighborhood. It is computed as in Eq. 2.20:

$$\hat{r}_{u,i} = \frac{\sum_{v \in N_{u(i)}} r_{v,i}}{|N_{u(i)}|} \quad (2.20)$$

Given $N_{u(i)}$ is the subset of the active user's neighborhood where the neighbors have already rated item i .

User based weighted average of neighborhood is an improvement of the simple average due to its defect that it treats every neighbor equally ignoring the different degrees of similarity. Weights are assigned to the neighbors of the active user in such way that the more similar neighbors are the more consideration for the prediction they provide. It is computed as follows:

$$\hat{r}_{u,i} = \frac{\sum_{v \in N_{u(i)}} w_v \cdot r_{v,i}}{\sum_{v \in N_{u(i)}} |w_v|} \quad (2.21)$$

User based weighted deviation from average of neighborhood considers the different rating preference of users. Some users intend to rate higher than the rest, and some users are more critical and usually give lower ratings. As in Eq. 2.22, the ratings are first

normalized by the corresponding user rating average to solve this problem:

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{v \in N_{u(i)}} w_v \cdot (r_{v,i} - \bar{r}_u)}{\sum_{v \in N_{u(i)}} |w_v|} \quad (2.22)$$

The following aggregation approaches are used for item based methods.

Item based average of neighborhood:

$$\hat{r}_{u,i} = \frac{\sum_{j \in N_{i(u)}} r_{u,j}}{|N_{i(u)}|} \quad (2.23)$$

Given $N_{i(u)}$ is the subset of the active item's neighborhood where the neighbors have a known rating.

Item based weighted average of neighborhood:

$$\hat{r}_{u,i} = \frac{\sum_{j \in N_{i(u)}} w_j \cdot r_{u,j}}{\sum_{j \in N_{i(u)}} |w_j|} \quad (2.24)$$

Item based deviation from average of neighborhood:

$$\hat{r}_{u,i} = \bar{r}_i + \frac{\sum_{j \in N_{i(u)}} w_j \cdot (r_{u,j} - \bar{r}_j)}{\sum_{j \in N_{i(u)}} |w_j|} \quad (2.25)$$

2.2.3 Model based CF methods

This other type of CF methods are based on models and are becoming more and more popular since their preliminary success in the famous Netflix prize competition. Despite the relatively high cost of model learning processes, the prediction time of model based CF methods has been dramatically reduced comparing to memory based CF methods, and the recommendation quality is further improved as well. Unlike memory based CF approaches that have to consider all users/items to find the nearest neighbors, and generate predictions directly from neighbors' information, model based methods first learn the latent factors that encode the intrinsic characteristics of both user and item from adequate training data, and then predict the unknown ratings taking advantage of the learned predictive model. The training data can be either explicit ratings or implicit feedbacks as long as the model can gradually obtain the optimal parameter values in the corresponding latent space. There exists a fundamental assumption for all kinds of model based CF methods: they assume that both the users and items can be represented in a low dimensional latent

factor space learned from the original user-item space. These low dimensional representations are in forms of low rank vectors, and usually are used to model users' preferences on items.

Intensive studies on model based CF methods were carried out in the last two decades (Koren and Bell, 2015; Su and Khoshgoftaar, 2009; Zhang, Yao, and Sun, 2017; Portugal, Alencar, and Cowan, 2018; Kluver, Ekstrand, and Konstan, 2018). We do not pursue to introduce all proposed methods, but some of the most representative ones, categorized based on the adopted machine learning model families.

Evolutionary algorithms

Evolutionary algorithms are first adopted for CF recommendations. These models try to provide simple yet effective meta-heuristic solutions with the help of similarity metrics. Recently (Horvath and Carvalho, 2017) give a survey on evolutionary algorithms applied in RS. The Particle Swarm Optimization (PSO) technique was employed in (Ujjiin and Bentley, 2003) to learn user personal preferences. (Lorenzi, Santos, and Bazzan, 2005) was inspired by the honey bee dance, and conducted the recommendations in a case-based fashion. A similar bio-inspired system named TARS (Bedi and Sharma, 2012) selects a small and best neighborhood based on biological metaphor of ant colonies. As one of the most popular evolutionary algorithms, genetic algorithm (GA) was adopted in various works. (Pascoal et al., 2014) used GA to compose a similarity function for event recommendation. (Bobadilla et al., 2011) gradually refined the weighting parameters for a novel similarity metric using GA. (Ar and Bostanci, 2016) proposed a similar refinement mechanism for three other standard similarity metrics.

Fuzzy models

Fuzzy models are also frequently adopted for model based CF RS. A recent survey (Yera and Martinez, 2017) summarized previous approaches that make use of fuzzy tools, as illustrated in Figure 2.3. Fuzzy clustering algorithms, especially the famous fuzzy c-means clustering algorithm, are sometimes used as a preparation step for later memory based methods. (Chen, Ying, and Gong, 2008) first filled unknown ratings using rough set theory, then computed user similarities based on the results of fuzzy clustering. (Treerattanapitak and Jaruskulchai, 2012) introduced a modified fuzzy c-means to CF by changing the original objective function with an exponential function. (Nilashi, Ibrahim, and Ithnin, 2014) combined their Adaptive Neuro-Fuzzy Inference System with the subtractive clustering and higher order singular value decomposition for multi-criteria CF. (Leung, Chan, and Chung, 2006) proposed a fuzzy association rule technique with multi-level similarities. First they discretized ratings using fuzzy logics, then novel measures like fuzzy support and

fuzzy confidence were integrated into the traditional association rule mining techniques. (Kant and Bharadwaj, 2013) proposed a novel CF model based on fuzzy naïve Bayesian classifier, and combined it with reclusive methods to enhance the recommendation performance.

FIGURE 2.3: Fuzzy model based CF methods (Yera and Martinez, 2017).

Matrix factorization models

Matrix factorization models are very popular in current CF research and have been proved to be efficient and effective. The general idea of these models is to factorize the explicit or implicit user-item interaction matrix into two low rank matrices P and Q that capture, respectively, the latent factors of users and items in a low dimensional latent space.

Matrix decomposition based models The typical matrix decomposition technique, Singular Vector Decomposition (SVD), is usually utilized to obtain P and Q , and the approximation matrix $\hat{R}$. This can be formulated in the following general way as Eq. 2.26 by minimizing the matrix norm error together with a certain regularization term. Normally the Frobenius norm is adopted. (Koren, Bell, and Volinsky, 2009; Kumar, Girase, and Mukhopadhyay, 2015; Ramlatchan et al., 2018) are several interesting surveys that summarize and categorize the existing CF methods based on matrix factorization models.

$$\min_{P^*, Q^*} \sum_{(u,i) \in K} (r_{u,i} - \hat{r}_{u,i})^2 + \lambda (\| \Omega \|_F^n) \quad (2.26)$$

Given $K = \{(u, i) | r_{u,i} \neq \bullet\}$, λ the regularization parameter, n the l_n matrix norm, and Ω the set of parameters.

There are several ways to generate the rating predictions based on the extracted low rank matrices. The trivial way is to directly compute Eq. 2.27 (Sarwar et al., 2002; Webb, 2006).

$$\hat{r}_{u,i} = p_u q_i^T \quad (2.27)$$

(Paterek, 2007) improved the rating prediction by introducing bias b_u and b_i for each user and item in such a way that

$$\hat{r}_{u,i} = \mu + b_u + b_i + p_u q_i^T \quad (2.28)$$

where μ is the global average rating. The state of the art method for the SVD family called SVD++, proposed by (Koren, 2008), incorporates the implicit feedback like the binary matrix that indicates the rated and unrated items, or purchased and unpurchased items for users, as well as the explicit ratings, to improve the prediction accuracy. The predictions are computed according to the following equation:

$$\hat{r}_{u,i} = \mu + b_u + b_i + q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j) \quad (2.29)$$

where $N(u)$ refers to the item set that user u has shown implicit preference by purchase or rental actions. In this equation, the user vector is enhanced by the implicit feedback to better capture the user preference. (Guo, Zhang, and Y-S, 2015) further extends the SVD++ model by incorporating social trust information on real social networks as

$$\hat{r}_{u,i} = \mu + b_u + b_i + q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j + |T(u)|^{-\frac{1}{2}} \sum_{v \in T(u)} z_v) \quad (2.30)$$

where $T(u)$ refers to the group of users that user u trusts. Thus, each user is modeled by combining three latent vectors: the vector $\mathbf{p}$ learned from explicit ratings, the vector $\mathbf{y}$ learned from the set of items that user u has shown implicit interest, and the vector $\mathbf{z}$ learned from the group of u 's trusted users.

The regularization term in Eq. 2.26 depends on the parameters of the model and the adopted l_n Frobenius norm. Usually in order to control the model complexity, the regularization parameter λ is the same for all model parameters. The majority of matrix factorization models apply the l_2 norm (Sarwar et al., 2002; Paterek, 2007; Koren, 2008; Guo, Zhang, and Y-S, 2015). For example, the regularization term in (Sarwar et al., 2002) is as follows:

$$\lambda (\sum_u \|p_u\|^2 + \sum_i \|q_i\|^2) \quad (2.31)$$

In (Ji et al., 2016) the l_1 is adopted to obtain sparser solutions in the following way:

$$\lambda(\sum_u \|p_u\| + \sum_i \|q_i\|) \quad (2.32)$$

Apart from the Frobenius norm, several researches (Mazumder, Hastie, and Tibshirani, 2010; Hastie et al., 2015; Ciliberto, Stamos, and Pontil, 2017) establish their objective function considering the trace norm, also known as nuclear norm or Ky-Fan n -norm. The trace norm is the sum of singular values of the approximation matrix $\hat{R}$. (Srebro and Shraibman, 2005) explain the connections among rank, trace norm and max norm as matrix complexity measures. Based on these connections, (Srebro, Rennie, and Jaakkola, 2005) propose the maximum margin matrix factorization (MMMF), which constrains the norm of P and Q instead of their ranks. It means MMMF allows the factor matrices to have theoretically infinite dimensionality, however, only a few factors have large values, i.e. the number of factors is not limited, but the overall strength of these factors is constrained. This constraint is mathematically equivalent to constricting the trace norm of the approximation matrix $\hat{R}$. MMMF is further extended in (Rennie and Srebro, 2005; Weimer et al., 2008; Kumar et al., 2017).

Many research works intend to put non-negativity constraints on both the factorized matrix R and the obtained factor matrices P and Q in order to enhance the interpretability of the learned model. (Lee and Seung, 1999) establish the objective function as in Eq. 2.33 to learn the part-based data representations.

$$\min_{p^*, q^*} \sum_{(u,i) \in K} (r_{u,i} - \log p_u q_i^T) - p_u q_i^T \quad (2.33)$$

Subject to the constraints that R , P and Q are all non negative matrices.

Later in their extending work (Lee and Seung, 2001) propose two alternative objective function for Non Negative Matrix Factorization (NMF) with the same non-negativity constraints:

$$\min_{p^*, q^*} \sum_{(u,i) \in K} (r_{u,i} - p_u q_i^T)^2 \quad (2.34)$$

$$\min_{p^*, q^*} \sum_{(u,i) \in K} r_{u,i} \log \frac{r_{u,i}}{p_u q_i^T} - r_{u,i} + p_u q_i^T \quad (2.35)$$

Similar to Eq. 2.26, Eq. 2.34 minimizes the sum of squared errors. Eq. 2.35 adopts the asymmetric divergence of R from $\hat{R} = PQ^T$. And when $\sum R = \sum \hat{R} = 1$, it reduces to Kullback-Leibler divergence. In this sense, R and PQ_T can be seen as normalized probability distributions. It is proved that NMF with KL-divergence is equivalent to PLSA (Ding, Li, and Peng, 2006).

Considering the previous two versions of objective function using the Frobenius

norm and divergence, (Cai et al., 2011) introduce graph regularization terms into both equations to leverage both the nonnegative constraints and the local invariance assumption in manifold learning theory (Eq. 2.36 and 2.37).

$$\min_{p^*, q^*} \sum_{(u,i) \in K} (r_{u,i} - p_u q_i^T)^2 + \frac{\lambda}{2} \left(\sum_{u \in U} p_u^T p_u d_{u,u} - \sum_{u,v \in U} p_u^T p_v w_{u,v} \right) \quad (2.36)$$

Given a graph with $|U|$ nodes, $W_{|U| \times |U|}$ a weight matrix, $D_{|U| \times |U|}$ the Laplacian matrix of the graph.

$$\min_{p^*, q^*} \sum_{(u,i) \in K} \left(r_{u,i} \log \frac{r_{u,i}}{p_u q_i^T} - r_{u,i} + p_u q_i^T \right) + \frac{\lambda}{2} \sum_{u \in U} \sum_{v \in U} \sum_{k=1}^K (p_{u,k} \log \frac{p_{u,k}}{p_{v,k}} + p_{v,k} \log \frac{p_{v,k}}{p_{u,k}}) w_{u,v} \quad (2.37)$$

Given K the rank of P and Q .

(Luo et al., 2014) integrate the Tikhonov regularizing terms into Eq. 2.34 to get Eq. 2.38 in order to improve the accuracy of recommendations. They also propose novel non-negative single-element-based update rules, which only iteratively update the involved factor vector in P and Q rather than the whole matrices, and significantly improve the efficiency of model training.

$$\min_{p^*, q^*} \sum_{(u,i) \in K} (r_{u,i} - p_u q_i^T)^2 + \lambda_p \sum_u \|p_u\|_F^2 + \lambda_q \sum_i \|q_i\|_F^2 \quad (2.38)$$

Subject to $P \geq 0, Q \geq 0$.

Once the objective function is established, the model starts to learn from observed data adopting a certain learning algorithm. Gradient-based algorithms, especially Stochastic Gradient Descent (SGD), are usually used to iteratively train the proposed model due to their easy implementation. The prediction error $e_{u,i} = r_{u,i} - \hat{r}_{u,i}$ for each observed rating is used to update the parameters of model according to a learning rate γ until convergence. Take Eq. 2.27 as an example, the parameters p_u and q_i are updated in following way:

$$\begin{aligned} p_u^{j+1} &= p_u^j + \gamma(e_{u,i}^j q_i^j - \lambda p_u^j) \\ q_i^{j+1} &= q_i^j + \gamma(e_{u,i}^j p_u^j - \lambda q_i^j) \end{aligned} \quad (2.39)$$

Since both P and Q are unknown, optimizing the above non-convex objective functions directly is NP-hard. However, if one of the two matrices is kept as fixed constant, the considered objective function becomes a convex function of the other (Ramlathan et al., 2018). This learning algorithm is called Alternating Least Squares (ALS), which alternatively update each parameter when the rest are kept fixed until the convergence of the considered objective function. Take the same example as

above, the parameters are updated as follows:

$$\begin{aligned} p_u^{j+1} &= Q^{jT} (Q^j Q^{jT} + \lambda I)^{-1} p_u^j \\ q_i^{j+1} &= P^{jT} (P^j P^{jT} + \lambda I)^{-1} q_i^j \end{aligned} \quad (2.40)$$

Comparing to SGD algorithm, ALS has better scalability since instead of iterative updates, it can benefit from parallelism (Zhou et al., 2008; Mllib, 2018). The computation of each component of the matrices can be conducted independently. This advantage is especially valuable when the training observation set is large, like implicit input data.

Probabilistic models/Graphical models Another family of matrix factorization models are based on probabilistic models, both statistical probabilistic ones and Bayesian probabilistic ones. Since these methods can be represented by graphical models, this type of matrix factorization techniques are also called graphical model based CF methods. (Mnih and Salakhutdinov, 2008; Salakhutdinov and Mnih, 2008; Ling, Lyu, and King, 2014) Probabilistic matrix factorization (PMF) was first proposed by (Mnih and Salakhutdinov, 2008) to overcome the bad scalability and cold start problems of memory based CF methods. Compared to other matrix factorization models, PMF enjoys better efficiency and accuracy on sparse and imbalanced datasets. It can be seen as a probabilistic extension of SVD for the partially observed matrix R . This matrix factorization model assumes that the observation noise follows a Gaussian distribution, and both P and Q follow a zero-mean spherical Gaussian prior distribution as well, as indicated in Eq. 2.41. They place improper prior distributions for hyper-parameters in order to update the parameters in a closed form.

$$\begin{aligned} p(R|p, q, \sigma^2) &= \prod_{(u,i) \in K} N(r_{u,i} | p_u q_i^T, \sigma^2) \\ p(P|\sigma_p^2) &= \prod_{u \in U} N(p_u | 0, \sigma_p^2 \mathbf{I}) \\ p(Q|\sigma_Q^2) &= \prod_{i \in I} N(q_i | 0, \sigma_Q^2 \mathbf{I}) \end{aligned} \quad (2.41)$$

The log of posterior distribution over P and Q can be computed based on the corresponding Gaussian likelihood and prior distributions. Since the exact Inference of the log posterior distribution over parameters is given the ratings is intractable, MAP estimation is typically adopted for the model parameters. Maximizing the log posterior distribution with fixed hyper-parameters is equivalent to minimizing the objective function on the right side of Eq. 2.42.

$$\begin{aligned}
-\ln p(P, Q | R, \sigma_P^2, \sigma_Q^2) &\propto \sum_{(u,i) \in K} (r_{u,i} - p_u q_i^T)^2 + \lambda_P \sum_{u \in U} \|p_u\|_F^2 + \lambda_Q \sum_{i \in I} \|q_i\|_F^2 \\
\lambda_P &= \frac{\sigma_P^2}{\sigma_P^2}, \lambda_Q = \frac{\sigma_Q^2}{\sigma_Q^2}
\end{aligned} \tag{2.42}$$

(Mnih and Salakhutdinov, 2008) also suggest to pass the simple linear Gaussian function through a non-linear logistic function $g(p_u q_i^T)$ so as to bound the predicted values into the valid value interval, which better suits the non-linear nature of the rating prediction task.

Instead of conducting MAP estimation, (Salakhutdinov and Mnih, 2008) apply Bayesian inference for the PMF model. They use Markov chain Monte Carlo (MCMC) methods (such as Gibbs sampling) for approximate inference the parameters of the proposed Bayesian probabilistic matrix factorization (BPMF) model. This model preserves both the Gaussian likelihood function for observed ratings and the Gaussian prior distributions over P and Q , and places the conjugate Gaussian-Wishart priors on the user and item hyper-parameters to facilitate later the inference step. Initialized by pre set values, the parameters and hyper-parameters are alternatively sampled from their corresponding conditional distributions given the other fixed. Instead of using MCMC methods, (Nakajima et al., 2013) propose to leverage variational Bayesian methods for the model inference. (Chen, Zhu, and Heng, 2018) improve the variational inference step in a novel distributed form to enhance the scalability and efficiency of BPMF. Both PMF and BPMF face a common drawback: they are sensitive to outliers. Extending the work of (Wang et al., 2012), which conducted a point estimation by the expectation-maximization algorithm, (Wang and Yeung, 2013) do the inference in a total Bayesian treatment. They also adopt a Laplace mixture with the generalized inverse Gaussian distribution as the noise model in order to improve model robustness. (Guo et al., 2017) propose a non-parametric probabilistic matrix factorization model called ALAME, which assumes an asymmetric Laplace distribution for the noise that can be decomposed into a Gaussian distribution and an inverse-gamma distribution. It turns out much easier to carry out variational inference on the decomposed simple distributions.

There are several probabilistic models for NMF as well. Unlike BPMF and some extensions, (Hernando, Bobadilla, and Ortega, 2016) assume that the observed ratings are sampled from a Binomial distribution. Since the inference can not be solved in a closed form, variational inference technique is used to conduct in a totally Bayesian way. Document recommendation or news recommendations, as a text-based application of CF techniques, partially but intrinsically satisfy the non-negativity constraints since the fully observed matrix is usually the co-occurrence matrix in the form of bag-of-words. A plenty of techniques subject to the non-negativity constraints are intensively studied in this filed. PLSA (Hofmann, 1999) regard latent

factors as topics, and assume the observations of word-document co-occurrences follow a mixture of conditionally independent multinomial distributions. LDA (Blei, Ng, and Jordan, 2003) further extends PLSA with a Bayesian approach to avoid the overfitting problem.

Factorization machine models There is another family of factorization methods called factorization machines (FM) (Freudenthaler, Schmidt-thieme, and Rendle, 2009; Rendle, 2010) that regards user and item latent vectors as equal variables and leverage the advantages of both Support Vector Machine (SVM) and matrix factorizations. In collaborative filtering scenario, traditional SVM models usually fail due to the sparsity of user-item interactions. For example, polynomial SVM is typically trained with at most one observation for parameter $w_{u,i}$, and $w_{u',i'}$ will reduce to 0 since no interaction is previously observation, which makes polynomial SVM degenerate to linear SVM. Unlike SVM, FM models all user-user, item-item and user-item interactions using factorized parameters to break the independence of parameters. Eq. 2.43 defines a general factorization machine when degree equals 2. In this equation, V refers to the whole set of variables represented as $n \times k$ latent vectors, which includes users, items as well as plenty of side information such as time, meta data etc.

$$\hat{r}(x) = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n v_i v_j^T x_i x_j \quad (2.43)$$

FM associates each feature with only one latent vector to model the latent effect between features. FFM proposed by (Juan et al., 2016) further integrates the field information into FM, obtains several latent vectors for each feature, and selects the corresponding vector to the other one for the inner product of each feature pairs. The field information is integrated as shown in Eq. 2.44.

$$\hat{r}(x) = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n v_{i,f_j} v_{j,f_i}^T x_i x_j \quad (2.44)$$

Given f_i and f_j the corresponding feature indexes for i and j .

Neural network models

Besides the popular matrix factorization CF techniques, neural network models are treasured by researchers of RS community as well, and are intensively studied as another important state-of-the-art branch, especially in the last decade when deep learning models (neural network models with hierarchical complicate structure) are proved to be extremely effective in many domains like image/speech recognition, natural language processing etc. The impressive performance superior to human

ability in certain tasks encourages many researchers to apply these promising techniques to constantly update the state-of-the-art frontier. Comparing to other families of model based CF techniques, neural network based ones can more easily capture the intrinsic non-linear and non-trivial user-item interactions, and obtain different levels of abstraction by means of complex representations achieved by hierarchical network structure. Recently a survey (Zhang, Yao, and Sun, 2017) gives a deep systematic insight review on RS techniques that integrate different shallow or deep neural network models, and a two-dimension scheme for classification of deep learning based RS is also provided as shown in Figure 2.4. The low dimensional representation assumption still holds for this family of RS methods since the deep learning techniques are also recognized as the representation learning techniques. In general these methods based on neural network models need to achieve two objectives: representation learning and rating/ranking prediction, either jointly or sequentially. Different neural embedding techniques, such as word2vec (Mikolov et al., 2013), GloVe (Pennington, Socher, and Manning, 2014), doc2vec (Le and Mikolov, 2014) and StarSpace (Wu et al., 2018), are typically applied to learn the distributed representations (Rumelhart, Hinton, and Williams, 1986) of item descriptions in a shadow window to leverage the local contextual information of target item based on explicit ratings and side information like text descriptions. A series of publications such as Item2vec (Barkan and Koenigstein, 2016), Prod2Vec (Grbovic et al., 2015) and Meta-Prod2Vec (Vasile, Smirnova, and Conneau, 2016), are inspired by adopting embedding techniques developed from Skip-Gram with Negative Sampling (Mikolov et al., 2013) to recommendation field. As for the rating/ranking prediction task, both neural network models and previously introduced latent factor models like matrix factorization can be iteratively trained when the learned low dimensional representations are fed into them. According to whether there are interactions between the representation learning and prediction components, the neural network based models can be further categorized into *loosely coupled methods* and *tightly coupled methods* (Wang, Wang, and Yeung, 2015; Zhang, Yao, and Sun, 2017). As shown in Figure 2.5, *loosely coupled methods* sequentially conduct representation learning and prediction as a uni-direction work flow. On the contrary, *tightly coupled methods* alternatively and iteratively optimize the parameters of the neural network model for representation learning and the parameters of the applied neural network or other latent factor model for the prediction task.

In Figure 2.4 eight recent promising neural network models are listed on the left side, namely Multilayer Perceptron (MLP), Autoencoder (AE), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Deep Semantic Similarity Model (DSSM), Restricted Boltzmann Machine (RBM), Neural Autoregressive Distribution Estimation (NADE) and Generative Adversarial Network (GAN) (Zhang, Yao, and Sun, 2017). These eight models and their variants are the most influential representatives of the famous deep learning models. These models can also be

FIGURE 2.4: Two-dimension scheme for classification of deep learning based RS. The left part illustrates the neural network models, and the right part illustrates the integration models. (Zhang, Yao, and Sun, 2017).

categorized into generative, discriminative and hybrid models; or supervised, semi-supervised and unsupervised models. Since the deep learning techniques have been

FIGURE 2.5: Loosely and tightly coupled methods.

