



T.C.

AKSARAY UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE

**DEPARTMENT OF ELECTRICAL-ELECTRONIC AND
COMPUTER ENGINEERING**

**ADOPTING SEVERAL MODELS TO ALLEVIATE SPARSITY
AND COLD-START PROBLEMS USING DATA MINING
TECHNIQUES FOR RECOMMENDATION SYSTEMS**

DOCTORAL THESIS

Ali Mohsin Ahmed AL-SABAAWI

SUPERVISOR

Assist. Prof. Dr. Yusuf Erkan YENICE

Assoc. Prof. Dr. Hacer KARACAN

AKSARAY, 2021

Aksaray University, Graduate School of Natural and Applied Sciences Approval, Ali Mohsin Ahmed AL-SABAAWI, Doctor of Philosophy with thesis, student No. 172353801 successfully presented the Doctor of Philosophy Thesis titled “**ADOPTING SEVERAL MODELS TO ALLEVIATE SPARSITY AND COLD-START PROBLEMS USING DATA MINING TECHNIQUES FOR RECOMMENDATION SYSTEMS**”, prepared after satisfying all the requirements identified in the concerned bylaws, before the jury whose signatures are affixed below.

Thesis Advisor: Asst. Prof. Dr. Yusuf Erkan YENICE

Aksaray University

I approve that this thesis is a Doctor of Philosophy in scope and quality

.....

Member: Assoc. Prof. Dr. Özkan Ufuk NALBANTOĞLU

Erciyes University

I approve that this thesis is a Doctor of Philosophy in scope and quality

.....

Member: Assoc. Prof. Dr. İsmail BAYRAKLI

Aksaray University

I approve that this thesis is a Doctor of Philosophy in scope and quality

.....

Member: Prof. Dr. Mehmet Resit TOLUN

Konya Food and Agriculture University

I approve that this thesis is a Doctor of Philosophy in scope and quality

.....

Member: Assoc. Prof. Dr. Emre ARSLAN

Aksaray University

I approve that this thesis is a Doctor of Philosophy in scope and quality

.....

Date of Defense: 3 / 2 / 2021

I agree that this thesis which is accepted by the jury, has fulfilled the requirements for being a Doctor of Philosophy Thesis.

.....

Assoc. Prof. Dr. Mehmet Ali HINIS
Director of Natural and Applied Science

DECLARATION

The idea for this Ph.D. thesis was taken from the literature regarding the topics. It was announced here that the whole work presented in this project has been composed and originated by myself unless otherwise specified. The study was conducted completely under the supervision of Asst. Prof. Dr. Yusuf YENICE and Assoc. Prof. Dr. Hacer KARACAN.

Ali Mohsin Ahmed AL-SABAAWI



ACKNOWLEDGEMENT

Full of tears of Happiness and joy, my gratitude and praising go to God. As the eye can see, only the One is the originator and the doer of this thesis, yet many blokes bear a hand to the attainment of this dissertation. Without them, the latter may not even be completed, problems would not be solved, and certainly changes would not occur for the benefit of others. Onset, I would love to thank my dear supervisor Prof. Dr. Yusuf YENICE for his guidance and instructions; he guided me with patience and respect all the way long during my research journey. I owe my laud to him. Also, I would like to thank my dear co-supervisor Prof. Dr. Hacer KARACAN who in the other hand was the eyes, the oversight and the stakeout of this research. There are no words can describe my gratitude to her, I owe my appreciation, love and respect toward her, without her guidance this thesis may not even have existed. Moreover, my sincere gratitude goes to the members of monitoring thesis committee Prof. Dr. Ufuk NALBANTOĞLU and Assoc. Prof. Dr. Ismail BAYRAKLI for their valuable comments of this dissertation.

I dedicate my thesis to my heaven-sent parents who are and still the eye of the inspiration, the key of my transformation and the light of my revelation. All the words will stand frozen when it comes to their estimation. My estimations go also to my brothers and sisters for their prayers, love, and support. A deep appreciation goes to my beloved wife who in the other hand helped be conduct this research with her appreciated actions: “a good wife makes a good husband”, thank you to fulfilling my life. A specific appreciation goes to my beautiful children Malek and Harith, I am making a lot of sacrifices so that you can have the life you deserve. Ultimately, all appreciation goes to my friends, who stood by my side and aided me, I want to tell them, may Allah bless you all. All the latter deserve the credit for developing the premise of this thesis and May their time and efforts be fulfilled with success and prosperity.

Ali Mohsin Ahmed AL-SABAAWI
AKSARAY, 2021

TABLE OF CONTENTS

ACKNOWLEDGEMENT	i
TABLE OF CONTENTS	ii
ÖZET	vi
ABSTRACT	vii
LIST OF FIGURES	viii
LIST OF TABLES	xi
LIST OF ABBRIVIATIONS	xiii
1. INTRODUCTION	1
1.1 Overview	1
1.2 Problem Background	4
1.3 Problem Statements	8
1.4 Objectives of the Research	9
1.5 Scope of the Study	10
1.6 Significance of The Study	10
1.7 Contributions of the Study	11
1.8 Thesis Outline	11
2. LITERATURE REVIEW	13
2.1 Overview of Recommender System	13
2.2 Feedback Information on the Recommender Systems	14
2.2.1 User-Item rating matrix	15
2.3 Types of The Recommender Systems	15
2.3.1 Collaborative filtering approach	16
2.3.1.1 Memory-based approach	17
2.3.1.2 Model-based approach	19
2.3.2 Content-based approach	27
2.3.3 Knowledge-based system	28
2.3.4 Hybrid Approach	29
2.4 Challenges of Recommendations Systems	32
2.4.1 Data sparsity	32
2.4.2 Cold-start	32
2.4.3 Scalability	33
2.4.4 Synonymy	34
2.4.5 Privacy preserving	34
2.4.6 Gray sheep	35
2.4.7 Shilling attack	35
2.5 Distance and Similarity Measurement	36
2.6 Evaluation Metrics	41
2.6.1 Accuracy metrics	41
2.6.1.1 Statistical accuracy metrics	41
2.6.1.2 Decision support accuracy metrics	42
2.6.2 Rank-based metrics	43
2.6.3 Rate coverage	45
2.7 Social Network	45