TABLE 2.2: A summary on number of publications using different neural network models.

	Recommend Solely on Learning	Rely on Deep	Integrate Deep Learning with Traditional RS	
			<i>loosely coupled meth- ods</i>	<i>tightly coupled meth- ods</i>
Model Using Single Deep Learning Technique	MLP (9) AE (9) CNN (3) RNN (18) DSSM (3) RBM (5) NADE (2) GAN (0)		MLP (5) AE (8) CNN (6) RNN (3) DSSM (0) RBM (0) NADE (0) GAN (1)	MLP (1) AE (8) CNN (8) RNN (1) DSSM (0) RBM (2) NADE (0) GAN (0)
Deep Com- posite Model	8		2	0

intensively investigated in recent years, and their applications in recommendation are usually hybrid models, it will take too much space to explain the details. Besides it is out of the scope of this thesis to exhaust all these techniques, even their applications to RS filed, in this section we will not give the detailed network structure of corresponding models or their mathematic formulations. In Table 2.2 we summarize the number of recent publications on RS based on neural network models. For more information of these proposed models we encourage readers to consult (Zhang, Yao, and Sun, 2017). A typical feature for this family of models is that different types of neural networks perform significantly well for some application fields as different side information is available. For example, CNN models are usually used for computer vision because images/videos are grid-based and can benefit from convolution operations; while RNN models are often applied for sequential data like text since the memory mechanism functions better in this scenario.

Following we will give a case study for youtube video recommendation based on neural network models (Covington, Adams, and Sargin, 2016) as an example of this family of methods. Figure 2.6 shows the general architecture of the RS where candidate generation corresponds to representation learning in Figure 2.5, and the ranking refer to the prediction component.

Both the candidate generation and the ranking stage adopt neural network model. Figure 2.7 gives the architecture of the first neural network model. The embedded video watch and search history, and some other context information are fed into this feedforward neural network. These input embeddings, as well as other parameters, are gradually learned by applying the normal gradient descent back-propagation

FIGURE 2.6: Youtube video recommendation architecture.

FIGURE 2.7: Deep candidate generation model architecture showing embedded sparse features concatenated with dense features.

updates. This neural network can be viewed as a generalization of matrix factorization optimized under rank loss. A set of hundreds of approximately relevant candidate videos are generated for each user.

In the ranking stage much more features are provided to the model since the objective is to select the best few videos from hundreds of candidates instead of the whole video corpus which consists of millions. As illustrated in Figure 2.8, the architecture of the ranking model is similar to the previous one, only changing the softmax function to weighted logistic regression in order to assign an independent score to each candidate video. Then the candidates are ranked based on their score to get the top-N recommendations. For the online testing, the expected watch time per video is adopted as the final ranking function.

FIGURE 2.8: Deep ranking network architecture depicting embedded categorical features (both univalent and multivalent) with shared embeddings and powers of normalized continuous features.

2.3 Demographic RS

RS that leverage demographic information about the users are one of the earliest family. They recommend rated or purchased items with positive feedbacks by demographically similar users to the active user considering features like age, gender, position, etc. Generally a classifier is trained with this demographic information to learn the inner association among users. (Zhao et al., 2014) proposed METIS, which makes product recommendation based on matching the users' demographic data extracted from their public profiles with product information learned from microblogs

and text reviews. (Al-Shamri, 2016) compares several user profiling techniques in demographic RS using real world data and show the advantages and disadvantages of each.

There are some severe drawbacks of the pure demographic RS:

- Collecting the demographic information for all users is very expensive in terms of time and labor, which results in that complete demographic data is rarely available.
- There is too much variability in the preferences of each demographic group. For a user, his interest towards a certain item may change under different conditions.
- (Ekstrand et al., 2018) question that demographic RS show a great utility difference towards different user demographic groups. Users from larger groups usually enjoy better utility, and this is unfair for users from smaller groups.

The classical solution to the demographic approach drawbacks is to combine is to combine demographic information with other types of RS.

2.4 Domain-based taxonomy

In previous sections we conducted the RS taxonomy based on the differences of input sources and applied methodologies. Recently, more and more researchers and developers are engaged to RS field, especially the application on specific domains; besides the explicit or implicit user-item interaction data, adequate side information are also available that require appropriate preprocessing or post-processing techniques from other fields such as natural language processing, computer vision, time series analysis etc. (Bobadilla et al., 2013b) survey plenty of literature on diverse RS scenarios such as movie, music, television, books, documents, e-learning, e-commerce, mobile applications in market, tourism, etc. These typical RS application fields are classified into general domains where certain types of information sources are available as introduced below.

2.4.1 Location based RS

Thanks to the popularity of mobile Internet, many valuable data can be collected in a unconstrained way by means of mobile phone devices. Recommendations based on geographical location data is an example. Applications like social networks, map navigator, ridesharing or urban delivery etc. greatly benefit from the availability of

real time location information. Recommendation of location, trajectory, user, event/activity or content are conducted on basis of user's current or historical geographical location information. (Bao et al., 2015; Bothorel et al., 2018) provide deep reviews of location based RS and categorize them by their objectives.

1. Location recommendation task is further divided into Point of Interest (POI) recommendation (Park, Hong, and Cho, 2007; Levandoski et al., 2012; Yang et al., 2017) and Next POI recommendation (Feng et al., 2015; He et al., 2016a; Ying et al., 2018). Take restaurant recommendation as an example, apart from the explicit ratings a restaurant receives and previous consumption history, the geographical distance also plays a very important role in the user's final decision. The next POI recommendation problem is to recommend the successive POI given the user's current position by inferring his latent behavior patterns.
2. Trajectory recommendation, or similar tasks such as trip, path, route or itinerary recommendation, is an extension to POI recommendation by leveraging users' GPS trajectory history or the check-in history via geo-tags to recommend a sequence of (unrepeated) POIs (Yin et al., 2014; Dai et al., 2015; Menon et al., 2017; Cai, Lee, and Lee, 2018).
3. User recommendation can be further sub-categorized into popular user discovery, friend recommendation and community discovery (Bao et al., 2015). Popular users are traditionally discovered as the core nodes of a social network which are intensively connected. In geographical location related area, these popular users are more experienced in sharing and recommending locations, and are often capable to provide recommendations of higher qualities (Ying et al., 2011). Friend recommendation based on location information lies on the intuition that people are more willing to connect with those are geographically close, and people that frequently visit similar locations are more likely to become friends because they potentially share the same tastes (Li et al., 2008; Zheng et al., 2011; Zhao, Ma, and Zhang, 2017). Community discovery with location information (Hung, Chang, and Peng, 2009; Xiao et al., 2014; Liu and Wang, 2017) also assumes that users that have similar trajectory patterns belong to the same community.
4. Event/activity recommendation is intrinsically correlated with location information and can only be conducted when this geographical information is available (Zheng et al., 2010; Yin et al., 2013; Qiao et al., 2014; Yang et al., 2015; Mo et al., 2018). Given the current location and/or previous locations, Beijing for example, possible activities like visiting the Great Wall, eating in the Gui street etc..., and events such as Mayday concert, global marathon etc... can be recommended to a certain user that best match his profile or in a collaborative filtering way.

5. Content recommendation is another problem that benefits from location information in terms of recommendation quality and efficiency. On content distribution platforms like twitter, sina microblog, spotify etc., location together with other implicit information, are used for recommending content like news, images, videos, music etc (Cheng and Shen, 2014; Kazai, Yusof, and Clarke, 2016; Cheng and Shen, 2016).

2.4.2 Social RS

Social RS refer to those that make recommendations based on the social network which contains plenty of valuable information about user-wise relationships. Depending on the designed structure, the social network can be directed or undirected. For example, Facebook maintains the mutual friendship between user pairs, thus the network is undirected. While Twitter typically uses "Follow" relationship (Kwak et al., 2010) to model the directed user links: a user can follow many other user, but not all the followed users will follow back, i.e. the relationships among users are asymmetric. Another directed network structure is constructed based on the trust and distrust relationships applied by Epinions. If a user chooses to trust another user, he is implicitly indicating that they probably share similar tastes or preferences. All these symmetric or asymmetric relationships can be further quantified by analyzing the strength of user-wise links and discovering user communities (Backstrom and Leskovec, 2011; Gupta et al., 2013; Aiello and Barbieri, 2017; Seo et al., 2017). These side information are first encoded into a user-user matrix, and leveraged in two general ways: 1) directly used to conduct user recommendations such as friend recommendation in Facebook or connection recommendation in LinkedIn. 2) combined with other input sources to enhance the quality of item recommendation. Integrating these network link information helps to increase the robustness of RS, especially for CF methods: memory based approaches can benefit in the neighborhood search step, while model based methods can adapt these side information sources like trust into their objective function to boost recommendation performance. This useful information is preserved only in the graph/network structure. Finally tags are another popular information source in social networks obtained from users' collaborative tagging behavior that explicitly express their preferences. (Bogers, 2018) summarize many recommendation methods that incorporate tags into traditional models. Instead of the simple user-item matrix, they extend it to a 3-D user-item-tag tensor in order to provide enriched sources to the recommendation process.

2.4.3 Review based RS

The incorporation of textual review contributed by users on specific movies, music, products, hotels etc. has led to the emergence of review based RS. These valuable

user reviews are important user-item interaction side information, where opinion and emotions reflecting user preferences can be extracted by techniques such as text mining and sentiment analysis etc., which will greatly benefit the enhancement of content based RS and CF RS. There are mainly two ways that reviews are leveraged to learn user preferences: unique opinion source (Poirier et al., 2010) or supplementary information to explicit or implicit ratings (Lei, Qian, and Zhao, 2016; Zhang et al., 2017b). By transforming unstructured textual reviews into structured user/item profiles, traditional content based RS or CF RS can alleviate the famous sparsity and new user problems. When there are sufficient rating data, reviews can be used to cross validate the reliability of rating predictions. (Chen, Chen, and Wang, 2015) classified the state-of-the-art research on review based RS into two families: *review-based user profile building* and *review-based product profile building*. The former is to construct user profiles exploiting textual reviews they generated; and the latter intends to enrich product/item profiles with user comments on the key characteristics. They further divide *review-based user profile building* into three subtypes according to the emphasized user profile types: *term-based profile*, *rating profile* and *feature preference*. (Chen, Chen, and Wang, 2015) also list the eight elements contained in the user reviews (Figure 2.9).

FIGURE 2.9: An illustration of review based RS (Chen, Chen, and Wang, 2015).

2.4.4 Time aware RS

Timestamps accompanying ratings are often automatically collected by RS; they are very informative data that encode temporal factors that affect user preferences. Traditional RS assume that user interests are static during a relatively long period, and what have been learned from user rating/consumption history can reflect his current preference. However, there are some obvious temporal patterns in users' decision

making process that may violate this assumption. Take location recommendation as an example: during weekdays people always visit familiar places, while at weekends they are more interested in visiting novel places for entertainment. It is important to be aware of the temporal dynamics of user preferences; e.g. they may change their music interest and get tired of songs they liked years ago. (Ding and Li, 2005) propose to apply a temporal decay rate and rely more on the recent history data. (Koren, 2009) reports that the simple weighting design (Ding and Li, 2005) is not sufficient to capture the complex temporal factors. (Song, Elkahky, and He, 2016) figure out that the main challenge for introducing time dimension to traditional RS is the high cost of parameter estimation and inference for the extra parameters which make it impractical for large scale applications. They propose a novel architecture based on deep neural network that models the combination of long-term static and short-term temporal user preferences, which significantly outperforms state-of-the-art algorithms. (Campos, Díez, and Cantador, 2014) summarize some research on time aware RS, and put special emphasis on the evaluation of such RS.

2.5 RS evaluation

The evaluation of RS is crucial for both academic and industrial communities: the former needs to prove their novel models outperform the state of the art; and the latter requires fair evaluation metrics to select the most appropriate methods for their applications. In general there are three ways to conduct evaluations: online evaluation (A/B test), offline evaluation and user evaluation. In this thesis we will focus on offline evaluations since these are the most widely adopted form. Generally, the cross validation techniques, such as random sub-sampling, k-fold cross validation, leave-one-out cross validation etc., are employed in the evaluation process. Figure 2.10 shows the general mechanism for cross validation used to generate quality results from the evaluation measures. Several surveys comprehensively discuss this topic (Shani and Gunawardana, 2011) from many aspects. Different properties of RS are considered in the evaluation of recommendation results as listed below.

2.5.1 Accuracy

Since the appearance of RS, the prediction accuracy is valued as one of the most important properties. Those recommendation algorithms that provide more accurate predictions are assumed to be more preferable to users. The prediction error calculation requires known rating/ranking of test data, as a result, the prediction accuracy evaluation is conducted in offline mode.

FIGURE 2.10: RS evaluation process (Bobadilla et al., 2013b): 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, the better quality of the proposed recommendation algorithm.

Rating based accuracy

When explicit user-item ratings are fed into the RS, the rating that a user would give to a previously unseen item is predicted. The prediction accuracy is evaluated based on the predicted values and the corresponding real ratings.

The most commonly used accuracy metrics are Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), as shown in Eq. 2.45.

$$\begin{aligned} MAE &= \frac{1}{|K|} \sum_{(u,i) \in K} |r_{u,i} - \hat{r}_{u,i}| \\ RMSE &= \sqrt{\frac{1}{|K|} \sum_{(u,i) \in K} (r_{u,i} - \hat{r}_{u,i})^2} \end{aligned} \quad (2.45)$$

$K = \{(u,i) | r_{u,i} \neq \bullet\}$ is the same as defined in Eq. 2.26. Because different rating datasets may have different valid ranges, these ratings are sometimes first normalized by the range $r_{max} - r_{min}$.

Another problem is when rating distribution is highly unbalanced, the final prediction error will be dominated by few users who have rated many items. In this case, it is better to calculate the error for each user, and then average over all the users, as we implemented in our Java library (Ortega et al., 2018) and shown in Eq. 2.46.

$$\begin{aligned} \text{AverageMAE} &= \frac{1}{|U|} \sum_{u \in U} \frac{1}{|I_u|} \sum_{i \in I_u} |r_{u,i} - \hat{r}_{u,i}| \\ \text{AverageRMSE} &= \sqrt{\frac{1}{|U|} \sum_{u \in U} \frac{1}{|I_u|} \sum_{i \in I_u} (r_{u,i} - \hat{r}_{u,i})^2} \end{aligned} \quad (2.46)$$

User acceptance based accuracy

In some cases, the accuracy is not measured by the prediction error, but the user acceptance of the recommended items reflected as rating or purchase history. For this scenario, items are binarized into relevant and irrelevant ones comparing to a predefined relevance/rating threshold. Users' previous acceptance data for different items are available in the offline evaluation mode. By assuming that previously unaccepted items are not interesting to the user, several metrics such as Precision, Recall and F1, can be defined as follows:

$$\begin{aligned} \text{Precision} &= \frac{1}{|U|} \sum_{u \in U} \frac{|\{i \in O_u | r_{u,i} \geq \mu\}|}{|O_u|} \\ \text{Recall} &= \frac{1}{|U|} \sum_{u \in U} \frac{|\{i \in O_u | r_{u,i} \geq \mu\}|}{|\{i \in I_u | r_{u,i} \geq \mu\}|} \\ \text{F1} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned} \quad (2.47)$$

Where O_u is the recommendation items for user u , I_u is rated items by user u , and μ is the predefined relevance threshold.

If the recommendation list is presorted in a declining order, then these metrics can be changed (Eq. 2.48) to evaluate the accuracy emphasizing the first few recommended items. These variants can also be seen as ranking based accuracy metrics, which are introduced later.

$$\begin{aligned} \text{Precision@}n &= \frac{1}{|U|} \sum_{u \in U} \frac{|\{i = O_u(j) | r_{u,i} \geq \mu, j \leq n\}|}{n} \\ \text{Recall@}n &= \frac{1}{|U|} \sum_{u \in U} \frac{|\{i = O_u(j) | r_{u,i} \geq \mu, j \leq n\}|}{|\{i \in I_u | r_{u,i} \geq \mu\}|} \\ \text{F1@}n &= \frac{2 \times \text{Precision@}n \times \text{Recall@}n}{\text{Precision@}n + \text{Recall@}n} \end{aligned} \quad (2.48)$$

Given j the index of the first n items to recommend. There are some other user acceptance based accuracy metrics such as Receiver operating characteristic (ROC) curve, Area Under the ROC Curve (AUC) etc., widely used as well. ROC can be calculated as the ratio between recall (true positive rate) and fallout (false positive

rate, indicated in Eq. 2.49). And AUC refers to its literal meaning.

$$Fallout = \frac{1}{|U|} \sum_{u \in U} \frac{|\{i \in O_u | r_{u,i} < \mu\}|}{|\{i \in I_u | r_{u,i} < \mu\}|} \quad (2.49)$$

Ranking based accuracy

In many recent RS applications, an ordered item list is provided based on predicted sequence of user preference. The reason behind is that user pay more attention on the first few recommended items rather than considering equally all the recommendations. The generated list depends on the numerical rating values or indicated relationship of item pairs. To measure the accuracy of recommendation lists, some ranking metrics are usually used.

The R-score (Breese, Heckerman, and Kadie, 1998) emphasizes on the head of the list by assuming that the attraction of successive items decreases following an exponential decay rate as defined in Eq. 2.50. O is the recommendation list for user u , d is the neutral vote, α is the halflife item index representing the item that has a 50% probability to be visited by the user, and $R_u^{\max}$ is the best possible ranking for user u .

$$\begin{aligned} R_u &= \sum_{i=1}^{|O|} \frac{\max((r_{u,O_i} - d), 0)}{2^{(i-1)/(\alpha-1)}} \\ R &= 100 \frac{\sum_{u \in U} R_u}{\sum_{u \in U} R_u^{\max}} \end{aligned} \quad (2.50)$$

Another widely accepted ranking metric is Normalized Discounted Cumulative Gain (NDCG) (Järvelin and Kekäläinen, 2002) coming from information retrieval. This metric applies a logarithmic decay rate and pays more attention to possible relevant items in the middle of the list. Eq. 2.51 calculates the overall DCG and NDCG for all elements in the recommendation; while Eq. 2.52 only considers the first n elements in the list. The DCG in the equations represents Discounted Cumulative Gain (DCG), $IDCG$ is the ideal DCG , $Gain_{u,O_i}$ is the relevance gain for user u given item O_i , and b is the freedom parameter defined as the logarithmic base.

$$\begin{aligned} DCG &= \frac{1}{|U|} \sum_{u \in U} \sum_{i=1}^{|O|} \frac{2^{Gain_{u,O_i} - 1}}{\log_b(i+1)} \\ NDCG &= \frac{DCG}{IDCG} \end{aligned} \quad (2.51)$$

$$\begin{aligned} DCG@n &= \sum_{i=1}^n \frac{2^{Gain_{u,O_i} - 1}}{\log_b(i+1)} \\ NDCG@n &= \frac{DCG@n}{IDCG@n} \end{aligned} \quad (2.52)$$

Sometimes the availability of rating data cannot be guaranteed since in some recommendation scenarios, only the click through rate data is collected. Instead of predicting ratings for item candidates first and then sorting them based on predicted values, these methods skip the pointwise explicit ratings by directly leveraging pairwise or listwise ordering information to generate an ideal sorting.

2.5.2 Properties beyond accuracy

Besides the prediction accuracy of RS, there are many other properties that can be considered too. Depending on the objectives, different recommendation results emphasizing different properties may be appropriate for different users. For example, a curious user may accept some novel movie recommendations that he does not expect, while a conservative user sticks to the most accurate results. When selecting from hotel recommendations, people prefer those with high value for money, and high reliability at the same time. And when someone is tired of listening to pop music all days, he may expect some more diverse recommendations to explore the new interest. The current RS are required not only to fulfill the original task, but also to achieve the users' objectives as much as possible. Many surveys (Herlocker et al., 2004; Ge, D-Battenfeld, and Jannach, 2010; Hurley and Zhang, 2011; Shani and Gunawardana, 2011; Said and Bellogín, 2014; Kotkov, Wang, and Veijalainen, 2016; Kunaver and Požrl, 2017; Kaminskas and Bridge, 2017; Kluver, Ekstrand, and Konstan, 2018) summarize the research efforts on the emerging properties beyond accuracy, each of which will be discussed separately in the following sections.

Coverage

According to (Shani and Gunawardana, 2011; Kaminskas and Bridge, 2017), coverage is usually a two-fold property of entire RS that measure the proportion of users/items on which the system can make effective recommendations. This is due to the unbalanced distribution of user-item interactions: some active users rate many items, and some popular items are rated by many users. The rest only contribute little information. Methods that achieve higher coverage have stronger ability to overcome "cold-start" problem.

The coverage mentioned in (Herlocker et al., 2004; Ge, D-Battenfeld, and Jannach, 2010) is item coverage, or catalog coverage, which refers to the proportion of items which the system succeeds to recommend. The most simple formulation is defined in Eq. 2.53, given O_u is the recommendations for user u .

$$ItemCoverage = \frac{|\{i | i \in \{O_u | u \in U\}\}|}{|I|} \quad (2.53)$$

The other fold of the term coverage refers to user coverage, which measures the proportion of users to whom the system can make recommendations (Bellogín et al., 2013). Sometimes the system is not able to recommend items to users who just rate very few or none items. This measure is a good indicator of how well the method solves the new user problem.

Novelty

Novelty is first inspired from the information retrieval community. It is defined as the ability to recommending items that are previously unknown to the user in the recommendation field (Shani and Gunawardana, 2011; Castells, Hurley, and Vargas, 2015). Some researchers assume the popularity of items is negatively correlated to novelty because they are probably already known by the user. Others calculate certain distances between items rated by the user and items in the recommendation list (Castells, Vargas, and Wang, 2011). There is no widely accepted mathematical formulation for the computation of novelty. Novelty should be considered together with accuracy since recommending irrelevant items can improve novelty, meanwhile it will significantly reduce accuracy.

Serendipity

Serendipity is improved from novelty by combining unexpectedness (Adamopoulos and Tuzhilin, 2015) and usefulness into recommendation results. An example can be helpful to distinguish serendipity from novelty: imagine that a user likes a singer and has positively rated several of his songs, then if the system recommends him some songs of the singer that he has not listened before, it is about novelty; if the systems recommends him some songs from another similar unknown singer, and the user likes them, it is about serendipity. Improving serendipity helps RS to avoid obvious recommendations in CF based methods that may make the user doubt the usefulness of the system, and to solve the over-specification problem in content based methods (Ge, D-Battenfeld, and Jannach, 2010). However, it is still an open question how to judge the quality of the serendipitous recommendations due to the fact that under different contexts, the users' judgment varies and it is highly subjective. What's more, a good serendipitous RS requires a well balanced consideration between novelty, unexpectedness and usefulness since simply emphasizing on unexpectedness and ignoring usefulness will lead to irrelevant and useless recommendation results.

Diversity

Diversity is another important property that has been recently intensively studied (Castells, Hurley, and Vargas, 2015; Kaminskis and Bridge, 2017; Kunaver and Požrl, 2017; Antikacioglu and Ravi, 2017). Imagine that you have rated "Avatar" and "Star War" in some movie website, then the website recommends you 20 more science fiction movies. You will expect some more diverse recommendations like love movies. This is what diversity aims to do. Several proposals are risen for the computation of diversity. (Zhang and Hurley, 2008) encourage the use of distance/dissimilarity function, (Zhang and Hurley, 2009) use concentration curve to model item distribution, and (Bellogín, Cantador, and Castells, 2010) calculate diversity based on entropy. Similar to properties mentioned above, a proper tradeoff between diversity and accuracy should be carefully achieved because user will not be satisfied by some meaningless irrelevant recommendations.

Privacy

RS users have paid increasing attention to their privacy of personal sensitive data. Systems are not allowed to pro-actively or passively leak private data to any third-parties. The recent leakage of Facebook social network data has been maliciously leveraged for inducible recommendation of political election, which directly leads to the ever strictest European data protection act. An increasing number of privacy-preserving RS researches (Aïmeur et al., 2008; Li et al., 2011; Ghazinour, Matwin, and Sokolova, 2016; Badsha, Yi, and Khalil, 2016) are being reported that intent to preserve users' privacy without significant loss of accuracy in recommendations. There are generally three types of techniques to protect users' privacy as listed below:

1. Homomorphic Encryption: By applying some encryption algorithms, the sensitive user privacy data can be transformed into non-sensitive meaningless ciphertext.
2. Perturbation by using Noise Injection: This technique injects randomized perturbation noise data into the user personal data before passing the original data to the system.
3. Differential Privacy: This technique does not allow the inference of the presence or absence of a certain user record so that a user can not guess private data of other users based on his recommendation results.

Some researches such as (Liu et al., 2017) combine the above techniques to generate hybrid models.

Robustness

The robustness of a RS measures its resistance to different types of attacks intending to influence the recommendation results. For example, in order to promote new movies, producers could hire professionals to inject new biased profiles into movie review and recommendation websites like IMDb and DouBan, in such a way that their new movies are promoted and their competitors are under estimated. This operation is often called shilling attack or push/nuke attack (O'Mahony, Hurley, and Silvestre, 2005; Hurley, 2011). Another type of attack is based on the special structure of social networks where some densely connected power users perform like opinion leaders, and if they are enticed to mislead their followers, a great part of the audience will be influenced (Wilson and Seminario, 2013). (Turk and Bilge, 2018) summarize different attack schemes, and they conduct a robustness analysis for many multi-criteria collaborative filtering recommendation algorithms. The reported experimental results show that current RS are highly vulnerable to artificial manipulations. This fact motivates the investigations on attack detection (Chirita, Nejdil, and Zamfir, 2005; Wu et al., 2012; Zhou et al., 2016) and novel recommendation algorithms with higher robustness (Park et al., 2006; Gao et al., 2018).

Utility

Utility of RS, as introduced in (Shani and Gunawardana, 2011; Jannach and Adomavicius, 2016), can be explained in both customers and service providers' viewpoint. The main purpose of the users is acquirement of satisfying recommendations so that they can gain more knowledge, explore new interest, improve travel convenience etc. Utility is a more general measure about the overall judgment on how well the system meets people's requirements, and how useful it is on basis of his satisfaction degree. (Ekstrand et al., 2018) recently warn that the distribution of utility is unbalanced across diverse user groups of varying sizes. Their experimental results using a utility-based metric show that many current RS discriminate relatively smaller demographic user groups by providing less accurate recommendations. And they urge that evaluation protocols that explicitly quantify the degree to which the system satisfies all its users are urgently needed. From service providers' view, the principal objective is to make profits and keep their revenue growing. In this case they can compare the money they make to the expected revenue. Utility can also be represented by some key indicators such as user acceptance rate, user active time per week etc. However, one forbidden thing for the providers is to take advantage of the asymmetric information and manipulate recommendation results to mislead users to their desired directions.

2.5.3 Reliability

Reliability, or confidence (Shani and Gunawardana, 2011) is one of the most important issues of RS, yet still relatively few researches on this topic have been carried out. It measures how reliable the system finds its rating/ranking predictions are. Reliability is related to previously mentioned trust property of RS when trust and distrust relationships among users of social networks are available. In this scenario, apart from the "Word of mouth" opinions, positive/negative feedbacks from close friends or opinion leaders are also considered to further improve the reliability values, but in most cases, only rating information is ready for use. Generally, the recommendation result is enhanced by assigning a reliability score to each prediction, which usually appears in the form of <rating, reliability> pairs. This can benefit users a lot by providing more useful information. For instance, if two restaurants are predicted by the same rating, a user will naturally choose the one with higher reliability. In the case that one movie is given a high rating but low reliability, and another with slightly lower rating but much higher reliability, the user will investigate further for genre and actor information of both before making a decision. Although no research has been reported yet, considering the robustness property in the assignation of reliability is of great importance since the attack profiles generate bias towards item predictions as well as their reliability. As one of the key topics of this thesis, reliability will be comprehensively surveyed in the following sections.

Reliability measures

The CF RS state of the art (Ricci et al., 2011; Bobadilla et al., 2013b; Godoy and Corbellini, 2016; Yang et al., 2016) includes a variety of research fields and application areas (GomezUribe and Hunt, 2016; Wang, Deng, and Yin, 2016; Matuszyk and Spiliopoulou, 2017) where reliability has been relegated to a secondary level. The first papers in this area often used the term 'confidence' and they were located in the classic K-Nearest-Neighbors (KNN) method. This is a conceptually simple method, where recommendations are made by the set of K most similar (Nearest) users (Neighbors) to the active one. The idea behind the method is to recommend items that the active user does not know and that their neighbors have voted positively. Using KNN, confidence can be easily defined in terms of similarity of the neighborhood to the active user or the number of ratings involved in each prediction.

When social networks emerged, RS data was enriched with tags, followers and followed information, as well as with all types of items and users' links. This was the starting point to create RS trust networks (ODonovan and Smyth, 2005; Yuan et al., 2010; Park et al., 2016) obtaining users and items' reputation on the one hand, and

prediction and recommendation trust measures on the other. Confidence/reliability values were obtained from trust and reputation values (Shambour and Lu, 2012; Martinez et al., 2015; Massa and Avesani, 2007; Guo, Zhang, and Yorke-Smith, 2015; Moradi, Ahmadian, and Akhlaghian, 2015) and they were used to improve accuracy (Bellogín and Castells, 2014; Hernando et al., 2013b; Mahdi et al., 2017; Matuszyk and Spiliopoulou, 2017). In RS where social information is available there is a possibility of using this additional information to obtain more reliable measures of confidence and trust. The next step in the reliability research path arrives when the Internet of things popularizes the existing context-aware RS (Adomavicius et al., 2011; Wang, Deng, and Yin, 2016; Yang et al., 2016). Memory-based CF leading exponents are the KNN method and their necessary similarity measures (Bobadilla, Serradilla, and Bernal, 2010; Bobadilla et al., 2013a), whereas the model-based CF leading exponents are the different matrix factorization methods (Hernando, Bobadilla, and Ortega, 2016; Zhang et al., 2017a). Most of the reliability/confidence research has been carried out in the memory-based CF field (Mazurowski, 2013; Moradi and Ahmadian, 2015; Zhang, Guo, and Chen, 2016a), whereas the model-based CF has only occasionally covered prediction reliability (Chang and Hsiao, 2011).

A significant paper (Hernando et al., 2013b) in the CF reliability of predictions field provides a method to obtain specific reliability measures (RM) specially fitting the needs of different specific recommender systems. This method defines positive and negative factors: the greater the value of each positive factor for a prediction, the greater the value of the reliability of the prediction; the greater the value of each negative factor for a prediction, the lesser the value of the reliability of the prediction. This method mathematically combines positive and negative factors in order to compose a prediction's reliability measure. This measure is tested, showing the expected behavior: "the more reliable a prediction, the less liable to be wrong".

Authors of (Moradi and Ahmadian, 2015) provide a set of factors designed to obtain a reliability measure based on trust-aware information. First, they make the trust network of each active user, later the trust-based reliability measure made from (Hernando et al., 2013b) is used to evaluate the quality of the predicted rating, finally they reconstruct the trust networks for those of the users with lower reliability. The reliability measure proposed in (Moradi and Ahmadian, 2015) just works on datasets containing trust-aware information.

To implement a novel matrix imputation method, (Rezaeimehr et al., 2018) provides a reliability measure. (Rezaeimehr et al., 2018) uses its imputation method to estimate missing ratings and to impute them into the ratings matrix. The key idea is to differentiate between reliable and not reliable predictions: only reliable predictions are imputed into the ratings matrix.

Assigning uniform weights to the missing data is not an effective approach and it

fails to keep up with the dynamic nature of online data (He et al., 2016b). Dynamically weighting missing data is analogous to establish a dynamic reliability measure (weight). The fast eALS (Alternating Least Squares) learning algorithm (He et al., 2016b) is based on item popularity and it outperforms MF methods, using an incremental update strategy into the machine learning process. Item popularity learning is typically based on gradient descent, and gradient descent algorithms slow down when item popularity has a tailed distribution (Rendle and Freudenthaler, 2014). A non-uniform item sampler is proposed in (Rendle and Freudenthaler, 2014); it oversamples informative pairs to speed up convergence.

Several papers propose CF reliability measures based on the rating matrix and the KNN method; (Koren and Sill, 2011) predicts a full probability distribution of the expected item ratings, rather than only a single score for an item. One of the advantages this approach brings is a novel method to estimate the confidence level in each individual prediction. (Moradi, Ahmadian, and Akhlaghian, 2015) proposes several confidence (reliability) measures for individual rating predictions. The reliability term is also used in the RS context as a quality measure to weight some other parameter, such as in (Bellogín and Castells, 2014), where they introduce weights for neighbor selection. The entropy concept in RS (Wang, Zhang, and Lu, 2015) can be applied beyond accuracy, to design a reliability measure. Reliability measures can be applied in the item-to-item or user-to-user CF process in order to retrieve visual representations of RS databases (Hernando et al., 2013c; Hernando et al., 2013a). Users or item relations can be shown as edges of tree graphs; reliability of relations can be represented using different edge colors or transparencies. The reliability concept was used to weight the confidence of users and information. This concept has currently been used, in the RS field, to establish a reliability measure to improve accuracy of trust-aware recommender systems (Moradi and Ahmadian, 2015; Park et al., 2016; Yuan et al., 2010). They assign reliability to users for improving the trust network. Tag folksonomy-based RS provide significant information to tackle research on reliability (Godoy and Corbellini, 2016).

In (Mahdi et al., 2017), the Pareto dominance concept (Ortega et al., 2013) is applied to identify non-dominant users. (Matuszyk and Spiliopoulou, 2017) focuses on the same main concept as (Mahdi et al., 2017): Reliability values are obtained from a specific user similarity. In this case reliability values are related to trust networks, Pareto dominance is not used, and the methodology from (Hernando et al., 2013b) has been followed. RS research has been based on improving prediction and recommendation results; nevertheless, there is a set of potential goals that have not been properly covered. In (Jannach and Adomavicius, 2016), the authors provide a framework of RS recommendation goals and purposes; they establish some interesting future challenges and introduce alternative problem formulations. In paper (Lerche, Jannach, and Ludewig, 2016), authors enrich traditional ratings data; they remind users of items they have viewed or consumed in the past. Results show that

recommending only reminders is not the most effective strategy.

In (Zhang, Guo, and Chen, 2016a), they improve product ranking recommendations using prediction uncertainty; they retrieve confidence values from memory-based data and predictions. Using the confidence values, they obtain a posterior rating distribution. (Chang and Hsiao, 2011) is the only paper that, apparently, covers the reliability concept in model-based CF, nevertheless, it centers on the data quality of the input matrix and nominal rating (instead of ordinal rating) management: Far from the objectives of this chapter. We can conclude that reliability/confidence CF research has been mainly associated with social information, particularly where explicit trust networks can be applied.

Reliability research has focused on explicit information that can be easily understood and combined in a wide variety of ways. When we switch research to the model-based dimension we have to deal with latent factors that do not have explicit meanings. Finding solutions to RS challenges (novelty, reliability, recommendations explanation, etc.) by using model-based approaches is much more abstract and difficult than affording these challenges in the memory-based dimension. Nevertheless, commercial RS are directed toward model-based architectures, since they can better handle the huge sizes of their datasets and they can improve accuracy results. It is the researchers' responsibility to provide model-based solutions that go beyond accuracy.

Reliability quality measures

In order to design a suitable reliability quality measure (RQM) we need to understand the different approaches used to establish RM. Several reliability/confidence measures have been proposed, some of them focusing on specific areas: trust, social information, context-aware, etc. KNN based RM are often proposed.

The classic paper (Herlocker et al., 2004) differentiates the terms strength (prediction value) and confidence (reliability value) in a recommendation. They explain confidence as "how sure the recommender system is that its recommendation is accurate". This paper does not provide any specific confidence quality measure (QM). The contribution made by our work, later introduced in Chapter 4, is to take the principles of the classic papers (Bobadilla et al., 2013b; Herlocker et al., 2004; Ricci et al., 2011; Shani and Gunawardana, 2011; Wu, He, and Yang, 2012) and translate them into two formalized, unambiguous and easy to implement RS-RQM.

In (Herlocker, Konstan, and Riedl, 2000) the authors claim that suitable RS explanations of recommendations lead to increased user confidence. They show a simple confidence interval of 1-5 stars as an explanation for movie recommendations; they argue users can benefit from observing these confidence scores. From the user's

point of view, “RS preference ratings are malleable and can be significantly influenced by the recommendation received. The effect is sensitive to the perceived reliability of a RS and, thus, not a purely numerical or priming-based effect” (Adomavicius et al., 2013). This approach is original, and has recently been expanded in (Hernando et al., 2013b). It shows processed information to the user, as reliability evidence. Our approach covers the alternative option: to calculate each reliability value from the rating matrix (Hernando et al., 2013c; Mazurowski, 2013; Moradi and Ahmadian, 2015; Martinez et al., 2015). Reliability values can be used in different phases of RS; e.g. recommendation explanations, improving accuracy, etc.

McLaughlin and Herlocker (McLaughlin and Herlocker, 2004) create a “belief difference distribution” indicating the belief in the prediction values. The intuition is that the less similar a neighbor is to the active user, the less belief we have that the neighbor’s observed rating is the correct rating for the active user. This approach allows us to define a simple RM based on the KNN method. We provide an alternative, in which KNN-RM is taken as the variability of the neighborhood votes of each active user.

Shani and Gunawardana (Shani and Gunawardana, 2011) define confidence in the recommendation as the system’s trust in its recommendations or predictions. It shows the most common measurement of confidence to be the “probability that the predicted value is true, or the interval around the predicted value where a predefined portion of the true values lie”. Our work takes this same reasoning as a hypothesis. Finally, (Shani and Gunawardana, 2011) found that users liked receiving some recommendations of items that they were already familiar with. They determined that to increase user confidence in the system, it is very important to find credible (known) recommendations.

Koren and Sill (Koren and Sill, 2011) predict a full probability distribution of the expected item ratings, rather than only a single score for an item. One of the advantages this approach brings is a novel method to estimate the confidence level in each individual prediction. Mazurowski (Mazurowski, 2013) proposes several confidence (reliability) measures for individual rating predictions. They use the dataset ratings to extract confidence values associated to rating values. Mazurowski (Mazurowski, 2013) also provides a method to obtain the quality of the RM. As far as we know, this method is the best RQM approach published, so we have adopted it as baseline for the RQM we propose in our work. The reliability term is also used in the RS context as a QM to weight some other parameter, such as in (Bellogín and Castells, 2014), where they introduce weights for neighbor selection. From the CF-KNN method, papers (Bellogín and Castells, 2014; Koren and Sill, 2011; McLaughlin and Herlocker, 2004) implement probabilistic functions and weights to infer reliability values. Although their strategy is adequate and it obtains reasonable results, it suffers from generality because their approaches are KNN centered.

The reliability concept was used to weight the confidence of users and information systems. This concept has been currently used, in the RS field, to establish a RM to improve accuracy of trust-aware recommender systems (Moradi and Ahmadian, 2015; Park et al., 2016; Martinez et al., 2015). They assign reliability to users to improve the trust network. Trust and reliability have been combined in several papers. Often, this approach can only be applied to datasets incorporating additional information, such as content-based data or folksonomies. What we propose in this chapter can be applied to any dataset. Hernando et al. (Hernando et al., 2013c) provide a structured related work section focusing on trust, reputation, credibility and reliability/confidence terms in the CF area.

RM can be applied in the item to item or user to user CF processes in order to get visual representations of RS databases (Hernando et al., 2013b; Hernando et al., 2013a). User or item relations can be shown as edges of tree graphs; reliability of relations can be represented using different edge colors or transparencies. The RQM we propose can serve to determine which is the best RM for the visual representations proposed in (Hernando et al., 2013b; Hernando et al., 2013a).

Hernando et al. (Hernando et al., 2013c) define a general RM suitable for any arbitrary CF-RS. It also shows a method for obtaining specific RM specially suited to the needs of different specific RS. This approach effectively comprises the design of new CF-RM. Additionally, to test RM quality, they process the mean absolute error of the set of predictions having a reliability value higher than a threshold.

In order to test RMs' quality, (Matuszyk and Spiliopoulou, 2017) uses confidence curves: plots showing predictions' reliability values versus predictions' errors. Additionally, (Matuszyk and Spiliopoulou, 2017) provides a simple method to evaluate confidence curves. To test (Matuszyk and Spiliopoulou, 2017) quality measure, some simple methods have been put to the test: support and variability for user and item, resampling and injecting noise. Reliability estimation is especially valuable to weight imputation of predictions to the ratings matrix.

2.6 Summary

In this chapter we introduce the foundations of RS. The objective of this comprehensive review is to give readers a full view of the continuous evolution and the latest frontiers in RS community. Researchers, especially new beginners, will benefit from this in-depth orientation. In the following sections, related works on specific topics will be given to further enrich readers' knowledge on concrete RS subfields. The state-of-the-art research works are first surveyed according to the traditional taxonomy: content based RS, collaborative filtering RS and Demographic RS sequentially. The advantages and shortcomings of each type are discussed. Because thesis concentrates on pure and enhanced CF RS, a more detailed classification is conducted

into several hierarchical levels. As introduced in Chapter 1, the incorporation of rich side information has empowered CF RS by integrating both other RS types and promising techniques from other fields. A novel domain-based taxonomy shows us the new trends in RS beyond the user-item rating matrix. Different RS properties that should be fully evaluated are listed later, and as our key topic of this thesis, reliability is separately listed and the recent publications on RMs and RQMs are intensively surveyed.

Chapter 3

Reliability measures for recommender systems

Besides the well known accuracy metrics, providing a reliability value to each prediction and recommendation is very important in current recommender systems: Users should know which recommendations are reliable and which ones are risky. Despite its growing importance, research into collaborative filtering reliability has rarely been developed in the model-based area. This chapter explains a matrix factorization-based architecture and method that provides a reliability value to each prediction/recommendation. The reliability values obtained have been put to the test, and, when applied, they show improvements in prediction and recommendation quality in different recommender systems; additionally, they provide a range of values that are understandable to users. The same content given in this chapter is also available in (Zhu et al., 2018b) under the permission of the publisher Elsevier (Elsevier, 2018).

3.1 Introduction

Recommender Systems (RS) are playing an increasing part in technology to better the lives of every individual. RS popularity is growing fast, and users rely upon RS recommendations more and more every day; accordingly, RS scientists have the responsibility to enrich results by providing a reliability value to each prediction or recommendation. Users should know that some recommendations are more reliable than others; in fact, people like to know this information: Before we buy a product or book a hotel, beyond the averaged recommendation rating information (e.g. one to five stars), we carefully look at the number of opinions or votes other customers have given. We usually prefer a 4-star hotel based on 1000 opinions to a 5-star hotel based on 4 opinions.

In the above example, the number of opinions (or number of votes) plays the role of an easy to understand reliability value: Predictions and recommendation can be defined using the pair <averaged vote, number of votes>. Using Collaborative Filtering (CF) model-based RS, the prediction and recommendation processes are much more complex: Reliability values, as well as prediction values, must be obtained from hidden factors; fortunately, we can show recommendation results in the same simple pair representation: <prediction value, reliability value>. Reliability values will be mainly used to automatically (implicitly) filter the least reliable predictions or to show these reliability values to users and allow them to explicitly filter recommendations. Beyond this main use (beyond accuracy), model-based reliability values will open up some future opportunities such as: Improving cold-start results, explaining recommendations, extending the group recommendations methods, and merging sensor reliabilities with prediction reliabilities on the Internet of things RS.

This chapter proposes a method to assign a reliability value to each prediction and, by extension, to each recommendation. This method is based on an architectural approach, rather than on a new mathematical model or algorithm. Moreover, the proposed method does not require any arbitrary parameters or specific algorithms; it is based on the use of any existing matrix factorization (MF) techniques. Therefore, it can be directly used in MF models, but not in other CF approaches, such as item or user-based cf. The method is based on the hypothesis that accuracy and reliability should be closely related: The more reliable a prediction the more accurate this prediction should be; similarly, experiments are based on this hypothesis to test the validity of the method. The tests' results show that reliable predictions are more accurate than non-reliable ones, and reliable recommendations present more precision.

In summary, we work on the hypothesis that the more suitable a reliability is, the better accuracy results will provide when applied: predictions with higher reliabilities should provide more accurate (less error) results, whereas we expect higher prediction errors on low reliability recommended items. This hypothesis follows the

guidelines of the most relevant papers published in the area; by way of example, in (Shani and Gunawardana, 2011) it is stated: a) “Perhaps the most common measurement of confidence is the probability that the predicted value is indeed true”, b) “We can design for each specific confidence type a score that measures how close the method confidence estimate is to the true error in prediction”, and c) “Another application of confidence bounds is in filtering recommended items where the confidence in the predicted value is below some threshold”.

Into paper (Mazurowski, 2013), the reliability quality measure is tested by analyzing the way in which reliability values are related to prediction errors. In particular the “confidence curve”, which is represented by the reliability values on the x-axis and the prediction errors on the y-axis, is evaluated. (Hernando et al., 2013b) is based on the same principle; it indicates: “This reliability measure is based on the usual notion that the more reliable a prediction, the less liable to be wrong”. Finally, (Herlocker et al., 2004) states: “Evaluations of recommenders for this task must evaluate the success of high-confidence recommendations, and perhaps consider the opportunity costs of excessively low confidence”.

This chapter’s most relevant contribution to RS state of the art is based on providing a measure of prediction and recommendation reliabilities, under the following restrictions: a) Not to rely on the use of additional data to the ratings cast: Social information, demographic data, temporal values, etc., and b) to use the most popular CF model: The MF method. Any reliability measure using data different to the rating matrix data will only be useful for those RS providing this specific type of data (social, geographical, etc.). The proposed method in this chapter is valid for MF-based CF RS. Restriction b) our method (use of MF) allows the proposed solution to be applicable to most modern RS, which are model-based rather than memory-based and make widespread use of MF techniques.

Addressing the attainment of a reliability measure using MF techniques entails a difficulty that, as far as we know, has not been dealt with to date. In contrast to memory-based methods, where heuristic approaches have a place, in MF we are faced with a model-based method where the learning factors are hidden. Heuristic techniques do not seem to apply to factors in which the meaning is not known; a reasonable approach is the application of modern techniques of machine learning.

In this chapter we propose an innovative architectural solution in the context of RS, with the same approach as some of the hierarchical architectural solutions currently used in the field of deep learning: We establish two differentiated levels of abstraction, both supported by MF as the machine learning method and arranged hierarchically. The first hierarchical level will provide us with the reliability associated with each existing rating prediction, while the second level will spread those values towards ratings not made.

As introduced in the Reliability measure section 2.5.3 of Chapter 2, as far as we know, there is no publication covering prediction and recommendation reliability measures on latent factor-based models. Most of this field of research has focused on obtaining reliability measures using memory-based methods, in general, and datasets with social information or taxonomies, in particular. Section 3.2 explains the details of the proposed method; finally, Section 3.3 sets out the experiments carried out to validate the quality of the results.

3.2 A novel reliability quality measure for recommender systems

In this chapter we propose a novel method that is based on the fact that accuracy and reliability should be closely related: We expect to get high accuracies when predictions are considered reliable; on the other hand, if a prediction is not considered reliable, we accept a lower accuracy result. In other words: Users expect high quality results associated with high reliability suggestions, and they only accept low quality results when associated with low (“risky”) reliability predictions.

The architectural solution adopted is based on the usual approach of engineering and physics in which a large group of measurements are taken using different devices, in different environments, and at different times. The set of information: $\langle \text{device, environment, time} \rangle$ is completed with each measurement value obtained, forming the enriched set $\langle \text{device, environment, time, measurement} \rangle$. Since the set of available data has internal relationships, it is possible to obtain additional information through several processes. For example, if we are talking about sensors located in hostile environments and normal environments, we can obtain the probability that a measurement collected in a hostile environment has an error within a margin, which will be different from the probability when the measurement is taken in a normal environment. The same analysis can be established for temporal sample collections, in which the environment conditions vary or the capacities of the sensors themselves degrade.

The key to the above process is the enrichment of the information when there are large amounts of related measurements. The different architectural and mathematical methods to obtain results exploit these sets of enriched information. In our case, the basic set of information contains the ratings of the RS users: $\langle \text{user, item, rating} \rangle$, while the rich set adds the prediction error (not the rating prediction): $\langle \text{user, item, rating, prediction error} \rangle$. Here, different users play the role of different sensors. Users have different opinions about the set of items (diverse ratings), whereas sensors are exposed to environmental conditions (diverse measures). Also, in this example, the operational mechanism of the sensors is replaced by the selected CF method (MF for us).

To complete the picture, sensors often go through a calibration phase, where they measure the deviation of their measurement from a reference of higher quality. In our example we can use an equivalent mechanism: The cross-validation method, with which we get the prediction error:

$$predictionerror = |predictionvalue - realvalue(rating)| \quad (3.1)$$

By its very nature, the cross-validation method applies to the <user, item> pairs where there are ratings, and not to the <user, item> pairs where no ratings are cast.

Starting from a matrix of ratings (from users to items), and by applying CF techniques, we are able to obtain a matrix of rating predictions: $ratings \rightarrow MF \rightarrow ratingpredictions$. Starting from the matrix of rating predictions, and applying cross-validation, we are able to obtain a matrix of error predictions: $ratingpredictions \rightarrow errorpredictions$. We will give the name “known reliabilities (known errors)” to the set of ‘error predictions’, because there is a prediction and its corresponding error for each existing rating.

Each “known reliability” indicates the numerical deviation between measurement (prediction) and reality (the rating). The greater the deviations, the worse the prediction method when applied to the specific dataset. In this way, each prediction error gives us a reliability value. Starting from the ratings, we obtain an abstraction level with the semantic meaning of “known reliabilities”.

Similar to the deep learning architectures based on hierarchical feature extraction, in a first machine learning process, our method starts from the semantic level “known ratings” to obtain “known reliabilities”. Subsequently, a second machine learning process is applied that starts from the semantic level “known reliabilities” to obtain the semantic level “reliability predictions”. Our hierarchical architecture uses MF to implement both machine learning abstraction levels. Figure 3.1 shows the big picture of the proposed method.

Using working RS, data scientists can access the underlying matrix of ratings $R_{users,items}$. From the ratings matrix a model can be created in order to run a modern model-based RS. In this chapter we have used the Non-negative Matrix Factorization (NMF) technique, due to MF’s ability to abstract hidden factors, generalize results and spread inner knowledge.

In a running RS, from the model we can retrieve prediction estimations ($p_{u,i}$) of unrated items; usually, the N highest predictions are chosen as recommendations. The process area in Figure 3.2 is located at the arrow tagged as “MF prediction process”. The architecture to retrieve reliability predictions ($l_{u,i}$) is outlined, in Figure 3.2, at the arrow tagged as “proposed method”.

Cross validation techniques provide a set of prediction error values: One error value for each prediction to a <user,item> rating. Figure 3.2 shows this process at the arrow

FIGURE 3.1: Hierarchical architecture of the proposed method: The big picture.

tagged as “lower abstraction machine learning level”. This is the necessary information to obtain the Mean Absolute Error (MAE): The inverse value of the accuracy. Whereas prediction estimations of items to users ($p_{u,i}$) are obtained for each unrated item (items that the user theoretically does not know), cross validation techniques work on the set of rated items in order to obtain the differences between ratings and predictions.

The proposed method uses the prediction errors as a starting point. To this first machine learning abstraction level (based on NMF), we apply a second machine learning level to spread known prediction errors (prediction errors of the rated items). The result is a set of estimations of the prediction errors (estimated prediction errors of the unrated items). Figure 3.2 shows this process at the arrow tagged as “higher abstraction machine learning level”.

Since we have stated that accuracy and reliability should be closely related, the set of estimated prediction errors provides the appropriate measure of the prediction reliabilities: The lower the estimated prediction error the higher the prediction reliability.

FIGURE 3.2: Hierarchical architecture of the proposed method: Details.

In a $[0,1]$ interval context:

$$\text{predictionReliability}_{u,i} = 1 - \text{predictionError}_{u,i} \quad (3.2)$$

Briefly, the proposed method provides reliability values to the RS predictions and recommendations. The reliability value of each prediction is, by definition, the inverse of the estimated error of this prediction. To estimate predictions errors (non-rated items) we use a NMF stage from the real prediction errors (from rated items); this machine learning high abstraction level is based on a first NMF machine learning stage, where, from each rating, a prediction estimation of the rating is obtained.

The proposed method performance is just the performance of the chosen model-based technique. Usually, it will duplicate the MF processing load when applied to the current RS. As large commercial RS run model-based solutions, they can easily adopt the proposed approach.

The main advantages of the method design are:

- Reliability values derived from a sensible and convincing principle: High reliabilities are related to low prediction errors.
- Very easy to understand, since it is based on architectural principles as opposed to new mathematical approaches.

- There are no heuristic algorithms, new specific arbitrary thresholds or inconvenient parameters.
- Its implementation is simple, since it can be based on existing machine learning libraries.
- Uses abstraction levels, in a deep learning approach, where reliabilities are obtained from previous machine learning results.
- This method is “universal”, in the sense that it only uses ratings or numerical values; therefore, all collaborative filtering RS can use it.

Finally, in a reminder way, MF techniques convert sparse rating matrices into dense factor matrices. Resulting matrices condense the fundamental information of the rating matrices. The number of factors of the factorized matrices is much smaller than the number of items (or users) of the ratings matrices. We can think of each factor as one or several characteristics of the elements that make up the matrix rating; e.g. action movies, young and female viewers, etc. MF factors are “hidden” factors, since we do not know their specific meaning. From the factorized matrix we can predict the value of each not issued rating; i.e. how much does user u value item i : $\hat{r}_{u,i}$; when the most relevant factors of a user fit with the most relevant factors of an item, it means that the user will like the item. Mathematically, the classical MF process can be expressed as follows:

Let R of size $|U| \times |I|$ be the ratings matrix, where U is the set of users and I is the set of items.

Let K be the number of hidden factors we want to use to make the MF.

Let P of size $|U| \times K$ and Q of size $|I| \times K$ be the two matrices that will approximate R into a $\hat{R}$ matrix.

$$R \approx P \times Q^T = \hat{R} \quad (3.3)$$

Each row of P contains the K factors representing each dataset user, whereas each row of Q contains the K factors representing each dataset item. Thus, the prediction of a missing rating from a user i to an item j is:

$$\hat{r}_{i,j} = p_i \cdot q_j^T = \sum_{f=1}^K p_{i,f} \cdot q_{f,j} \quad (3.4)$$

The idea behind the MF is a simple machine learning process, where we try to minimize the $\hat{R}$ error:

$$e_{i,j}^2 = (r_{i,j} - \hat{r}_{i,j})^2 = r_{i,j} - \sum_{f=1}^K p_{i,f} \cdot q_{f,j})^2 \quad (3.5)$$

To minimize the error, we modify $p_{i,f}$ and $q_{f,j}$ to find the most appropriate values. Then we use the gradient descent of both variables:

$$\frac{\partial}{\partial p_{i,f}} e_{i,j}^2 = -2(r_{i,j} - \hat{r}_{i,j})q_{f,j} = -2e_{i,j}q_{f,j} \quad (3.6)$$

$$\frac{\partial}{\partial q_{f,j}} e_{i,j}^2 = -2(r_{i,j} - \hat{r}_{i,j})p_{i,f} = -2e_{i,j}p_{i,f} \quad (3.7)$$

From Eq. 3.6 & 3.7, the iterative updating rules from iteration t to iteration $t + 1$ are:

$$p_{i,f}^{t+1} = p_{i,f}^t + 2\alpha e_{i,j}q_{f,j} \quad (3.8)$$

$$q_{f,j}^{t+1} = q_{f,j}^t + 2\alpha e_{i,j}p_{i,f} \quad (3.9)$$

Where α is a constant whose value determines the approaching speed to the minimum approximation error.

3.3 Experiments

This section will test three main objectives to validate the proposed method:

- Reliability value distributions are not limited to one or several small ranges of the complete interval $[0..\text{MaxReliability}]$. Reliability values are expected to cover a large range of the interval $[0..\text{MaxReliability}]$, following a Gaussian-like or uniform-like distribution.
- Reliability values fulfill their basic principle: “The lower the estimated prediction error the higher the prediction reliability, and vice versa”. The test results must show better accuracy values when high reliability predictions are chosen.
- Choosing high reliability predictions produces some improvements in recommendation quality.

Additionally, we have conducted an experiment to show it is possible to replace the “number of recommendations (N)” parameter with the “reliability value required (b)”. In many real RS it will be more appropriate to request “recommendations better than quality b” instead of requesting “N recommendations”.

The experiments use three well-known RS datasets: MovieLens, FilmTrust and Netflix. Our method only requires ratings, and Table 3.1 shows these datasets’ main parameters. Table 3.2 shows the experimental settings and parameters.

TABLE 3.1: Datasets used to test the proposed method.

	MovieLens 10M	FilmTrust	Netflix
Number of users	71.567	427	480.189
Number of items	10.681	1827	17.770
Number of ratings	10.000.054	11.848	100.480.507

TABLE 3.2: Experimental settings and parameters.

	MovieLens 10M	FilmTrust	Netflix
Model based method for the proposed algorithm	Non Negative Matrix Factorization (NMF)	NMF	NMF
Cross validation Testing & Training	20% test users, 20% test items	10% test users, 10% test items	20% test users, 20% test items
NMF number of factors	6	7	8
NMF number of iterations	50	80	120

3.3.1 Reliability value distribution

In this section we test the suitability of the reliability values. If the reliability values were limited to a narrow range (e.g. [3..4] from a [1..5] range), they could work for computational purposes, but they would not be convincing for users. Users need an understandable set of values: e.g. a [2,5..4,5] recommendation reliability range would be fine, since it is similar to the recommendation reliability values people would expect.

Since the proposed method provides a reliability value for each <user,item> prediction, we have averaged each user and each item prediction reliability value and obtained their probabilities. Figure 3.3 shows the results when the method is applied to each of the datasets in Table 3.1.

The results show similar behavior for the large datasets: MovieLens and Netflix; the interval of reliabilities is centered on the [3,25..3,5] interval for users and items, and the range is understandable for people (2,5 to 4,5 stars as typical reliability values). FilmTrust reliability distribution is good for users. The item distribution shows erratic behavior that can be explained due to the small size of the dataset.

In short, the results show the proposed method provides suitable distributions of prediction reliability values. People can understand and feel comfortable with the reliability values associated with their item predictions and recommendations. RS based on small datasets sizes should be carefully tested before the deployment of this method.

FIGURE 3.3: Distributions of the reliability values obtained for users and items. Datasets: MovieLens (top), FilmTrust (centre) and Netflix (down). X-axis: Reliability values, y-axis: Probability values.

3.3.2 Accuracy vs. reliability

The method proposed in this chapter is motivated by the concept that the more reliable a prediction is, the more accurate this prediction would be. With this in mind we will test the prediction errors obtained when different levels of, a priori, prediction reliability are stated.

The first experiment will process the Mean Absolute Error (MAE) of the predictions. To each prediction we will assign a reliability value 1,2,3,4,5, using the proposed

FIGURE 3.4: MAE values obtained for each possible reliability value in the set 1,2,3,4,5. Datasets: MovieLens (top), FilmTrust (centre) and Netflix (down). X-axis: Discrete reliability values, y-axis: MAE. Black columns: MAE obtained for the corresponding reliability value. Grey columns: MAE of the RS (without any reliability filtering).

method and applying a round function to force integer reliability values (more understandable to users using the typical star representations). Rounding off reliability values will worsen the expected results, but it will show the “worst case” when applied to real scenarios (reliability values shown as a discrete number of stars).

Figure 3.4 shows the results obtained using the datasets in Table 3.1. The black columns represent the MAE obtained for each reliability value (1, 2, . . . , 5) predictions (e.g. black column on x-axis value 5, shows the averaged MAE of all the predictions with an associated reliability value 5). As we expected, the error (MAE) decreases when the reliability value increases (the bigger the reliability value, the better the accuracy result). These results support the validity of the proposed method: The reliability values obtained are not only adequate for people, but they also fulfill the

stated prerequisite (the larger the reliability value the smaller the prediction error).

The gray columns in Figure 3.4 show the whole RS MAE (i.e. not using reliability information). In all three cases (MovieLens on top, FilmTrust in the middle and Netflix on the bottom), the behavior is correct: The lowest reliability predictions 1,2,3 return the highest error (MAE) results, whereas the highest reliability predictions 5 return the lowest error results.

It is important to remember that improving accuracy is not an objective of this work. These accuracy results are valuable because they confirm the prediction reliability semantics: The higher the reliability of a prediction, the lower the probability that this prediction will be wrong.

Figure 3.4 shows the dependency: “Prediction reliability versus prediction error”. We have run an additional experiment to provide more detailed information: “Prediction error versus prediction reliability”. In this case, we are not limited to the 1,2,3,4,5 set of discrete reliability values. Figure 3.5 shows the results of this additional experiment. The predictions have been grouped according to 15 error intervals (x-axis). For each prediction group, the graphs show the averaged prediction reliabilities (y-axis); e.g. the MovieLens dataset’s smaller errors (0.06 and 0.33) correspond to 4.28 and 4.26 prediction reliability average (this is a sign, as expected, of a large quantity of reliability values 4 & 5).

As can be seen in Figure 3.5, all three datasets tested show the expected behavior: The smaller the predictions error, the greater their prediction reliabilities. High error predictions show less regular behavior, due to the higher variance of the group of prediction reliabilities, particularly in the FilmTrust dataset, where the number of ratings and items is small.

3.3.3 Precision vs. reliability

Whereas Section 3.3.2 tests prediction quality, Section 3.3.3 tests recommendation quality. Improving recommendation quality is a more difficult objective than improving prediction quality: State of the art papers report higher improvements in predictions than in recommendations.

In this section we will test recommendation improvements using the precision quality measure. The precision measure returns the proportion of relevant recommendations regarding the total number of recommendations:

$$precision = \#relevantrecommendations / N;$$

where *relevantrecommendations* is the subset of the *N* recommendations where their rating is greater than or equal to a threshold. Using our prediction reliability information, in order to improve precision, it is necessary to:

FIGURE 3.5: Averaged reliabilities values obtained for 15 groups of predictions errors. Datasets: MovieLens (top), FilmTrust (centre) and Netflix (down). X-axis: Averaged errors, y-axis: Averaged reliabilities.

1. Remove non-reliable recommendations.
2. Insert relevant (prediction greater than or equal to the threshold) and reliable recommendations to replace the recommendations removed.

Note that the recommendations inserted will have lower or equal prediction values than the ones removed, since the original N size recommendation set is filled with the N highest prediction values. We expect to compensate this drawback with great success in the veracity of our hypothesis: Reliable recommendations will have

small prediction errors, and relevant reliable predictions will lead to relevant recommendations. Any improvement in recommendation quality (precision) will indicate excellent behavior of the method proposed in this chapter.

Figure 3.6 shows the results of the precision experiment on MovieLens and FilmTrust datasets. Netflix is ruled out due to processing restrictions. Parameter b , in the graphs, represents the reliability value to consider a prediction as reliable: $b = 5$ means we only recommend relevant predictions that have reliability value 5. $b = 4$ means we only recommend relevant predictions with the reliability values 4 or 5. $b = 0$ means we recommend relevant predictions with any reliability value: 0-5; since using $b = 0$ does not filter reliable predictions; it refers to typical CF precision processing and will be used as a baseline.

FIGURE 3.6: Recommendation quality (precision) improvements. b : Reliability value ($b = 0$ is the baseline). Datasets: MovieLens (top), FilmTrust (down). X-axis: Number of recommendations (N), y-axis: Precision results.

The results show an improvement in precision in all the potential scenarios, particularly when applied to: a) High values of N , b) the highest reliability value ($b = 5$), c) MovieLens dataset. These are the expected results, since: a) When the number of

recommendations (N) is high, it is easier to find non-reliable recommendations to remove and to find reliable recommendations to replace the removed ones, b) the higher the reliability value (b), the smaller the prediction errors, and accordingly, replacing relevant recommendations will give a greater probability of being correct (relevant), and c) Using MovieLens there are many more predictions to choose as replacing recommendations; FilmTrust is a small dataset, so there are fewer opportunities to find reliable and relevant predictions to select as replacing recommendations.

Beyond the detailed explanation of the evolution of the graphs, the significant interpretation, in the context of this chapter, highlights the validity of the proposed method. The experiments show the correct behavior of the reliability values not only in testing prediction quality (MAE), but also in increasing recommendation quality (precision).

3.3.4 Reliability vs. number of recommendations

Often, there are not enough relevant and reliable predictions to fill the list of N requested recommendations to a user. This circumstance depends mainly on the nature of the RS, the chosen value for the number of recommendations (N), the chosen value for the reliability threshold (b), the size of the ratings dataset, the prediction method, the model used, and the coverage obtained.

Since we are filtering (removing) non-reliable recommendations to insert (if found) relevant and reliable recommendations, the recommendation coverage in the proposed method is smaller than or equal to the usual recommendation coverage. Figure 3.7 shows the number of relevant and reliable recommendations found (y-axis), regarding the number of requested recommendations (x-axis). If we look at the FilmTrust dataset graph (on the bottom) we realize that:

- By restricting (filtering) relevant recommendations with reliability 5 ($b = 5$) we find less cases (y-axis) than filtering relevant recommendations with reliability 4 or 5 ($b = 4$), and so on. Note that b acts as a threshold (reliability $\geq$ value) in this section, whereas in it was a parameter (reliability = value). In this process, we are selecting the set of N higher predictions greater than or equal to the relevancy threshold, and then we remove non-reliable recommendations from this set; finally we look for relevant and reliable predictions to replace the ones removed. Sometimes, we will not find enough reliable and relevant predictions to reach the requested N recommendations: The higher the threshold b , the lower the number of reliable and relevant recommendations.
- The higher the number of requested recommendations (N , x-axis), the higher the number of relevant and reliable recommendations found (y-axis): When

FIGURE 3.7: Coverage of the proposed method. b : Reliability threshold. Datasets: MovieLens (top), FilmTrust (bottom). Y-axis: Number of relevant (correct) & reliable recommendations, X axis: Number of requested recommendations.

$N_2 > N_1 \Rightarrow$ there are new potential $N_2 - N_1$ predictions to be added as relevant and reliable recommendations.

- The higher the number of requested recommendations (N), the lower the proportion of relevant and reliable recommendations found; e.g. FilmTrust, $b = 4$, $N = 30$ provides 15 recommendations (50%), whereas $N=80$ provides 30 recommendations (37%): The reason for this behavior is the limited number of relevant & reliable available predictions for each user; i.e. The new potential $N_2 - N_1$ predictions to be added (above paragraph) can be covered for some users and cannot be covered for other users.

The MovieLens graph, at the top of Figure 3.7, shows very similar behavior when $b = 3$ or $b = 4$ are applied. This situation tells us that there are very few “reliability star” predictions that are relevant enough to be considered as recommendations. On the other hand, when $b = 4$ is applied the number of recommendations significantly increases, compared to the $b = 5$ threshold. This result ties in with the MovieLens

graph in Figure 3.3, where we can see there are very few 5 “star” reliability items. The FilmTrust graph, at the bottom of Figure 3.7, shows the same behavior as the MovieLens graph, except for the distinction between $b = 3$ and $b = 4$. The differences are explained for the non-similar distribution of the reliability values in both datasets (see Figure 3.3).

An important useful feature of the proposed method consists of the possibility to request “reliable” recommendations. Instead of asking for N recommendations, we can request only reliable recommendations (where ‘reliable’ is specified through threshold b). We can also request a maximum number of reliable recommendations: e.g. using FilmTrust we can request 15 recommendations with reliability $b=5$; the RS will use internally (on average) $N = 40$ (see Figure 3.7).

3.3.5 Comparative results using a state of the art baseline

While previous sections showed the proposed method correct operation, this section tests the superiority of the proposed method with respect to the scheme published in (Hernando et al., 2013b). In this way, we will compare results obtained with (Hernando et al., 2013b) (the baseline) with respect to results obtained with the proposed method. In particular we will analyze the relationship between reliability and accuracy that occurs in both cases. The greater negative correlation between reliability and error (MAE) will indicate the better of the results: the higher the reliability of the prediction the lower its error must be, and vice versa.

Paper (Moradi and Ahmadian, 2015) is based on the method proposed by (Hernando et al., 2013b), however it can not be taken as a baseline for our experiments, because it uses trust information that is only available into a limited number of datasets, such as Epinions or Flixter. Our method just uses the ratings of the users to the items, and therefore offers a universal solution for CF RS.

The proposed method can be applied to any MF-based model: MF, NMF, BNMF, etc. In order to illustrate this concept, we chose BNMF (Hernando, Bobadilla, and Ortega, 2016) as the MF-based model that we will use in this section’s experiments. BNMF is a probabilistic model that allows us to choose the sparsity of the users’ hidden factors. BNMF facilitates important objectives, such as the clustering of users or items (Bobadilla et al., 2017).

Table 3.3 shows the tested datasets, parameters, similarity measure and aggregation approach of the baseline method (Hernando et al., 2013b), parameters of the proposed method and tested quality measure. Chosen parameter values are the usual ones in the state of the art experiments for the selected datasets.

TABLE 3.3: Datasets used to test the proposed method, baseline parameters, proposed method parameters and tested quality measure.

	FilmTrust	MovieLens 10M	Netflix
Model based method for the proposed algorithm	Bayesian non Negative Matrix Factorization (BNMF)	BNMF	BNMF
Baseline	KNN based reliabilities model (KNN)	KNN	KNN
Tested quality measure	Inverse correlation between MAE and reliability (ICMAE and Rel)	ICMAE and Rel	ICMAE and Rel
Cross validation Testing & Training	20% test users, 20% test items	10% test users, 10% test items	20% test users, 20% test items
KNN similarity metric & aggregation approach	JMSD similarity metric & Deviation From Mean (DFM) aggregation approach	JMSD and DFM	JMSD and DFM
KNN neighborhood	40	80	150
BNMF number of factors	6	7	8
BNMF α	0.8	0.8	0.8
BNMF β	5	5	10
Number of iterations	50	80	120

Figure 3.8 and Table 3.4 show results obtained from experiments stated into Table 3.3. It can be observed: a) Both the proposed method and the baseline achieve adequate reliabilities: inverse correlation between reliability values and prediction errors, and b) In all cases there is a significant improvement of the proposed method with respect to the baseline.

3.3.6 Introducing additional information

Current RS combine several information sources in order to improve their results. Although the traditional objective of this strategy is to improve prediction and recommendation results, it may also be valid to improve other objectives, such as novelty, diversity, serendipity, coverage, or reliability. In the present case (reliability), in order to complement the results from the hidden factors of the method itself, we have selected as sources of additional information: a) Number of ratings issued by each user, and b) number of ratings received by each item. It is reasonable to think

FIGURE 3.8: Inverse relation between reliability and Mean Absolute Error. Datasets: FilmTrust (top), MovieLens 10 M (centre) and Netflix (bottom). Y-axis: reliability values normalized to the interval $[0,1]$ X-axis: MAE. mf is the proposed method, implemented using BNMF; knn is the baseline method, implemented using JMSD. The highest (negative) correlation values correspond to the best experimental results. To simplify the visualization of the graphs, only a small proportion of each dataset elements has been represented using dots.

TABLE 3.4: Reliability versus Mean Absolute Error correlation results. The highest (negative) correlation values correspond to the best experiment results.

	FilmTrust	MovieLens 10M	Netflix
Baseline method (Hernando et al., 2013b)	-0.163	-0.115	-0.042
Proposed method	-0.398	-0.266	-0.271
Improvement	0.235	0.151	0.229

that users who have rated more items will have better chances of receiving reliable recommendations. In the same way, intuition tells us that recommendations of highly voted items should be more reliable.

The proposed method implicitly includes this additional information, compressed into the hidden factors. We are now explicitly enhancing the additional information source; accordingly we can expect some prediction improvements, but not a heavy refinement. To achieve a stronger improvement in the quality of the reliability results, it would be necessary to include information sources not based on user ratings. The disadvantage of this type of information sources is their lack of universality: They can only be used in specific subsets of RS and their datasets. In order to determine how to collect this information in an effective way, it is necessary to know the distribution of ratings among users, and also among items. Figure 3.9 shows these distributions in the dataset that we will take as a reference: MovieLens 1M. As expected, most of the users have cast 20-60 ratings, and most of the items have received less than 60 ratings.

FIGURE 3.9: Number of rating probabilities. Black: User probabilities; Grey: Item probabilities. X-axis: Number of ratings; y-axis: Probabilities.

In order to convert the number of user and item ratings to our reliabilities scale 0,1,2,3,4,5, and taking into account that there are cases with hundreds and even thousands of ratings, we have established these simple equations:

$$reliability_{user} = \ln(1 + ratings_{user}) - 2. \text{if}(reliability_{user} > 5), \text{then} reliability_{user} = 5. \quad (3.10)$$

$$reliability_{item} = \ln(1 + ratings_{item}). \text{if}(reliability_{item} > 5), \text{then} reliability_{item} = 5. \quad (3.11)$$

The “1” inside the parentheses avoids the logarithm of zero. These are the cases where an item has not received any ratings or when a user has not issued any ratings (for different datasets to MovieLens). The “2” outside the parenthesis serves to adequately convert the scale of users ratings (Figure 3.9) to the scale of our reliabilities 0,1,2,3,4,5. In the MovieLens dataset, items may have zero ratings, whereas users must have 20 or more ratings; for that reason the reliability equations are slightly different for users and items.

We have three different reliability measures associated with each prediction: 1) Proposed method, 2) number of ratings issued by each user, and 3) number of ratings received by each item. After trying out different reasonable ways to aggregate these three measures, the best quality results were obtained by:

$$reliability_{user,item} = \min(proposedmethod_{user,item}, reliability_{user}, reliability_{item}). \quad (3.12)$$

Assigning as prediction reliability to the lowest reliability obtained in the different methods makes a lot of sense: It serves as a filter to enhance the higher reliability values. We will only assign high reliability values to the predictions that are simultaneously reliable in all methods used. Using this strategy, the prediction quality results obtained are shown in Figure 3.10.

In Figure 3.10 we look for results in which high values of reliability give us low values of error, whereas low values of reliability have associated high values of error. The method proposed in this chapter offers the pattern that best fits the expected behavior, however the “Mixed” method (Eq. 3.12) slightly improves a very important goal: Predictions with associated high reliabilities 3,4,5 present better quality (they offer minor errors). In short, the “mixed” method presents somewhat more adequate behavior in its predictions of high reliabilities, and much less adequate in its predictions of low reliabilities; i.e. compared to the proposed method, the mixed method is somewhat better at indicating the reliability of what users will like, but worse at forecasting the reliability of what users will not like.

With respect to the recommendation quality obtained using the additional information, Figure 3.11 shows the results: The proposed method (gray lines) is the experiment baseline; the mixed method (black lines) shows a better performance (higher precision values) when applied to a low number of recommendations (N). However,

FIGURE 3.10: Prediction quality (Mean Absolute Error) values obtained using several reliability measures. X-axis: Reliability values; y-axis: MAE. Methods: “Proposed” method, “Users”: Number of ratings issued by each user, “Items”: Number of ratings received by each item, “Average”: MF MAE without applying reliability methods, “Mixed”: Aggregation of the “Proposed”, “Items” and “Users” methods (Eq. 3.12)

when a high number of recommendations 90,110 is set, the performance falls below the proposed method. The explanation for this behavior is found in the restrictive nature of the “Mixed” reliability measure: We are taking the smallest reliability measure (Eq. 3.12), so that the high reliability values ($b = 4$, and especially $b = 5$) are scarce and the number of predictions to which these reliability values are assigned are to the same extent. Consequently, it is much more improbable to get a large number of recommendations (high N) when the reliability values are high ($b = 4$ or $b = 5$), and the method used is “Mixed”.

3.3.7 Testing shilling attacks impact

Recommender systems are highly vulnerable to attacks. In this section we test the proposed method’s behavior when subjected to a shilling attack. We will simulate the incorporation of a group of malicious users whose objective is to fraudulently promote some items and degrade others. The desired behavior of our method is that, after the attack, predictions to the attacked items will decrease their reliability. If the attacked items have less reliable predictions, users will question the choice of these items and will also decrease the likelihood that the RS will choose those predictions as recommendations. In short, if the proposed method decreases the

FIGURE 3.11: Recommendation quality (Precision) values obtained using the “Proposed” and the “Mixed” reliability measures. Reliability values $b = 4$, and $b = 5$. X-axis: N (number of recommendations); y-axis: Precision.

reliability values of the attacked items, the impact of the shilling attacks will also decrease.

First, we exposed the MovieLens 1M dataset to a shilling attack. The parameters of the attack are: Total number of attacked items: 20; proportion between promoted items and degraded items: Random; number of malicious users: 743; number of ratings issued by each malicious user (in addition to the attacked item): Randomly between 20 and 40; number of malicious users attacking each item: Randomly between 20 and 50. Malicious ratings were generated using a Gaussian distribution centered on the existing average of the attacked item (RS usually provide users with the average item rating). Malicious ratings have been randomly kept at a margin of -20% to 20% of the attacked item’s existing average. This attack has been designed to promote or degrade recently introduced items in the RS: The attacked items have been randomly selected from the dataset with an inverse probability to their number of ratings.

Chart a) from Figure 3.12 shows the evolution of the prediction reliabilities when the dataset is exposed to different levels of attack: 250, 500 or 750 malicious users. The values show the average prediction reliabilities that non-attacker users get from the attacked items. In the case “No Attacks”, the value shows the average prediction reliabilities that users get from all the items. As it can be seen, the expected behavior is met: 1) Prediction reliabilities decrease when a shilling attack occurs, and 2) As the shilling attack becomes stronger, the reliability obtained from the attacked items decreases.

FIGURE 3.12: Shilling attack: Reliability values (a) and MAE results (b). X-axis: “No Attacks”: MovieLens 1M; “250, 500, 750”: MovieLens 1M attacked by 250, 500 or 750 malicious users.

Chart b) from Figure 3.12 shows the evolution of the Mean Absolute Error (MAE) when the dataset is exposed to different levels of attack: 250, 500 or 750 malicious users. As expected, the quality of predictions decreases as the attack increases. Both charts (a and b) reinforce the underlying concept in this work about the inverse relationship between error and reliability.

In summary, the proposed method behaves adequately in situations of shilling attack, reducing the prediction reliabilities of the attacked items. In this way, the negative impact of the attacks is reduced: Users’ confidence in the attacked items decreases and the probability of recommending the attacked items becomes smaller. The approach presented here is novel and original; combined with other existing methods and algorithms it could be used to detect shilling attacks in RS or to reduce their consequences. An interesting line of research would be to test the method’s ability to detect attacked items, based on the variation of their reliabilities over time.

3.4 Summary

In this chapter a RM for matrix factorization based CF RS is proposed. People increasingly trust RS predictions and recommendations. Users do not realize that some predictions are more reliable than other ones, and RS do not provide us with any configuration parameters to choose the desired degree of reliability of the incoming recommendations. Current RS should incorporate, either implicitly and/or explicitly, the concept of prediction and recommendation reliability.

The proposed method is based on a simple, sensible and plausible hypothesis: “Accuracy and reliability should be closely related: The more reliable a prediction, the more accurate this prediction would be”. The method architecture and the validation experiments are designed following this hypothesis.

The results of the experiments confirm the validity of the method: a) The reliability values assigned to users and item predictions follow suitable distributions b)

Reliable predictions improve the quality of the RS prediction (MAE), c) Reliable predictions improve the RS recommendation quality (precision).

The extra information of each prediction (its reliability value) can be used to request sets of recommendations satisfying a reliability threshold. In many cases it will be more suitable to request “high reliability level recommendations” than to request a fixed number of recommendations.

The proposed method can incorporate additional information to the RS ratings in order to improve the quality of its results. It also has the potential to reduce the consequences of shilling attacks. An interesting line of research would be to test the method’s ability to detect attacked items, based on the variation of their reliabilities over time.

Chapter 4

Evaluation of the quality of reliability measures

As discussed in Chapter 3, users want to know the reliability of the recommendations; they do not accept high predictions if there is no reliability evidence. Recommender systems should provide reliability values associated with the predictions. Research into reliability measures requires the existence of simple, plausible and universal reliability quality measures. Research into recommender system quality measures has focused on accuracy. Moreover, novelty, serendipity and diversity have been studied; nevertheless there is an important lack of research into reliability/confidence quality measures.

This chapter proposes a reliability quality prediction measure (RPI) and a reliability quality recommendation measure (RRI). Both quality measures are based on the hypothesis that the more suitable a reliability measure is, the better accuracy results it will provide when applied. These reliability quality measures show accuracy improvements when appropriated reliability values are associated with their predictions (i.e. high reliability values associated with correct predictions or low reliability values associated with incorrect predictions).

The proposed reliability quality metrics will lead to the design of brand new recommender system reliability measures. These measures could be applied to different matrix factorization techniques and to content-based, context-aware and social recommendation approaches. The recommender system reliability measures designed could be tested, compared and improved using the proposed reliability quality metrics. The same content given in this chapter is also available in (Bobadilla et al., 2018) under the permission of the publisher Elsevier (Elsevier, 2018).

4.1 Introduction

In the Recommender Systems (RS) field, confidence or reliability has been defined as (Herlocker et al., 2004) “How sure the recommender system is that its recommendation is accurate”. Quality metrics are a key factor for researchers in Collaborative Filtering (CF) RS. By combining quality measures (QM) and open datasets, researchers can improve results from previous works. RS researchers have focused on accuracy QM to test their methods and algorithms. Nevertheless, reliability measures (RM) and reliability quality measures (RQM) did not have the importance of accuracy or novelty research.

We claim RM are very important to RS users, since we know prediction and recommendation values have only a relative meaning. Electronic commerce clients usually look up the number of users that have rated products; we prefer a 4-star rated product based on 50 opinions to a 4.5-star rated product based on 2 opinions. In this case, the client naive quality metric is just the number of opinions. Sometimes we check the mass function of opinions: we prefer a 3-star product based on twenty 3-star opinions to a 3-star product based on ten 1-star and ten 5-star opinions.

Following the above examples, an electronic commerce website could design a really simple RM combining both the number of ratings and the inverse of the rating’s standard deviation. Additionally, it could add some other useful information such as the KNN number of neighbors involved in the prediction, content-based information, etc. This electronic commerce website could provide each client recommendation with the pair: <number of stars, reliability value>. Clients would understand this information as a set of their friends recommending some films: “we believe you will really love film A, but we are pretty sure you will like film B”; Film A <0.95,0.60> , Film B <0.70,0.92>.

It is necessary to know the difference between a RM and a RQM. The first one assigns reliability values to each <user,item prediction>; the second one applies a testing strategy (such as cross-validation) to obtain the quality of the reliability values, that is the quality of the RM. Figure 4.1 shows these concepts.

As can be seen in Figure 4.1 , RM or reliability methods provide reliability values ($L_{u,i}$). These values can be obtained in the same RS stage that returns predictions, or they can be obtained using separate algorithms or methods. Using machine learning techniques (such as matrix factorization), reliability values could be obtained: a) directly from a modified machine learning algorithm, b) from the generated model (factorized matrices in the MF example). As far as we know there are no RS methods to get prediction, reliability pairs from machine-learning models only based on rating datasets; this is an important open research field that requires reliability quality metrics (such as the ones proposed) to be explored.

FIGURE 4.1: Reliability measures and reliability quality measures.

From Figure 4.1 we can determine that testing reliability quality (vertical arrow) is different to obtaining reliability values (horizontal arrow). Whereas this chapter focuses on testing RS reliability quality, obtaining reliability values from machine learning techniques or models will require specific research and will generate new publications.

Researchers can design a variety of RS-RM: a) based on the RS' nature and its data: content-based, collaborative, demographic, social, location-aware, etc. b) based on their filtering approaches: KNN, matrix factorization, bio-inspired methods, etc. It is necessary to test each proposed RM result, comparing each new approach with the existing ones. This process will lead us, as researchers, to improve the reliability values we provide to users, via their online service companies.

Testing RM requires some general, simple and suitable reliability quality metrics, e.g. for testing prediction and recommendation measures we generally use simple accuracy quality metrics: *MAE*, precision, recall, etc. The reliability quality metrics designed should be general enough to test reliability values coming from social filtering, KNN, matrix factorization methods, etc. They should be simple enough to be universally adopted and they should be based on an acceptable RS concept.

As far as we know, there is a lack of general purpose reliability CF-QM, such as the ones used to test accuracy or novelty: (Mazurowski, 2013) “while it appears to be acknowledged in the literature that an accurate estimation of prediction confidence would be of great use, little systematic research has been published toward

this goal". Early papers use the term "confidence" to define users' trust in the system recommendations (Herlocker et al., 2004; Ricci et al., 2011; Swearingen and Sinha, 2001). Later, some RM were established to assign confidence values associated to KNN (Bobadilla, Serradilla, and Bernal, 2010; McLaughlin and Herlocker, 2004) parameters, (Hernando et al., 2013c; Moradi and Ahmadian, 2015) such as the neighborhood similarities to the active user.

Currently, the term "reliability" is often used. There are a variety of papers that incorporate social network trust relation RM to improve RS accuracy results (Park et al., 2016; Martinez et al., 2015). CF reliability values have also been used to provide RS visual representations (Hernando et al., 2013b; Hernando et al., 2013a). Finally, there are a few papers (Hernando et al., 2013c; Mazurowski, 2013) that provide general purpose CF-RM and a tailored method to test their quality.

In order to focus our proposal, we will briefly review existing solutions, and how they have been refined over the years. Figure 4.2 shows the evolution experienced by the way of measuring RS reliability quality. Graphs a) to e) represent the state of the art, while graph f) outlines the proposal presented in this chapter.

FIGURE 4.2: Historical evolution on the reliability quality field (from a to e) and our proposal (f).

When suitable RQM methods did not exist, RS in operation provided users with explicit values that clients could use as reliability information. The most significant examples are the number of clients rating a product or the number of comments received by an item: Figure 4.2a.

From the year 2000, publications appeared that focused on the way of measuring RS quality (Herlocker et al., 2004; Shani and Gunawardana, 2011; Swearingen and Sinha, 2001). These publications focused on how to numerically achieve the accuracy of predictions and recommendations. They incorporate sections containing general concepts to measure other quality objectives "beyond accuracy", such as novelty, diversity, serendipity and confidence. Indications in these papers for RQM focus mainly on exposing a concept: values of confidence (reliability) and values of accuracy must be related, "Probability that the predicted value is true" (Shani and Gunawardana, 2011); Figure 4.2b.

Currently, two papers have appeared providing the quality of reliability approaches. One of these papers (Hernando et al., 2013c) proposes a method to create CF-RM. To test RM qualities it establishes several reliability thresholds and, using graphs, it proves that "the more reliable a prediction, the less liable it is to be wrong" (Hernando et al., 2013c) : Figure 4.2c. Some other experiments in paper (Hernando et al., 2013c) present the concept of the "confidence curve": prediction reliability values are compared with prediction errors values. The expected behavior is an inverse relationship between reliability and error: Figure 4.2d. Mazurowski (Mazurowski, 2013) shows a method to find the quality of the RM; this method is based on the confidence curve analysis: Figure 4.2e.

Our proposal (Figure 4.2f) is to provide a RQM. Unlike graphs, curves and methods, QM allows us to obtain quality values in a direct, simple and universal way, free of ambiguities and variations in implementation. We will use the method described in (Mazurowski, 2013) as baseline that we will compare with our proposed RQM.

The most significant references on RQM have been introduced in 2.5.3 of Chapter 2. Section 4.2, "Proposal", explains the design considerations of the proposed RQM. Section 4.3, "Fundamentals", formalizes the details of the RQM designed. Finally, Section 4.4, "Experiments": a) introduces the scope and design of the work's experiments, b) defines each RM tested, c) explains the baseline method, d) shows the quality results of both reliability predictions and reliability recommendations, and e) provides a discussion on the most significant results.

4.2 Proposed methods

As shown in Figure 4.2f and explained in Section 4.1, we propose the design of RQM equations, which will improve the existing approaches: explicit information, graphs, confidence curves and methods. We work on the hypothesis that the more suitable a RM is the better accuracy results it provides when applied: predictions with higher reliabilities should provide more accurate (lower error) results, whereas we expect

higher prediction errors on low reliability recommended items. This hypothesis follows the guidelines of the most relevant papers published in the area; by way of example, Shani and Gunawardana (Shani and Gunawardana, 2011) state: a) “perhaps the most common measurement of confidence is the probability that the predicted value is indeed true”, b) “we can design for each specific confidence type a score that measures how close the method confidence estimate is to the true error in prediction”, and c) “another application of confidence bounds is in filtering recommended items where the confidence in the predicted value is below some threshold”.

In paper (Mazurowski, 2013), the RQM is tested by analyzing the way in which reliability values are related to prediction errors. In particular, the “confidence curve” is evaluated. The confidence curve is represented by the reliability values on the x-axis and the prediction errors on the y-axis. (Hernando et al., 2013c) is based on the same principle; it indicates: “this RM is based on the usual notion that the more reliable a prediction the less liable it is to be wrong”. Finally, (Herlocker et al., 2004) states: “evaluations of recommenders for this task must evaluate the success of high-confidence recommendations, and perhaps consider the opportunity costs of excessively low confidence”.

Along the lines of the papers reviewed, the design of the proposed RQM will follow the guidelines in Table 4.1 . We expect accurate RM to combine high reliability values with low prediction errors; this good behavior will be rewarded. On the contrary, high reliability values combined with high prediction errors will be penalized situations. High prediction errors are less serious if they are associated with low reliability values.

TABLE 4.1: Reliability quality measure: penalty and reward situations.

Prediction error	Reliability value	Result	Reliability quality
High	High	Big mistake	Big penalty
High	Low	Hit	Reward
Low	High	Big hit	Big reward
Low	Low	Mistake	Penalty

Below are several principles and considerations that we have taken into account in the design of the RQM proposed in this chapter:

- To avoid non-formal mechanisms and also methods or algorithms: the results should be mathematical equations, which are easy to interpret and use, unambiguous and similar to the existing RS accuracy, novelty or diversity quality measures.
- To return quality improvement results, not absolute values. Results should show improvement or worsening of accuracy (error): improvement of results using reliability information, compared to results obtained without using that information. Thus, a "0.15" quality result indicates that we can improve the

error by 15% using the values of the RM. A "-0.07" quality result indicates that the error worsens by 7%, and therefore the tested RM does not work at all.

- To provide two quality measures: 1) a measure of the reliability prediction quality, and 2) a measure of the reliability recommendation quality. The first one will be called *RPI*: "Reliability Prediction Improvement". The second one will be called *RRI*: "Reliability Recommendation Improvement".
- To avoid arbitrary parameters: using only ratings, predictions and reliability values. The only parameter that we can use is the threshold at which it is decided when a recommendation is relevant (as in the accuracy QM: precision and recall).
- To use only the ratings matrix, without relying on additional information that is not available in all RS datasets: social information, demographic data, context-aware raw data, etc.
- To encourage the stability of the results: processing the data in a uniform way, making use of all the data, avoiding saturation functions, discarding step functions, etc.
- To facilitate the integration of the QM into the current RS, providing simple, unambiguous, easy-to-interpret equations to implement and offering results that can be compared between different RM, collaborative filtering methods, and diverse datasets.

The current commercial RS can benefit from the proposed RQM in several ways:

- Providing users with explicit and accurate reliability values : it is necessary to select the most appropriate RM to the RS dataset. Additionally, the chosen RM must fit with the nature and volume of the RS data. The proposed RQM perform the task of discovering the appropriate RM at each stage of the RS evolution. Explicit and accurate reliability values contribute to increase users' confidence.
- Providing users with implicit and accurate reliability values : in this case, the provided reliability information is non-explicit, e.g. ordering recommendations according to their reliability degree. As in the previous case, the proposed RQM is necessary to choose the correct RM.
- Improving prediction and recommendation accuracies : RQM allow to refine RM results, which make it possible to obtain accurate reliability values. One way to improve RS accuracy is to provide users with only the best predictions and recommendations that also have high reliability values.
- Providing to the RS administrators with an extra monitoring tool : RS administrators analyze, improve and fine tune their systems based on numerical parameters such as accuracy, reliability, coverage, etc. They need accurate quality

measures, such as the *MAE* or the precision one. Currently there is no published RQM; this circumstance makes relevant the proposed RQM.

- Detecting particularly complex predictions and recommendations : e.g. cold start situations, non-standard (Grey sheep) users or items, shilling attacks, etc.

To incorporate this technology into a commercial RS, it must be taken into account that its execution is not carried out online, but rather a batch processing is done. Cross-validation methods, explained into Section 4.4, consume a large amount of execution time when applied to huge datasets. Reliability quality methods such as Mazurovsky's (Mazurowski, 2013) fast resample consume too large amounts of execution time, while our proposed RQM is a lightweight process, based on only the execution of a metric. In summary, the incorporation of our RQM is completely feasible in commercial RS, due to its lightweight measure nature and to that it can be processed in parallel to the operation of the RS itself. Into Section 4.4, Figure 4.4 , we show and explain the implementation main steps of the cross-validation process tailored to our RQM.

The two equations proposed are formalized in the next section: Reliability Prediction Improvement (*RPI*) and Reliability Recommendation Improvement (*RRI*). Both RQM follow the principles and considerations described in the previous paragraphs.

4.3 Formulation of the proposed models

The RQM design strategy will penalize predictions simultaneously showing high accuracy errors and high reliability values. Predictions showing high accuracy errors and low reliability values are not penalized. The underlying principle is: “the higher the prediction reliability value, the higher its accuracy error penalization; the lower the prediction error, the bigger its reliability value reward”.

Penalizations must be relative to the set of reliability values associated with the set of predictions or recommendations. In this way, if all the predictions reliability values were the same (i.e. 0.9 in $[0..1]$) we should not consider high and low values, and we should not apply an absolute penalization factor based on these values. This is the case for RS that do not provide reliability values: they can be considered as RS where their reliability parameters always have their maximum value (i.e. 1.0 in $[0..1]$). The accuracy of this type of RS will not be changed according to their implicit and identical reliability values.

The RQM design strategy must reward RM providing well balanced reliability density distributions, i.e. setting all the $[0..1]$ reliability values within the subinterval $[0.04..0.1]$ does not provide reliability variety to the users. A normal distribution

(or similar) of the reliability values will provide a balanced and diverse set of reliabilities, where values higher than the mean distribution produce greater prediction error penalizations than values lower than the mean.

We define the following parameters (Table 4.2):

TABLE 4.2: Parameters.

$r_{u,i}$	Rating of user u to item i
$p_{u,i}$	Prediction for the rating $r_{u,i}$
$l_{u,i}$	Reliability value associated with $p_{u,i}$
$\bar{l}$	Mean of the reliability values
$e_{u,i}$	Error of $p_{u,i}$, $e_{u,i} = p_{u,i} - r_{u,i} $
$\bar{e}$	Mean of the error values

To implement Table 4.1, we will consider “high” reliability values to be the set of values greater than the mean reliabilities (and “low”, to be the set of reliability values lower than the mean). Similarly, we will consider “high” prediction errors to be the set of values greater than the mean error predictions (“low” to be the set of values lower than the mean). Table 4.1 can be redefined as:

High prediction errors in Table 4.3 (first column) correspond to errors of prediction greater than the mean absolute error (MAE): $e_{u,i} > \bar{e}$, and therefore, the expression $(e_{u,i} - \bar{e})$ is positive; $(e_{u,i} - \bar{e})$ is negative when prediction errors are lower than the MAE. Similarly, high reliability values in Table 4.3 (second column) correspond to reliability values greater than the mean reliability $\bar{l}$: $l_{u,i} > \bar{l}$, and therefore, the expression $\bar{l} - l_{u,i}$ is negative; $\bar{l} - l_{u,i}$ is positive when reliability values are lower than the reliability mean.

TABLE 4.3: Reliability quality measure: Penalty and reward situations.

Prediction error ($e_{u,i} - \bar{e}$)	Reliability value ($\bar{l} - l_{u,i}$)	Result	Reliability quality
Positive	Positive	Positive	Reward
Positive	Negative	Negative	Big penalty
Negative	Positive	Negative	Penalty
Negative	Negative	Positive	Big reward

In order to achieve the results in Table 4.3 (third column) we combine prediction errors and reliability differences:

- $(e_{u,i} - \bar{e})(\bar{l} - l_{u,i})$. We define $RPI_{u,i} = (e_{u,i} - \bar{e})(\bar{l} - l_{u,i})$, as the “Reliability Prediction Improvement” related to the prediction of item i to user u . A suitable reliability value $l_{u,i}$ associated with a prediction $\langle p_{u,i}, l_{u,i} \rangle$ will produce a positive value in $RPI_{u,i}$; wrong reliability values will produce negative values in $RPI_{u,i}$.

- *RPI* , “Reliability Prediction Improvement” is the proposed RQM based on the *MAE* improvement. Eq. 4.1 shows its expression; it returns the average prediction improvement. *RPI* positive values reveal accurate RM, whereas *RPI* negative values reveal inappropriate ones. The higher the *RPI* result, the better the applied RM. The *RPI* equation denominator contains the traditional accuracy *MAE*, and therefore, *RPI* returns the *MAE* improvement value. The $\frac{1}{\theta_e \theta_l |T|}$ term in Eq. 4.1 has the unique function of standardizing the *RPI* result. T is the set of ratings, U the set of users, and I the set of items.

$$\begin{aligned}
 RPI &= \frac{\frac{1}{\theta_e \theta_l |T|} \sum_{u \in U, i \in I | r_{u,i} \neq \bullet} [e_{u,i} (e_{u,i} - \bar{e}) (\bar{l} - l_{u,i})]}{\bar{e}} \\
 \text{where :} \\
 T &= \{r_{u,i} \neq \bullet | u \in U \wedge i \in I\} \\
 e_{u,i} &= |r_{u,i} - p_{u,i}| \\
 \bar{e} &= MAE = \frac{1}{\#T} \sum_{u \in U, i \in I | r_{u,i} \neq \bullet} e_{u,i} \\
 \theta_e &= \frac{1}{\#T} \sum_{u \in U, i \in I | r_{u,i} \neq \bullet} |e_{u,i} - \bar{e}| \\
 \theta_l &= \frac{1}{\#T} \sum_{u \in U, i \in I | r_{u,i} \neq \bullet} |\bar{l} - l_{u,i}|
 \end{aligned} \tag{4.1}$$

Whereas *RPI* shows reliability prediction improvements, a second QM is proposed with the aim of dealing with “Reliability Recommendation Improvements”: *RRI*. *RPI* has been based on the hypothesis that the more suitable a RM is, the better accuracy results it will provide when applied. *RRI* will be based on a similar hypothesis: the more suitable a RM is, the better recommendation results it will provide when applied. We expect to find that relevant recommendations are related to high reliability predictions.

We define:

- Z_u as the set of N recommendations to user u .
- Θ as the threshold to consider that a recommendation is relevant.

Both traditional precision and recall recommendation QM are based on the set of relevant recommendations (recommendations where $r_{u,i} > \Theta$). Precision returns the proportion of relevant recommendations to the number of recommendations (N); recall returns the proportion of relevant recommendations to the number of relevant items. The proposed *RRI* measure takes the precision and recall essential information (relevant recommendations) and tests the reliability values associated to each relevant recommendation.

Eq. 4.2 tests the reliability quality in all the relevant recommendations ($i \in Z_u | r_{u,i} \geq \Theta$) to all users $u \in U$. Relevant recommendations associated with high reliability values ($l_{u,i} > \bar{l}$) produce positive results, whereas low reliability values ($l_{u,i} < \bar{l}$)

produce negative results. The higher the positive results the better the RM. Negative results show bad RM performance.

$$\sum_{u \in U} \sum_{i \in Z_u | r_{u,i} \geq \Theta} (l_{u,i} - \bar{l}) \quad (4.2)$$

Eq. 4.3 shows the *RRI* RQM for recommendations: it returns the gain factor between the quality of the relevant recommendations (Bellogín and Castells, 2014) and the total number of relevant recommendations. The higher the positive results the better the RM. The $\frac{1}{\alpha_l}$ term in Eq. 4.3 has the function of standardizing the *RRI* result.

$$RRI = \frac{\frac{1}{\theta_l} \sum_{u \in U} \sum_{i \in Z_u | r_{u,i} \geq \Theta} (l_{u,i} - \bar{l})}{\sum_{u \in U} \#\{i \in Z_u | r_{u,i} \geq \theta\}} \quad (4.3)$$

The proposed *RPI* and *RRI* RQM can be applied to any type of RS (content-based, collaborative, location-aware, social, etc.) and to any RS method (KNN, matrix factorization, Bayesian, etc.), since they only need a set of pairs $\langle p_{u,i}, l_{u,i} \rangle$ to operate. The $l_{u,i}$ values can be obtained using social network graphs, GPS coordinates, probabilistic values, etc.

4.4 Experimental results and discussions

4.4.1 Introduction

In this section, the proposed prediction RQM (*RPI*) is tested as follows:

1. Applying *RPI* to the results obtained by four RM: knn variability, support for user, support for item and fast resample. These RM are explained in the following subsection.
2. Making use of two classic RS datasets: MovieLens 1M and the Netflix prize dataset.
3. Comparing the *RPI* characteristics and results with the RQM published by Mazurovsky (Mazurowski, 2013) (our baseline).

As explained in the introduction section, Mazurovsky's method is the only one published that provides prediction RQM results testing several RM. Hernando et al. (Hernando et al., 2013c) use a reliability threshold value to verify that above this value the accuracy results improve; this verification cannot be considered as a method or as a RQM. Consequently, the method devised by Mazurovsky is taken as baseline in this chapter for the proposed prediction RQM. As far as we know there is no published recommendation RQM, therefore we cannot use a recommendation

RQM baseline. We will test the recommendation RQM proposed in this chapter (RRI) by applying it to the four selected RM.

Figure 4.3 shows a schematic of the experimentation process: 1) starting from two representative datasets in the RS field, 2) obtaining the prediction reliability values in each of the RM tested, 3) evaluating the RQM of each RM, and 4) comparing the quality of each RM and discussing the advantages and disadvantages of the proposed RQM (RPI) with regard to the baseline (Mazurovsky's method).

FIGURE 4.3: Experimentation process stages.

Experiments were performed using 80% training items and 20% testing items, and the same proportions (80% training, 20% testing) with users. Test users and test items were taken at random from all users and items of each dataset. Figure 4.4 shows the cross validation main steps to obtain reliability quality prediction values. We start from the disjoint RS training and testing sets. Testing set provides real ratings $R_{u,i}$ (correct values), whereas training set provides predicted values $P_{u,i}$ (applying some CF method). From correct ratings values and predicted ratings values, we obtain prediction error values $E_{u,i}$ (MAE). From the training set, the predicted reliability values $L_{u,i}$ are also obtained. Reliability quality prediction methods, measures and algorithms combine $R_{u,i}$, $P_{u,i}$, $E_{u,i}$ and $L_{u,i}$ to obtain the required reliability quality prediction values.

KNN prediction experiments were performed using a number of neighbors (K) from 20 to 400, step 20. KNN recommendation experiments were performed using $K = 200$, a relevancy threshold $\theta = 4$ and a number of recommendations (N) from 2 to 20, step 2. The MovieLens 1M dataset was taken from

<https://grouplens.org/datasets/movielens/>.

The Netflix prize dataset was collected between October 1998 and December 2005. The MovieLens 1M facts are: number of users: 4382, numbers of items: 3952, number of ratings: 10,000,209, min and max rating values: 1-5. The Netflix facts are: number of users: 480,189, numbers of items: 17,770, number of ratings: 100,480,507, min and max rating values: 1-5. No preprocessing was carried out on the datasets.

FIGURE 4.4: Cross validation process to obtain reliability quality prediction values.

The experiments section is structured as follows: 1) description of the tested RM, 2) explanation of the published method that we take as baseline in RQM prediction, 3) presentation of quality results in prediction reliability and comparing the proposed method with the baseline, 4) presentation of the quality results in recommendation reliability, and 5) discussion of the most relevant results.

4.4.2 Tested reliability measures

This section describes the operation of each RM that we are going to test. We chose the most significant RM from the Mazurowsky's paper (Mazurowski, 2013), and we added a RM not included in (Mazurowski, 2013). Mazurowsky tests the following RM: support for user, support for item, variability for item, resample, resample fast and inject noise. Resample and inject noise are non-scalable RM: the computational

resources they require are too large, so in (Mazurowski, 2013) resample fast is resented as the most suitable alternative to these two non-scalable RM. Variability for item is a simple RM, which we have replaced with an equivalent specialized RM: knn variability.

Table 4.4 describes and formalizes each of the RM tested in the experiments. Note that RMs have been ordered in Table 4.4 from lowest to highest computational complexity.

TABLE 4.4: Tested reliability measures.

Reliability Measure	Description	Formulation
Support for user	Number of ratings made by the user	$l_{u,i} = \#\{r_{u,j} r_{u,j} \neq \bullet, j \in I\}$, $\bullet$ means not voted
Support for item	Number of ratings received by the item	$l_{u,i} = \#\{r_{s,i} r_{s,i} \neq \bullet, s \in U\}$
KNN variability	Inverse of the variance of the ratings made on the item by the user's K-neighbors.	$l_{u,i} = \frac{\#V_{u,i}}{\sum_{s \in V_{u,i}} r_{s,i} - \bar{v}_{u,i} },$ $V_{u,i} = \{s \in K_u r_{s,i} \neq \bullet\},$ $\bar{v}_{u,i} = \frac{\sum_{s \in V_{u,i}} r_{s,i}}{\#V_{u,i}} \text{ where:}$ $K_u \rightarrow \text{set of neighbors of } u$
Fast resample	RM Resample repeats the entire prediction process of the CF algorithm using different subsets of the entire RS set of ratings. Fast resample uses a fraction of the original set to be resampled, making the process faster. For each of the N resampled matrices we obtain their set of predictions. Reliability is defined as the inverse of the predictions standard deviation.	R = ratings matrix, R^n = random select (R): $ R^n = \alpha R, \alpha \in [0..1], n \in [1..N],$ $P^n = R^n$ predictions $l_{u,i} = \frac{1}{std_{dev}\{P_{u,i}^n : n \in [1..N]\}}$

4.4.3 Mazurovsky's method

This section summarizes the method proposed in (Mazurowski, 2013) to measure the quality of each RM tested. This method will act as baseline of the prediction RQM we present (*RPI*). Mazurovsky's method determines the relationship between the reliability values and their associated prediction errors. The steps of the method are:

1. To determine a discrete set of K reliability intervals (e.g. $k \in 1..10$).
2. To create groups of predictions. Each group k contains the predictions with reliability k .

3. To calculate the accuracy (MAE) of each prediction, obtaining K groups of errors.
4. To eliminate bigger errors (5% of total errors).
5. To measure the amplitude of the error interval (confidence interval) of each of the K groups.
6. To obtain the measure of quality from the variation of the interval sizes (confidence intervals) with respect to the variation of the values of k (of reliability). This variation is expressed by a "confidence curve". Mazurowski (Mazurowski, 2013) generically explains how this stage could be implemented. The specific solution they provide is to subtract the confidence curve values in $k = 1$ and $k = K$.

Figure 4.5 graphically shows the elements involved in the baseline method.

FIGURE 4.5: Baseline method scheme.

4.4.4 Prediction reliability quality results

In this section we show the results of reliability quality on predictions: Figure 4.6 . The two top graphs show the results making use of the Netflix dataset, while the two lower graphs show the results making use of the MovieLens 1M dataset. The graphs on the left show the results of the proposed method and the graphs on the right show the results obtained using the baseline (Mazurowski, 2013).

FIGURE 4.6: Reliability quality results (referring to predictions) obtained when using several RM, reliability quality methods and recommender systems datasets. Axis x: K (number of neighbors), axis y : error (a and c: error improvement; b and d: absolute error). Experiments parameters: cross validation: 20% test users, 20% test items.

Graphs a) and c) from Figure 4.6 show the superiority of RM fast resample and knn variability , followed by support for item in the case of the Netflix dataset. The results are consistent with the degree of complexity of RM: support for user and support for item are very simple and fast RM, but their results are not competitive, especially when applied to high values of K (neighborhood).

Graphs a) and c) from Figure 4.6 show that the results of the proposed *RPI* method are more stable and better explainable than their baseline (Mazurowski, 2013) counterparts in Figure 4.6 b) and d). The baseline method reports a quality hard to justify for the support for item RM. This situation can be produced due to the poor approximation assumed by the evolution of the confidence curve only according to its first and last values (step 6 in the "Mazurowski's method" subsection). The discontinuities shown in graphs b) and d) are also, probably, due to the limitation indicated.

It is important to realize the difference in the scale of results of the proposed QM (*RPI*) and the baseline method: *RPI* reports the improvement (gain factor) that is reached when the correct predictions correspond to high reliabilities. The baseline method does not provide improvements: it returns the difference in the absolute value of the error between the maximum and the minimum reliability values. As an example: when using a value of $K = 200$ in MovieLens 1M and knn variability, *RPI* indicates a 21% improvement, while the baseline method gives us an absolute half-point error gain.

By analyzing the scale of the y-axis on the graphs, we can verify that the improvements obtained are greater when using the MovieLens 1M dataset than when using Netflix: Increasing the size of the dataset, in general, also increases the reliability of its predictions, which reduces the margin of improvement of the RM: (Shani and Gunawardana, 2011) "the confidence in the predicted property also grows with the amount of data".

Results obtained using the proposed method better explain the behavior of the RM than the results obtained using the baseline method: 1) the improvement of knn variability when the K values increase is logical because the greater the number of neighbors the greater the margin to feed the variability tested by the RM, 2) the improvement of fast resample when the values of K increase is also logical, because selecting a greater number of neighbors increases the impact of each resampling in the predictions obtained, and, therefore, increases the variation in the predictions of the different resamplings, and 3) the improvement of support for user and support for item, when increasing K , should not be as marked as the improvements obtained by knn variability, because in these cases the RM does not depend directly on the K value.

In summary: 1) *RPI* produces results that adequately explain the logic behind RM, while the baseline method produces some results that do not fit the expected behavior, 2) *RPI* provides more balanced, continuous and homogeneous results than the baseline method, and 3) *RPI* offers relative values (of quality improvement), whereas the baseline method provides absolute values that cannot be easily compared and depend on the error scales of each dataset, on each collaborative filtering algorithm and on each RM tested.

4.4.5 Recommendation reliability quality results

In this section we show the results of reliability quality in recommendations: Figure 4.7. Because there is no published measure or method that serves as baseline, only the recommendation RQM that we propose (*RRI*) is provided.

Figure 4.7 shows the *RRI* improvements obtained on the tested RM: knn variability, fast resample, support for user and support for item. Both graphs (Netflix

FIGURE 4.7: Reliability quality results (referring to recommendations) obtained when using several reliability measures and public datasets. Axis x: N (number of recommendations), axis y : error improvement (gain factor). Experiment parameters: K = 200, relevant threshold: 4, cross validation: 20% test users, 20% test items.

and MovieLens) present similar trends; in particular the trend to worsen quality is observed when the number of recommendations N increases. This trend is logical, since the most promising recommendations "disappear" as we increase the number of items to recommend; e.g. if we have only 6 reliable and relevant items, then N =

6 is adequate, but $N = 14$ might not be, as the last eight recommendation reliability values will be lower and the risk of recommendation will be greater.

The results in Figure 4.7 present knn variability as the best-performing RM, while fast resample reduces the performance compared to their prediction quality results. The most plausible explanation is that fast resample is a RM based on stability. It is not based on CF parameters, as is the case of knn variability. In this way, the variation in the range of reliability values it provides is small, and it is not as effective to choose predictions with the highest values of reliability (recommendations).

By analyzing the y-axis scales in Figure 4.6 and 4.7, it can be determined that recommendation improvements are much higher than prediction improvements, particularly when the number of recommendations is small. This difference is explained from a quantitative point of view: while prediction improvements are obtained by averaging a huge amount of predictions, the recommendation improvements refer to small amounts of recommendations. Thus, in the case of recommendations, only the relevant predictions that have very high values of reliability are chosen, and, therefore, improvements are higher.

4.4.6 Experiment discussion

This section summarizes the main conclusions drawn from the experiments, paying special attention to those that compare the proposed QM (*RPI*) with the baseline method.

General concepts:

- Each RM provides different quality improvements, where knn variability shows the best performance, especially in the quality of recommendations.
- Fast resample and knn variability are RM suitable for measuring the quality of predictions.
- In general, the larger a RS dataset, the lower the improvement errors that will reach the QM.
- The recommendation error improvements are greater than the prediction improvements.
- The lower the number of recommendations, the greater the improvement errors given.

Comparison between the RQM proposed and the baseline method:

- QM-*RPI* provides results that fit much better with those expected than the baseline method.

- *RPI* offers balanced and homogeneous results, while the baseline method generates variable and unstable results.
- *RPI* returns relative improvement results, whereas the baseline method returns absolute errors: *RPI* is more suitable to make comparisons and to establish analogies between QM, RM, datasets, and collaborative filtering methods.
- *RPI* is a QM defined by a single equation, whereas the baseline method is an algorithm: *RPI* is simpler, easier to understand, intuitive and universal.
- *RPI* lacks arbitrary parameters, whereas the baseline method requires some parameterized heuristics to interpret the confidence curve.
- As far as we know, no recommendation RQM approach has been published. We provide the recommendation QM-*RRI*.

4.5 Summary

This chapter proposes two RQMs named *RPI* and *RRI*. Users want to know the reliability of the recommendations; they look for the number of votes or comments on the films, hotels, products, etc. and they do not accept high predictions (usually 4 or 5 stars in e-commerce) if there is no reliability evidence.

Research into recommender system QM has focused on accuracy. Moreover, novelty, serendipity and diversity have been studied; nevertheless there is an important lack of research into reliability/confidence QM. It is important to promote RM associated with the predictions; the foundation for this research is to establish a set of appropriate RQM.

This chapter proposes a reliability quality prediction measure (*RPI*) and a reliability quality recommendation measure (*RRI*). Both QMs are based on the hypothesis that the more suitable a RM, the better accuracy results it will provide when applied. These RQM show accuracy improvements when appropriate reliability values are associated with their predictions (i.e. high reliability values associated with correct predictions and low reliability values associated with incorrect predictions).

The results of the experiments performed indicate the superiority of the RM knn variability and fast resampling. They also show us that larger datasets have fewer opportunities for reliability improvements. In addition, the lower the number of recommendations we make, the better recommendation quality results we obtain.

The prediction RQM (*RPI*) proposed offers better characteristics than the existing baseline methods: 1) its results are in line with those expected, and they are also stable and progressive, 2) *RPI* results indicate improvements (gain factors), suitable for making comparisons between different RM and various datasets, 3) it is defined by

an equation: it is simpler than a method or algorithm and does not contain arbitrary parameters.

This chapter also provides a recommendation RQM: *RRI*. No recommendation RQM has been published in the RS field. Its experimental results show that the improvements are much higher in recommendations than in predictions, especially when the number of recommendations is low. This circumstance indicates the importance of: 1) making use of reliability values to calculate RS recommendations, and 2) more closely examining the development of new RM to be applied to recommendation tasks.

Chapter 5

Conclusions and future works

As the final chapter of this thesis, Chapter 5 concludes our research efforts towards CF RS introduced in previous chapters. We first highlight our contributions: 1) A novel reliability measure is proposed for matrix factorization based collaborative filtering recommender systems; 2) Reliability quality measures for both prediction and recommendation are also proposed for any type of collaborative filtering recommender systems. Then we plan future works based on some new trends in CF RS.

Seen as a proof of the growing prosperity of artificial intelligence, recommender systems (RS) have been recognized as one of its most successful branches. More and more recommendation scenarios are established thanks to the emerging novel information sources. This thesis focuses on the most popular family of RS, known as collaborative filtering (CF) RS, and it offers some insights on the development of such areas. The performance of current RS is evaluated not only by the conventional recommendation accuracy, but also some novel properties. This thesis focuses on one of the most important RS properties, named reliability. After a comprehensive study of state-of-the-art publications, we observe a lack of: 1) Proper reliability measures for model based CF RS; 2) Universal reliability quality measures which can provide quantitative values. As a consequence, the main objective of this thesis is to provide solutions to such lacks.

This thesis makes two main innovative contributions to the state of the art. First we propose a novel reliability measure (RM) based on the fact that reliability and accuracy are highly correlated. Matrix factorization techniques are applied to obtain hierarchical abstraction levels. No heuristic inductions or predefined thresholds are involved in the proposed methods. Our generated numerical reliability values are independent of side information sources, which makes it universally applicable to all types of CF RS. The experiment results confirm that the proposed method is valid from three aspects: a) The reliability values assigned to users and item predictions follow suitable distributions b) The obtained set of reliable predictions improve the quality of the RS accuracy (MAE), c) Reliable predictions improve the RS recommendation quality (precision). Although the proposed method does not require the incorporation of side information sources, such additional information is always beneficial for the improvement of the quality of results. Our method also shows a positive potential against shilling attacks. Results obtained from the proposed RM validate the first half of our hypothesis.

We also propose the first two universal reliability quality measures (RQMs): prediction RQM and recommendation RQM. They are able to numerically measure the reliability of different RMs. To explain these concepts, we first clarify the difference between RMs and RQMs, and then we introduce the historical evolution on the reliability quality field. The designed RQMs follow the following principle: “the higher the prediction reliability value, the higher its accuracy error penalization; the lower the prediction error, the bigger its reliability value reward”. The obtained experiment results show, on several public real datasets, our RQMs have advantages over two representative baselines: the RM knn variability and the fast resampling. This supports the second half of our hypothesis.

Reliability values could be used on a large quantity of future works, such as: Filtering non-reliable predictions on cold-start situations, improving clustering of predictions, extending the proposed method to semi-structured data (news, HTML, XML,

etc.), explanation of recommendations, serendipity based on low reliability predictions, visual representation of RS, navigational tools through related items, extending the model covering recommendations to groups of users, merging sensor reliabilities with prediction reliabilities on the Internet of things RS, inferring reliable users and reliable items from reliable predictions, weighting imputation of predictions to the ratings matrix, and expanding the set of results offered by CF deep neural networks.

More promising future works will lead to the design of brand new recommender system RMs. These measures could be applied to different matrix factorization techniques and to content-based, context-aware and social recommendation approaches. The designed recommender system RMs can be tested, compared and improved using the proposed RQMs.

Appendix A

List of author's publications

This thesis summarizes years of part of our research works in recommender systems. As our academic contributions to the recommender system community, five papers have been published in top-level journals indexed by both Journal Citation Report (JCR) and Science Citation Index (SCI), and one more is under the second round review.

1. **Zhu, B.**, Ortega, F., Bobadilla, J. and Gutierrez, A. (2018). Assigning reliability values to recommendations using matrix factorization, *Journal of Computational Science* (JCR/SCI Q2, impact factor 1.906), 26, 165-177.
2. Bobadilla, J., Gutierrez, A., Ortega, F. and **Zhu, B.** (2018). Reliability Quality Measures For Recommender Systems, *Information Sciences* (JCR/SCI Q1, impact factor 4.378) 442-443, 145-157.
3. Ortega, F., **Zhu, B.**, Bobadilla, J. and Hernando, A. (2018). CF4J: Collaborative Filtering for Java, *Knowledge-Based Systems* (JCR/SCI Q1, impact factor 4.514), 152, 94-99.
4. **Zhu, B.**, Hurtado, R., Bobadilla, J., and Ortega, F. (2018). An efficient recommender system method based on the numerical relevances and the non-numerical structures of the ratings. *IEEE Access* (JCR Q1, SCI Q2, impact factor 3.557). Accepted and in production. Early access: DOI: 10.1109/ACCESS.2018.2868464.
5. Gil, S., Bobadilla, J., Ortega, F. and **Zhu, B.** (2018). VisualRS: Java framework for visualization of recommender systems information, *Knowledge-Based Systems* (JCR/SCI Q1, impact factor 4.514), 155, 66-70.
6. **Zhu, B.**, Valdiviezo, P., Bobadilla, J., and Ortega, F. (2018, under the second round review). Collaborative Filtering Similarity Measure based on Vectorial Reduction, Structural Information and Numerical Comparison. *IEEE Access* (JCR Q1, SCI Q2, impact factor 3.557).

Bibliography

- Adamopoulos, Panagiotis and Alexander Tuzhilin (2015). "On unexpectedness in recommender systems: Or how to better expect the unexpected". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 5.4, p. 54.
- Adomavicius, G., B. Mobasher, F. Ricci, and A. Tuzhilin (2011). "Context-Aware Recommender Systems". In: *AI Magazine*, pp. 67–80.
- Adomavicius, G., J.C. Bockstedt, S.P. Curley, and J.J. Zhang (Apr. 2013). "Do Recommender Systems Manipulate Consumer Preferences? A Study of Anchoring Effects". In: *Information Systems Research* 24, pp. 956–975.
- Aggarwal, C. C. (2016a). "Neighborhood-based collaborative filtering". In: *Recommender Systems*. Springer, pp. 29–70.
- Aggarwal, C.C. (Mar. 2016b). "An Introduction to Recommender Systems". In: *Recommender Systems*. Springer, Cham. Chap. 1, pp. 1–28.
- Ahn, H. J. (Jan. 2008). "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem". In: *Information Sciences* 178, pp. 37–51.
- Ahn, J-W., P. Brusilovsky, J. Grady, D. He, and S. Y. Syn (May 2007). "Open user profiles for adaptive news systems: help or harm?" In: *Proceedings of the 16th international conference on World Wide Web* (Banff, Alberta, Canada, May 8–12, 2007). WWW '07, pp. 11–20.
- Aiello, L. M. and N. Barbieri (2017). "Evolution of Ego-networks in Social Media with Link Recommendations". In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. WSDM '17. Cambridge, United Kingdom: ACM, pp. 111–120.
- Aïmeur, Esma, Gilles Brassard, José M Fernandez, and Flavien Serge Mani Onana (2008). "A lambic: a privacy-preserving recommender system for electronic commerce". In: *International Journal of Information Security* 7.5, pp. 307–334.
- Al-Shamri, M. Y. H. (Jan. 2016). "User profiling approaches for demographic recommender systems". In: *Knowledge-Based Systems* 100, pp. 175–187.
- Antikacioglu, Arda and R. Ravi (2017). "Post Processing Recommender Systems for Diversity". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '17. Halifax, NS, Canada, pp. 707–716. ISBN: 978-1-4503-4887-4.
- Ar, Yilmaz and Erkan Bostanci (2016). "A genetic algorithm solution to the collaborative filtering problem". In: *Expert Systems with Applications* 61, pp. 122–128.

- ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2016.05.021>. URL: <http://www.sciencedirect.com/science/article/pii/S0957417416302469>.
- Backstrom, L. and J. Leskovec (2011). "Supervised random walks: predicting and recommending links in social networks". In: *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, pp. 635–644.
- Badsha, Shahriar, Xun Yi, and Ibrahim Khalil (2016). "A practical privacy-preserving recommender system". In: *Data Science and Engineering* 1.3, pp. 161–177.
- Balabanović, M. and Y. Shoham (Mar. 1997). "content-based, collaborative recommendation". In: *Communications of the ACM* 40, pp. 66–72.
- Bao, J., Y. Zheng, D. Wilkie, and M. Mokbel (2015). "Recommendations in location-based social networks: a survey". In: *GeoInformatica* 19.3, pp. 525–565.
- Barkan, O. and N. Koenigstein (Sept. 2016). "ITEM2VEC: Neural item embedding for collaborative filtering". In: *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6.
- Bedi, Punam and Ravish Sharma (2012). "Trust based recommender system using ant colony for trust computation". In: *Expert Systems with Applications* 39.1, pp. 1183–1190. ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2011.07.124>. URL: <http://www.sciencedirect.com/science/article/pii/S0957417411010864>.
- Bellogín, Alejandro, Iván Cantador, and Pablo Castells (2010). "A study of heterogeneity in recommendations for a social music service". In: *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*. ACM, pp. 1–8.
- Bellogín, Alejandro, Iván Cantador, Fernando Díez, Pablo Castells, and Enrique Chavarriaga (2013). "An empirical comparison of social, collaborative filtering, and hybrid recommenders". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 4.1, p. 14.
- Bellogín, P. and I. Castells (Aug. 2014). "Neighbor selection and weighting in user-based collaborative filtering: a performance prediction approach". In: *ACM Transactions on the Web (TWEB)* 2.
- Billsus, D. and M. J. Pazzani (June 2000). "User Modeling for Adaptive News Access". In: *User Modeling and User-Adapted Interaction* 10, pp. 147–180. URL: <https://doi.org/10.1023/A:1026501525781>.
- Blei, D., A. Ng, and M. Jordan (2003). "Latent dirichlet allocation". In: *Journal of machine Learning research* 3, Jan, pp. 993–1022.
- Bobadilla, J., F. Ortega, and A. Hernando (Mar. 2012). "A collaborative filtering similarity measure based on singularities". In: *Information Processing and Management* 48, pp. 204–217.
- Bobadilla, J., F. Serradilla, and J. Bernal (2010). "A new collaborative filtering metric that improves the behavior of recommender systems". In: *Knowledge Based System* 23, pp. 520–528.

- Bobadilla, J., F. Ortega, A. Hernando, and J. Alcalá (Aug. 2011). "Improving collaborative filtering recommender system results and performance using genetic algorithms". In: *Knowledge-Based Systems* 24, pp. 1310–1316.
- Bobadilla, J., A. Hernando, F. Ortega, and A. Gutiérrez (Jan. 2012). "Collaborative filtering based on significances". In: *Information Sciences* 185, pp. 1–17.
- Bobadilla, J., F. Ortega, A. Hernando, and A. Arroyo (2013a). "A balanced memory-based collaborative filtering similarity measure". In: *International Journal of Intelligent Systems* 27, pp. 939–946.
- Bobadilla, J., F. Ortega, A. Hernando, and A. Gutiérrez (2013b). "Recommender systems survey". In: *Knowledge Based System*, pp. 109–132.
- Bobadilla, J., F. Ortega, A. Hernando, and A. Gutiérrez (2017). "Recommender systems clustering using Bayesian non negative matrix factorization". In: *IEEE Access* 6, pp. 3549–3564.
- Bobadilla, J., A. Gutiérrez, F. Ortega, and B. Zhu (Jan. 2018). "Reliability quality measures for recommender systems". In: *Information Sciences* 442–443, pp. 145–157.
- Bogers, T. (2018). "Tag-based recommendation". In: *Social Information Access*. Springer, pp. 441–479.
- Bothorel, Cécile, Neal Lathia, Romain Picot-Clemente, and Anastasios Noulas (2018). "Location recommendation with social media data". In: *Social Information Access*. Springer, pp. 624–653.
- Breese, John S, David Heckerman, and Carl Kadie (1998). "Empirical analysis of predictive algorithms for collaborative filtering". In: *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pp. 43–52.
- Cai, D., X. He, J. Han, and T. S. Huang (2011). "Graph regularized nonnegative matrix factorization for data representation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.8, pp. 1548–1560.
- Cai, G., K. Lee, and I. Lee (2018). "Itinerary recommender system with semantic trajectory pattern mining from geo-tagged photos". In: *Expert Systems with Applications* 94, pp. 32–40. ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2017.10.049>. URL: <http://www.sciencedirect.com/science/article/pii/S0957417417307315>.
- Campos, Pedro G, Fernando Díez, and Iván Cantador (2014). "Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols". In: *User Modeling and User-Adapted Interaction* 24.1-2, pp. 67–119.
- Castells, Pablo, Neil J Hurley, and Saul Vargas (2015). "Novelty and diversity in recommender systems". In: *Recommender Systems Handbook*. Springer, pp. 881–918.
- Castells, Pablo, Saúl Vargas, and Jun Wang (2011). "Novelty and diversity metrics for recommender systems: choice, discovery and relevance". In:
- Chang, T.M. and W.F. Hsiao (2011). "Model-based Collaborative Filtering to Handle Data Reliability and Ordinal Data Scale". In: *FSKD*.

- Chen, D., Y. Ying, and S. Gong (Oct. 2008). "A Collaborative Filtering Algorithm Based on Rough Set and Fuzzy Clustering". In: *2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery*. Vol. 1, pp. 17–20.
- Chen, G., F. Zhu, and Ph. A. Heng (2018). "Large-Scale Bayesian Probabilistic Matrix Factorization with Memo-Free Distributed Variational Inference". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12.3, p. 31.
- Chen, Li, Guanliang Chen, and Feng Wang (2015). "Recommender systems based on user reviews: the state of the art". In: *User Modeling and User-Adapted Interaction* 25.2, pp. 99–154.
- Cheng, Z. and J. Shen (2016). "On effective location-aware music recommendation". In: *ACM Transactions on Information Systems (TOIS)* 34.2, p. 13.
- Cheng, Zhiyong and Jialie Shen (2014). "Just-for-me: an adaptive personalization system for location-aware social music recommendation". In: *Proceedings of international conference on multimedia retrieval*. ACM, p. 185.
- Chirita, Paul-Alexandru, Wolfgang Nejdl, and Cristian Zamfir (2005). "Preventing shilling attacks in online recommender systems". In: *Proceedings of the 7th annual ACM international workshop on Web information and data management*. ACM, pp. 67–74.
- Ciliberto, C., D. Stamos, and M. Pontil (2017). "Reexamining low rank matrix factorization for trace norm regularization". In: *arXiv preprint arXiv:1706.08934*.
- Covington, P., J. Adams, and E. Sargin (2016). "Deep neural networks for youtube recommendations". In: *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, pp. 191–198.
- Dai, J., B. Yang, C. Guo, and Z. Ding (Apr. 2015). "Personalized route recommendation using big trajectory data". In: *2015 IEEE 31st International Conference on Data Engineering*, pp. 543–554. DOI: [10.1109/ICDE.2015.7113313](https://doi.org/10.1109/ICDE.2015.7113313).
- Desrosiers, C. and G. Karypis (2011). "A comprehensive survey of neighborhood-based recommendation methods". In: *Recommender systems handbook*. Springer, pp. 107–144.
- Ding, C., T. Li, and W. Peng (2006). "Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method". In: *AAAI*. Vol. 42, pp. 137–143.
- Ding, Yi and Xue Li (2005). "Time weight collaborative filtering". In: *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, pp. 485–492.
- Eirinaki, M., M. Vazirgiannis, and I. Varlamis (Aug. 2003). "SEWeP: using site semantics and a taxonomy to enhance the Web personalization process". In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '03, pp. 99–108.
- Ekstrand, M. D., M. Tian, I. M. Azpiazu, J. D. Ekstrand, O. Anuyah, D. McNeill, and M. S. Pera (2018). "All The Cool Kids, How Do They Fit In?: Popularity and Demographic Biases in Recommender Evaluation and Effectiveness". In: *Proceedings*

- of the 1st Conference on Fairness, Accountability and Transparency. Ed. by Sorelle A. Friedler and Christo Wilson. Vol. 81. Proceedings of Machine Learning Research. New York, NY, USA: PMLR, pp. 172–186.
- Elsevier (2018). *Can I include/use my article in my thesis/dissertation?* URL: <https://www.elsevier.com/about/policies/copyright/permissions> (visited on 09/16/2018).
- Fakhfakh, R., A. Ammar, and C. Amar (2017). “Deep Learning-Based Recommendation: Current Issues and Challenges”. In: *International Journal of Advanced Computer Science and Applications (IJACSA)* 8, p. 12.
- Feng, Shanshan, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan (2015). “Personalized Ranking Metric Embedding for Next New POI Recommendation.” In: *IJCAI*. Vol. 15, pp. 2069–2075.
- Freudenthaler, C., L. Schmidt-thieme, and S. Rendle (2009). “Factorization Machines Factorized Polynomial Regression Models”. In:
- G. Adomavicius, A. Tuzhilin (June 2005). “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions”. In: *IEEE Transactions on Knowledge and Data Engineering* 17, pp. 734–749.
- Gabrilovich, E. and S. Markovitch (Jan. 2007). “Computing semantic relatedness using Wikipedia-based explicit semantic analysis”. In: *Proceedings of the 20th international joint conference on Artificial intelligence* (Hyderabad, India, Jan. 6–12, 2007). *IJCAI’07*, pp. 1606–1611.
- Gao, Min, Bin Ling, Linda Yang, Junhao Wen, Qingyu Xiong, and Shun Li (2018). “From similarity perspective: a robust collaborative filtering approach for service recommendations”. In: *Frontiers of Computer Science*, pp. 1–16.
- Garcia, R. and X. Amatriain (2010). “Weighted content based methods for recommending connections in online social networks”. In: *Workshop on Recommender Systems and the Social Web*. Association for Computing Machinery, pp. 68–71.
- Ge, M., C. D-Battenfeld, and D. Jannach (Sept. 2010). “Beyond accuracy: evaluating recommender systems by coverage and serendipity”. In: *Proceedings of the fourth ACM conference on Recommender systems* (Barcelona, Spain, Sept. 26–30, 2010). *RecSys ’10*, pp. 257–260.
- Gemmis, M. de, P. Lops, G. Semeraro, and P. Basile (2008). “Integrating Tags in a Semantic Content-based Recommender”. In: *Proceedings of the 2008 ACM Conference on Recommender Systems*. *RecSys ’08*. Lausanne, Switzerland, pp. 163–170.
- Ghazanfar, Mustansar Ali, Adam Prügel-Bennett, and Sandor Szedmak (2012). “Kernel-Mapping Recommender system algorithms”. In: *Information Sciences* 208, pp. 81 – 104. ISSN: 0020-0255. DOI: <https://doi.org/10.1016/j.ins.2012.04.012>. URL: <http://www.sciencedirect.com/science/article/pii/S0020025512002587>.
- Ghazinour, Kambiz, Stan Matwin, and Marina Sokolova (2016). “YOURPRIVACYPROTECTOR, A recommender system for privacy settings in social networks”. In: *arXiv preprint arXiv:1602.01937*.

- Godoy, D. and A. Corbellini (2016). "Folksonomy-based recommender systems: a state-of-the-art review". In: *International Journal of Intelligent Systems* 31, pp. 314–346.
- Goldberg, David, David Nichols, Brian M. Oki, and Douglas Terry (Dec. 1992). "Using Collaborative Filtering to Weave an Information Tapestry". In: *Commun. ACM* 35.12, pp. 61–70. ISSN: 0001-0782. DOI: [10.1145/138859.138867](https://doi.acm.org/10.1145/138859.138867). URL: <http://doi.acm.org/10.1145/138859.138867>.
- GomezUribe, C. and N. Hunt (2016). "The netflix recommender system: algorithms, business value, and innovation". In: *ACM Transactions on Management Information Systems (TMIS)* 6.
- Grbovic, M., V. Radosavljevic, N. Djuric, N. Bhamidipati, J. Savla, V. Bhagwan, and D. Sharp (2015). "E-commerce in Your Inbox: Product Recommendations at Scale". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '15. Sydney, NSW, Australia, pp. 1809–1818.
- Guo, G., J. Zhang, and N. Y-S (2015). "TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings." In: *AAAI*. Vol. 15, pp. 123–125.
- Guo, G., J. Zhang, and N. Yorke-Smith (Jan. 2015). "Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems". In: *Knowledge-Based Systems* 74, pp. 14–27.
- Guo, X., B. Pan, D. Cai, and X. He (2017). "Robust Asymmetric Bayesian Adaptive Matrix Factorization". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 1760–1766.
- Gupta, P., A. Goel, J. Lin, A. Sharma, D. Wang, and R. Zadeh (2013). "WTF: The Who to Follow Service at Twitter". In: *Proceedings of the 22Nd International Conference on World Wide Web*. WWW '13. Rio de Janeiro, Brazil: ACM, pp. 505–514.
- Hastie, T., R. Mazumder, J. D. Lee, and R. Zadeh (2015). "Matrix completion and low-rank SVD via fast alternating least squares". In: *The Journal of Machine Learning Research* 16.1, pp. 3367–3402.
- He, Jing, Xin Li, Lejian Liao, Dandan Song, and William K Cheung (2016a). "Inferring a Personalized Next Point-of-Interest Recommendation Model with Latent Behavior Patterns." In: *AAAI*, pp. 137–143.
- He, X., H. Zhang, M.Y. Kan, and T.S. Chua (July 2016b). "Fast Matrix Factorization for Online Recommendation with Implicit Feedback". In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (Pisa, Italy, July 17–21, 2016). SIGIR'16, pp. 549–558.
- Herlocker, J.L., J.A. Konstan, and J Riedl (Dec. 2000). "Explaining collaborative filtering recommendations". In: *Proceedings of the 2000 ACM conference on Computer supported cooperative work* (Philadelphia, Pennsylvania, USA, Dec. 2–6, 2000). CSCW '00, pp. 241–250.

- Herlocker, J.L., J.A. Konstan, A. Borchers, and J. Riedl (Aug. 1999). "An algorithmic framework for performing collaborative filtering". In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (Berkeley, California, USA, Aug. 15–19, 1999). SIGIR '99, pp. 230–237.
- Herlocker, J.L., J.A. Konstan, L.G. Terveen, and J.T. Riedl (Jan. 2004). "Evaluating collaborative filtering recommender systems". In: *ACM Transactions on Information Systems (TOIS)* 22, pp. 5–53.
- Hernando, A., J. Bobadilla, and F. Ortega (Apr. 2016). "A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model". In: *Knowledge-Based Systems* 97, pp. 188–202.
- Hernando, A., R. Moya, F. Ortega, and J. Bobadilla (Oct. 2013a). "Hierarchical graph maps for visualization of collaborative recommender systems". In: *Journal of Information Science* 40, 97–106.
- Hernando, A., J. Bobadilla, F. Ortega, and J. Tejedor (Jan. 2013b). "Incorporating reliability measurements into the predictions of a recommender system". In: *Information Sciences* 218, pp. 1–16.
- Hernando, A., J. Bobadilla, F. Ortega, and A. Gutiérrez (Aug. 2013c). "Trees for explaining recommendations made through collaborative filtering". In: *Information Sciences* 239, pp. 1–17.
- Hofmann, T. (1999). "Probabilistic latent semantic analysis". In: *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pp. 289–296.
- Horvath, T. and A. C. P. L. F. de Carvalho (Sept. 2017). "Evolutionary computing in recommender systems: a review of recent research". In: *Natural Computing* 16, 441–462.
- Hung, Chih-Chieh, Chih-Wen Chang, and Wen-Chih Peng (2009). "Mining trajectory profiles for discovering user communities". In: *Proceedings of the 2009 International Workshop on Location Based Social Networks*. ACM, pp. 1–8.
- Hurley, N. and M. Zhang (Mar. 2011). "Novelty and Diversity in Top-N Recommendation – Analysis and Evaluation". In: *ACM Transactions on Internet Technology (TOIT)* 10, Article No. 14.
- Hurley, Neil J. (2011). "Robustness of Recommender Systems". In: *Proceedings of the Fifth ACM Conference on Recommender Systems*. RecSys '11. Chicago, USA, pp. 9–10. ISBN: 978-1-4503-0683-6.
- Jannach, Dietmar and Gediminas Adomavicius (2016). "Recommendations with a Purpose". In: *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, pp. 7–10.
- Järvelin, Kalervo and Jaana Kekäläinen (2002). "Cumulated gain-based evaluation of IR techniques". In: *ACM Transactions on Information Systems (TOIS)* 20.4, pp. 422–446.

- Ji, Y., W. Hong, Y. Shangguan, H. Wang, and J. Ma (2016). "Regularized singular value decomposition in news recommendation system". In: *Computer Science & Education (ICCSE), 2016 11th International Conference on*. IEEE, pp. 621–626.
- Juan, Y., Y. Zhuang, W-S. Chin, and C-J. Lin (2016). "Field-aware factorization machines for CTR prediction". In: *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, pp. 43–50.
- Kaminskas, Marius and Derek Bridge (2017). "Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems". In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 7.1, p. 2.
- Kant, V. and K. Bharadwaj (2013). "Integrating Collaborative and Reclusive Methods for Effective Recommendations: A Fuzzy Bayesian Approach". In: *International Journal of Intelligent Systems* 28, pp. 1099–1123.
- Kazai, Gabriella, Iskander Yusof, and Daoud Clarke (2016). "Personalised news and blog recommendations based on user location, Facebook and twitter user profiling". In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, pp. 1129–1132.
- Kim, Jong Woo, Byung Hun Lee, Michael J Shaw, Hsin-Lu Chang, and Matthew Nelson (Feb. 2001). "Application of decision-tree induction techniques to personalized advertisements on internet storefronts". In: *International Journal of Electronic Commerce* 5.3, pp. 45–62.
- Kluver, Daniel, Michael D Ekstrand, and Joseph A Konstan (2018). "Rating-based collaborative filtering: algorithms and evaluation". In: *Social Information Access*. Springer, pp. 344–390.
- Koren, Y. (2008). "Factorization meets the neighborhood: a multifaceted collaborative filtering model". In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 426–434.
- Koren, Y. and R. Bell (2015). "Advances in collaborative filtering". In: *Recommender systems handbook*. Springer, pp. 77–118.
- Koren, Y., R. Bell, and C. Volinsky (2009). "Matrix factorization techniques for recommender systems". In: *Computer* 8, pp. 30–37.
- Koren, Y. and J. Sill (Oct. 2011). "OrdRec: an ordinal model for predicting personalized item rating distributions". In: *Proceedings of the fifth ACM conference on Recommender systems* (Chicago, Illinois, USA, Oct. 23–27, 2011). RecSys '11, pp. 117–124.
- Koren, Yehuda (2009). "Collaborative filtering with temporal dynamics". In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 447–456.
- Kotkov, D., S. Wang, and J. Veijalainen (Nov. 2016). "A survey of serendipity in recommender systems". In: *Knowledge-Based Systems* 111, pp. 180–192.
- Kumar, B. D., S. Girase, and D. Mukhopadhyay (2015). "Role of matrix factorization model in collaborative filtering algorithm: A survey". In: *CoRR*, [abs/1503.07475](https://arxiv.org/abs/1503.07475).

- Kumar, V., A. K. Pujari, S. K. Sahu, V. R. Kagita, and V. Padmanabhan (2017). "Collaborative filtering using multiple binary maximum margin matrix factorizations". In: *Information Sciences* 380, pp. 1–11.
- Kunaver, Matevž and Tomaž Požrl (2017). "Diversity in recommender systems—A survey". In: *Knowledge-Based Systems* 123, pp. 154–162.
- Kwak, H., C. Lee, H. Park, and S. Moon (2010). "What is Twitter, a social network or a news media?" In: *Proceedings of the 19th international conference on World wide web*. AcM, pp. 591–600.
- Le, Q. and T. Mikolov (2014). "Distributed Representations of Sentences and Documents". In: *Proceedings of the 31st International Conference on International Conference on Machine Learning*. ICML'14. Beijing, China, pp. II–1188–II–1196.
- Lee, D. and H. Seung (2001). "Algorithms for non-negative matrix factorization". In: *Advances in neural information processing systems*, pp. 556–562.
- Lee, D. D. and H. S. Seung (1999). "Learning the parts of objects by non-negative matrix factorization". In: *Nature* 401.6755, p. 788.
- Lei, Xiaojiang, Xueming Qian, and Guoshuai Zhao (2016). "Rating prediction based on social sentiment from textual reviews". In: *IEEE Transactions on Multimedia* 18.9, pp. 1910–1921.
- Lerche, L., D. Jannach, and M. Ludewig (July 2016). "On the Value of Reminders within E-Commerce Recommendations". In: *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization* (Halifax, Nova Scotia, Canada, July 13–17, 2016). UMAP '16, pp. 27–35.
- Leung, C., S. Chan, and F. Chung (Oct. 2006). "A collaborative filtering framework based on fuzzy association rules and multiple-level similarity". In: *Knowledge and Information Systems* 10, pp. 357–381.
- Levandoski, Justin J., Mohamed Sarwat, Ahmed Eldawy, and Mohamed F. Mokbel (2012). "LARS: A Location-Aware Recommender System". In: *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*. ICDE '12. Washington, DC, USA: IEEE Computer Society, pp. 450–461. ISBN: 978-0-7695-4747-3. DOI: [10.1109/ICDE.2012.54](https://doi.org/10.1109/ICDE.2012.54). URL: <http://dx.doi.org/10.1109/ICDE.2012.54>.
- Levinas, C. A. (Sept. 2014). "An Analysis of Memory Based Collaborative Filtering Recommender Systems with Improvement Proposals". MA thesis. UPC-URV-UB. URL: <https://upcommons.upc.edu/handle/2099.1/22602>.
- Li, Dongsheng, Qin Lv, Huanhuan Xia, Li Shang, Tun Lu, and Ning Gu (2011). "Pistis: a privacy-preserving content recommender system for online social communities". In: *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*. Vol. 1. IEEE, pp. 79–86.
- Li, Quannan, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma (2008). "Mining user similarity based on location history". In: *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*. ACM, p. 34.

- Ling, G., M. R. Lyu, and I. King (2014). "Ratings meet reviews, a combined approach to recommend". In: *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, pp. 105–112.
- Liu, H., Z. Hu, A. Mian, H. Tian, and X. Zhu (Apr. 2014). "A new user similarity model to improve the accuracy of collaborative filtering". In: *Knowledge-Based Systems* 56, pp. 156–166.
- Liu, Siyuan and Shuhui Wang (2017). "Trajectory community discovery and recommendation by multi-source diffusion modeling". In: *IEEE Transactions on Knowledge and Data Engineering* 29.4, pp. 898–911.
- Liu, Xiao, An Liu, Xiangliang Zhang, Zhixu Li, Guanfeng Liu, Lei Zhao, and Xiaofang Zhou (2017). "When differential privacy meets randomized perturbation: A hybrid approach for privacy-preserving recommender system". In: *International Conference on Database Systems for Advanced Applications*. Springer, pp. 576–591.
- Lops, P., M. Gemmis, and G. Semeraro (Mar. 2011). "Content-based Recommender Systems: State of the Art and Trends". In: *Recommender Systems Handbook*. Ed. by F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. Springer Science+Business Media. Chap. 3, pp. 73–105.
- Lorenzi, Fabiana, Daniela Scherer dos Santos, and Ana LC Bazzan (2005). "Case-based recommender system inspired by social insects". In: *ResearchGate*.
- Luo, X., M. Zhou, Y. Xia, and Q. Zhu (2014). "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems". In: *IEEE Transactions on Industrial Informatics* 10.2, pp. 1273–1284.
- Magnini, B. and C. Strapparva (Oct. 2000). "Experiments in Word Domain Disambiguation for Parallel Texts". In: *ACL-2000 Workshop on Word Senses and Multilinguality*. Hong Kong, China: Association for Computational Linguistics, pp. 27–33.
- Mahdi, M., P. Moradi, A. Abdollahpouri, and M. Jalili (Jan. 2017). "A trust-aware recommendation method based on Pareto dominance and confidence concepts". In: *Knowledge-Based Systems* 116, pp. 130–143.
- Mak, H., I. Koprinska, and J. Poon (Oct. 2003). "INTIMATE: a Web-based movie recommender using text categorization". In: *Proceedings IEEE/WIC International Conference on Web Intelligence*. WI '03, pp. 602–605. DOI: [10.1109/WI.2003.1241277](https://doi.org/10.1109/WI.2003.1241277).
- Martinez, C., C. Porcel, J. Bernabe, and H. Viedma (Aug. 2015). "A model to represent users trust in recommender systems using ontologies and fuzzy linguistic modeling". In: *Information Sciences* 311, pp. 102–118.
- Massa, P. and P. Avesani (Oct. 2007). "Trust-aware recommender systems". In: *Proceedings of the 2007 ACM conference on Recommender systems* (Minneapolis, MN, USA, Oct. 19–20, 2007). RecSys '07, pp. 17–24.
- Matuszyk, P. and M. Spiliopoulou (June 2017). "Stream-based semi-supervised learning for recommender systems". In: *Machine Learning* 106, pp. 771–798.

- Mazumder, R., T. Hastie, and R. Tibshirani (2010). "Spectral regularization algorithms for learning large incomplete matrices". In: *Journal of machine learning research* 11, pp. 2287–2322.
- Mazurowski, M.A. (Aug. 2013). "Estimating confidence of individual rating predictions in collaborative filtering recommender systems". In: *Expert Systems with Applications* 40, pp. 3847–3857.
- McFee, B., L. Barrington, and G. Lanckriet (Oct. 2012). "Learning Content Similarity for Music Recommendation". In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.8, pp. 2207–2218.
- McLaughlin, R.M. and J.L. Herlocker (July 2004). "A collaborative filtering algorithm and evaluation metric that accurately model the user experience". In: *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* (Sheffield, United Kingdom, July 25–29, 2004). SIGIR '04, pp. 329–336.
- McNee, Sean M, Shyong K Lam, Catherine Guetzlaff, Joseph A Konstan, and John Riedl (2003). "Confidence displays and training in recommender systems". In: *Proc. INTERACT*. Vol. 3, pp. 176–183.
- Menon, A. K., D. Chen, L. Xie, and C. Ong (2017). "Revisiting revisits in trajectory recommendation". In: *Proceedings of the International Workshop on Recommender Systems for Citizens*. ACM, p. 2.
- Middleton, S. E., N. R. Shadbolt, and D. C. De Roure (Jan. 2004). "Ontological User Profiling in Recommender Systems". In: *ACM Transactions on Information Systems* 22, pp. 54–88.
- Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems* 26. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger. Curran Associates, Inc., pp. 3111–3119.
- Mllib (2018). *Spark: Collaborative Filtering-RDD-based API*. URL: <https://spark.apache.org/docs/2.2.0/mllib-collaborative-filtering.html> (visited on 07/25/2018).
- Mnih, A. and R. R. Salakhutdinov (2008). "Probabilistic matrix factorization". In: *Advances in neural information processing systems*, pp. 1257–1264.
- Mo, Yijun, Bixi Li, Bang Wang, Laurence T Yang, and Minghua Xu (2018). "Event recommendation in social networks based on reverse random walk and participant scale control". In: *Future Generation Computer Systems* 79, pp. 383–395.
- Mooney, R. J. and L. Roy (2000). "Content-based Book Recommending Using Learning for Text Categorization". In: *Proceedings of the Fifth ACM Conference on Digital Libraries*. DL '00. San Antonio, Texas, USA: ACM, pp. 195–204. ISBN: 1-58113-231-X. DOI: [10.1145/336597.336662](https://doi.org/10.1145/336597.336662). URL: <http://doi.acm.org/10.1145/336597.336662>.

- Moradi, P. and S. Ahmadian (Nov. 2015). "A reliability-based recommendation method to improve trust-aware recommender systems". In: *Expert Systems with Applications* 42, pp. 7386–7398.
- Moradi, P., S. Ahmadian, and F. Akhlaghian (Oct. 2015). "An effective trust-based recommendation method using a novel graph clustering algorithm". In: *Physica A: Statistical Mechanics and its Applications* 436, pp. 462–481.
- Nakajima, S., M. Sugiyama, S. Babacan, and R. Tomioka (2013). "Global analytic solution of fully-observed variational Bayesian matrix factorization". In: *Journal of Machine Learning Research* 14, Jan, pp. 1–37.
- Nilashi, M., O. Ibrahim, and N. Ithnin (2014). "Multi-criteria collaborative filtering with high accuracy using higher order singular value decomposition and Neuro-Fuzzy system". In: *Knowledge-Based Systems* 60, pp. 82–101.
- ODonovan, J. and B. Smyth (Jan. 2005). "Trust in recommender systems". In: *Proceedings of the 10th international conference on Intelligent user interfaces* (San Diego, California, USA, Jan. 10–13, 2005), pp. 167–174.
- O'Mahony, Michael P, Neil J Hurley, and Guénolé CM Silvestre (2005). "Recommender systems: Attack types and strategies". In: *AAAI*, pp. 334–339.
- Oord, Aäron van den, Sander Dieleman, and Benjamin Schrauwen (2013). "Deep Content-based Music Recommendation". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'13. Lake Tahoe, Nevada, pp. 2643–2651.
- Ortega, F., J.L. Sánchez, J. Bobadilla, and A. Gutiérrez (Aug. 2013). "Improving collaborative filtering-based recommender systems results using Pareto dominance". In: *Information Sciences* 239, pp. 50–61.
- Ortega, F., B. Zhu, J. Bobadilla, and A. Hernando (July 2018). "CF4J: Collaborative filtering for Java". In: *Knowledge-Based Systems* 152, pp. 94–99.
- Park, C., D. Kim, J. Oh, and H. Yu (Dec. 2016). "Improving top-K recommendation with truster and trustee relationship in user trust network". In: *Information Sciences* 374, pp. 100–114.
- Park, M., J. Hong, and S. Cho (2007). "Location-based recommendation system using bayesian user's preference model in mobile devices". In: *International Conference on Ubiquitous Intelligence and Computing*. Springer, pp. 1130–1139.
- Park, Seung-Taek, David Pennock, Omid Madani, Nathan Good, and Dennis DeCoste (2006). "NaïVe Filterbots for Robust Cold-start Recommendations". In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '06. Philadelphia, PA, USA, pp. 699–705. ISBN: 1-59593-339-5.
- Pascoal, L. M. L., C. G. Camilo, E. Q. da Silva, and T. C. Rosa (July 2014). "A social-evolutionary approach to compose a similarity function used on event recommendation". In: *2014 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1512–1519.
- Paterek, A. (2007). "Improving regularized singular value decomposition for collaborative filtering". In: *Proceedings of KDD cup and workshop*. Vol. 2007, pp. 5–8.

- Pazzani, M. J. and D. Billsus (Mar. 2007). "Content-Based Recommendation Systems". In: *The Adaptive Web: Methods and Strategies of Web Personalization*. Ed. by P. Brusilovsky, A. Kobsa, and W. Nejdl. Springer, Berlin, Heidelberg. Chap. 10, pp. 325–341.
- Pennington, J., R. Socher, and C. D. Manning (2014). "GloVe: Global Vectors for Word Representation". In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543.
- Poirier, Damien, Isabelle Tellier, Françoise Fessant, and Julien Schluth (2010). "Towards text-based recommendations". In: *Adaptivity, Personalization and Fusion of Heterogeneous Information*, pp. 136–137.
- Portugal, Ivens, Paulo Alencar, and Donald Cowan (2018). "The use of machine learning algorithms in recommender systems: A systematic review". In: *Expert Systems with Applications* 97, pp. 205 –227. ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2017.12.020>. URL: <http://www.sciencedirect.com/science/article/pii/S0957417417308333>.
- Qiao, Zhi, Peng Zhang, Yanan Cao, Chuan Zhou, Li Guo, and Binxing Fang (2014). "Combining Heterogenous Social and Geographical Information for Event Recommendation." In: *AAAI*. Vol. 14, pp. 145–151.
- Ramlatchan, A., M. Yang, Q. Liu, M. Li, J. Wang, and Y. Li (2018). "A survey of matrix completion methods for recommendation systems". In: *Big Data Mining and Analytics* 1.4.
- Rendle, S. (2010). "Factorization machines". In: *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, pp. 995–1000.
- Rendle, S. and C. Freudenthaler (Feb. 2014). "Improving pairwise learning for item recommendation from implicit feedback". In: *Proceedings of the 7th ACM international conference on Web search and data mining* (New York, USA, Feb. 24–28, 2014). WSDM '14, pp. 273–282.
- Rennie, J. DM. and N. Srebro (2005). "Fast maximum margin matrix factorization for collaborative prediction". In: *Proceedings of the 22nd international conference on Machine learning*. ACM, pp. 713–719.
- Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl (1994). "GroupLens: An Open Architecture for Collaborative Filtering of Netnews". In: *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*. CSCW '94. Chapel Hill, North Carolina, USA: ACM, pp. 175–186. ISBN: 0-89791-689-1. DOI: [10.1145/192844.192905](https://doi.acm.org/10.1145/192844.192905). URL: <http://doi.acm.org/10.1145/192844.192905>.
- Rezaeimehr, F., P. Moradi, S. Ahmadian, N.N. Qader, and M. Jalili (Jan. 2018). "TCARS: Time- and Community-Aware Recommendation System". In: *Future Generation Computer Systems* 78, pp. 419–429.
- Ricci, F., L. Rokach, and B. Shapira (Feb. 2011). "Introduction to recommender systems handbook". In: *Recommender Systems Handbook*, pp. 1–35.
- Ricci, F., L. Rokach, B. Shapira, and P.B. Kantor (2011). *Recommender Systems Handbook*. Springer.

- Rocchio, J. (Mar. 1971). "Relevance Feedback Information Retrieval". In: *The SMART retrieval system - experiments in automated document processing*. Ed. by G. Salton. Prentice-Hall, Englewood Cliffs, NJ, pp. 313–323.
- Rumelhart, D. E., G. E. Hinton, and R. J. Williams (Oct. 1986). "Learning representations by back-propagating errors". In: *Nature* 323, 533–536.
- Said, Alan and Alejandro Bellogín (2014). "Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks". In: *Proceedings of the 8th ACM Conference on Recommender Systems*. RecSys '14. California, USA, pp. 129–136. ISBN: 978-1-4503-2668-1.
- Salakhutdinov, R. and A. Mnih (2008). "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo". In: *Proceedings of the 25th international conference on Machine learning*. ACM, pp. 880–887.
- Sarwar, B., G. Karypis, J. Konstan, and J. Riedl (2002). "Incremental singular value decomposition algorithms for highly scalable recommender systems". In: *Fifth International Conference on Computer and Information Science*. Citeseer, pp. 27–28.
- Seo, Y., Y. Kim, E. Lee, and D. Baik (2017). "Personalized recommender system based on friendship strength in social network services". In: *Expert Systems with Applications* 69, pp. 135–148.
- Shambour, Q. and J. Lu (Dec. 2012). "A trust-semantic fusion-based recommendation approach for e-business applications". In: *Decision Support Systems* 54, pp. 768–780.
- Shani, Guy and Asela Gunawardana (2011). "Evaluating recommendation systems". In: *Recommender systems handbook*. Springer, pp. 257–297.
- Shardanand, U. and P. Maes (1995). "Social Information Filtering: Algorithms for Automating "Word of Mouth"". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '95. Denver, Colorado, USA: ACM Press/Addison-Wesley Publishing Co., pp. 210–217. ISBN: 0-201-84705-1. DOI: [10.1145/223904.223931](https://doi.org/10.1145/223904.223931). URL: <http://dx.doi.org/10.1145/223904.223931>.
- Shi, Y., M. Larson, and A. Hanjalic (July 2014). "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges". In: *ACM Computing Surveys (CSUR)* 47, Article No. 3.
- Song, Yang, Ali Mamdouh Elkahky, and Xiaodong He (2016). "Multi-Rate Deep Learning for Temporal Recommendation". In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '16. Pisa, Italy, pp. 909–912. ISBN: 978-1-4503-4069-4.
- Srebro, N., J. Rennie, and T. S. Jaakkola (2005). "Maximum-margin matrix factorization". In: *Advances in neural information processing systems*, pp. 1329–1336.
- Srebro, N. and A. Shraibman (2005). "Rank, trace-norm and max-norm". In: *International Conference on Computational Learning Theory*. Springer, pp. 545–560.
- Su, X. and T.M. Khoshgoftaar (Jan. 2009). "A survey of collaborative filtering techniques". In: *Advances in Artificial Intelligence* 2009, Article No. 4.

- Swearingen, K. and R. Sinha (Oct. 2001). "Beyond algorithms: An HCI perspective on recommender systems". In: *Proceedings of the ACM SIGIR 2001 Workshop on Recommender Systems*.
- Taieb, M. A. H., M. B. Aouicha, and A. B. Hamadou (Sept. 2013). "Computing semantic relatedness using Wikipedia features". In: *Knowledge-Based Systems* 50, pp. 260–278.
- Tang, Jie, Sen Wu, Jimeng Sun, and Hang Su (2012). "Cross-domain collaboration recommendation". In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 1285–1293.
- Treerattanapitak, K. and C. Jaruskulchai (2012). "Exponential fuzzy C-means for collaborative filtering". In: *Journal of computer science and technology* 27.3, pp. 567–576.
- Turk, Ahmet Murat and Alper Bilge (2018). "Robustness analysis of multi-criteria collaborative filtering algorithms against shilling attacks". In: *Expert Systems with Applications*.
- Ujjin, Supiya and Peter J Bentley (2003). "Particle swarm optimization recommender system". In: *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*. IEEE, pp. 124–131.
- University, Princeton (2018). WordNet. URL: <https://wordnet.princeton.edu/> (visited on 05/30/2018).
- Vasile, F., E. Smirnova, and A. Conneau (Sept. 2016). "Meta-Prod2Vec: Product Embeddings Using Side-Information for Recommendation". In: *Proceedings of the 10th ACM Conference on Recommender Systems*. RecSys '16. Boston, Massachusetts, USA, pp. 225–232.
- Verstrepen, Koen and Bart Goethals (2014). "Unifying Nearest Neighbors Collaborative Filtering". In: *Proceedings of the 8th ACM Conference on Recommender Systems*. RecSys '14. California, USA, pp. 177–184.
- Wang, H., N. Wang, and D-Y. Yeung (2015). "Collaborative deep learning for recommender systems". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 1235–1244.
- Wang, J., W.M. Deng, and X.Y. Yin (Feb. 2016). "Electronic commerce recommendation mobile crowd system based on cooperative data collection and embedded control". In: *EURASIP Journal on Embedded Systems* 1, pp. 1–7.
- Wang, Jun, Arjen P. de Vries, and Marcel J. T. Reinders (2006). "Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion". In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '06. Seattle, Washington, USA, pp. 501–508.
- Wang, N. and D. Y. Yeung (Dec. 2013). "Bayesian Robust Matrix Factorization for Image and Video Processing". In: *2013 IEEE International Conference on Computer Vision*, pp. 1785–1792.

- Wang, N., T. Yao, J. Wang, and D-Y. Yeung (2012). "A probabilistic approach to robust matrix factorization". In: *European Conference on Computer Vision*. Springer, pp. 126–139.
- Wang, Shuaiqiang, Shanshan Huang, Tie-Yan Liu, Jun Ma, Zhumin Chen, and Jari Veijalainen (2016). "Ranking-oriented collaborative filtering: A listwise approach". In: *ACM Transactions on Information Systems (TOIS)* 35.2, p. 10.
- Wang, W., G.Q. Zhang, and J. Lu (Aug. 2015). "Collaborative Filtering with Entropy-Driven User Similarity in Recommender Systems". In: *International Journal of Intelligent Systems, Special Issue: Intelligent Techniques for Data Science* 30, pp. 854–870.
- Webb, B. (2006). *Netflix update: Try this at home*. URL: <http://sifter.org/simon/journal/20061211.html> (visited on 07/20/2018).
- Weimer, M., A. Karatzoglou, Q. V. Le, and A. J. Smola (2008). "Cofi rank-maximum margin matrix factorization for collaborative ranking". In: *Advances in neural information processing systems*, pp. 1593–1600.
- wikipedia (2018). *wikipedia*. URL: <https://www.wikipedia.org/> (visited on 05/30/2018).
- Wilson, David C. and Carlos E. Seminario (2013). "When Power Users Attack: Assessing Impacts in Collaborative Recommender Systems". In: *Proceedings of the 7th ACM Conference on Recommender Systems*. RecSys '13. Hong Kong, China, pp. 427–430. ISBN: 978-1-4503-2409-0.
- Wu, L., A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston (Feb. 2018). "StarSpace: Embed All The Things!" In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI-18.
- Wu, W., L. He, and J. Yang (Aug. 2012). "Evaluating recommender systems". In: *Proceedings of the 2012 Seventh International Conference on Digital Information Management* (Macau, China, Aug. 22–24, 2012). ICDIM '12, pp. 56–61.
- Wu, Zhiang, Junjie Wu, Jie Cao, and Dacheng Tao (2012). "HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation". In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 985–993.
- Xiao, Xiangye, Yu Zheng, Qiong Luo, and Xing Xie (2014). "Inferring social ties between users with human location history". In: *Journal of Ambient Intelligence and Humanized Computing* 5.1, pp. 3–19.
- Yang, C., L. Bai, C. Zhang, Q. Yuan, and J. Han (2017). "Bridging Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendation". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '17. Halifax, NS, Canada: ACM, pp. 1245–1254. ISBN: 978-1-4503-4887-4. DOI: [10.1145/3097983.3098094](https://doi.org/10.1145/3097983.3098094). URL: <http://doi.acm.org/10.1145/3097983.3098094>.
- Yang, Dingqi, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu (2015). "Modeling user activity preference by leveraging user spatial temporal characteristics in LB-SNs". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45.1, pp. 129–142.

- Yang, Z., B. Wu, K. Zheng, X. Wang, and L. Lei (May 2016). "A Survey of Collaborative Filtering-Based Recommender Systems for Mobile Internet Applications". In: *IEEE Access* 4, pp. 3273–3287.
- Yera, Raciél and Luis Martínez (2017). "Fuzzy tools in recommender systems: A survey". In: *International Journal of Computational Intelligence Systems* 10.1, pp. 776–803.
- Yin, Hongzhi, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen (2013). "LCARS: a location-content-aware recommender system". In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 221–229.
- Yin, P., M. Ye, W. Lee, and Z. Li (2014). "Mining GPS Data for Trajectory Recommendation". In: *Advances in Knowledge Discovery and Data Mining*. Cham: Springer International Publishing, pp. 50–61.
- Ying, Haochao, Jian Wu, Guandong Xu, Yanchi Liu, Tingting Liang, Xiao Zhang, and Hui Xiong (2018). "Time-aware metric embedding with asymmetric projection for successive POI recommendation". In: *World Wide Web*, pp. 1–16.
- Ying, J., W. Lee, M. Ye, C. Chen, and V. Tseng (2011). "User association analysis of locales on location based social networks". In: *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*. ACM, pp. 69–76.
- Yuan, W., D. Guan, Y.-K. Lee, S. Lee, and S.J. Hur (Apr. 2010). "Improved trust-aware recommender system using small-worldness of trust networks". In: *Knowledge-Based Systems* 23, pp. 232–238.
- Zhang, F., Y. Lu, J. Chen, and S. Liu (Feb. 2017a). "Robust collaborative filtering based on non-negative matrix factorization and R_1 -norm". In: *Knowledge-Based Systems* 118, pp. 177–190.
- Zhang, M., X. Guo, and G. Chen (Mar. 2016a). "Prediction uncertainty in collaborative filtering: Enhancing personalized online product ranking". In: *Decision Support Systems* 83, pp. 10–21.
- Zhang, M. and N. Hurley (2009). "Statistical modeling of diversity in top-n recommender systems". In: *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*. IEEE Computer Society, pp. 490–497.
- Zhang, Mi and Neil Hurley (2008). "Avoiding monotony: improving the diversity of recommendation lists". In: *Proceedings of the 2008 ACM conference on Recommender systems*. ACM, pp. 123–130.
- Zhang, Mingyue, Xunhua Guo, and Guoqing Chen (2016b). "Prediction uncertainty in collaborative filtering: Enhancing personalized online product ranking". In: *Decision Support Systems* 83, pp. 10–21.
- Zhang, Shuai, Lina Yao, and Aixin Sun (2017). "Deep learning based recommender system: A survey and new perspectives". In: *arXiv preprint arXiv:1707.07435*.
- Zhang, Yongfeng, Qingyao Ai, Xu Chen, and W. Bruce Croft (2017b). "Joint Representation Learning for Top-N Recommendation with Heterogeneous Information Sources". In: *Proceedings of the 2017 ACM on Conference on Information and*

- Knowledge Management*. CIKM '17. Singapore, Singapore, pp. 1449–1458. ISBN: 978-1-4503-4918-5.
- Zhao, X., Y. Guo, Y. He, H. Jiang, Y. Wu, and X. Li (2014). “We Know What You Want to Buy: A Demographic-based System for Product Recommendation on Microblogs”. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '14. New York, New York, USA: ACM, pp. 1935–1944. ISBN: 978-1-4503-2956-9. DOI: [10.1145/2623330.2623351](https://doi.org/10.1145/2623330.2623351). URL: <http://doi.acm.org/10.1145/2623330.2623351>.
- Zhao, Xiangguo, Zhongyu Ma, and Zhen Zhang (2017). “A novel recommendation system in location-based social networks using distributed ELM”. In: *Memetic Computing*, pp. 1–11.
- Zheng, M., F. Min, H. R. Zhang, and W. B. Chen (Apr. 2016). “Fast recommendations with the m-distance”. In: *IEEE Access* 4, pp. 1464–1468.
- Zheng, Vincent Wenchen, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang (2010). “Collaborative Filtering Meets Mobile Recommendation: A User-Centered Approach.” In: *AAAI*. Vol. 10, pp. 236–241.
- Zheng, Yu, Lizhu Zhang, Zhengxin Ma, Xing Xie, and Wei-Ying Ma (2011). “Recommending friends and locations based on individual location history”. In: *ACM Transactions on the Web (TWEB)* 5.1, p. 5.
- Zhou, Wei, Junhao Wen, Qingyu Xiong, Min Gao, and Jun Zeng (2016). “SVM-TIA a shilling attack detection method based on SVM and target item analysis in recommender systems”. In: *Neurocomputing* 210, pp. 197–205.
- Zhou, Y., D. Wilkinson, R. Schreiber, and R. Pan (2008). “Large-scale parallel collaborative filtering for the netflix prize”. In: *International Conference on Algorithmic Applications in Management*. Springer, pp. 337–348.
- Zhu, B., R. Hurtado, J. Bobadilla, and F. Ortega (May 2018a). “An efficient recommender system method based on the numerical relevances and the non-numerical structures of the ratings”. In: *IEEE Access*.
- Zhu, B., F. Ortega, J. Bobadilla, and A. Gutiérrez (Apr. 2018b). “Assigning reliability values to recommendations using matrix factorization”. In: *Journal of Computational Science* 26, 165–177.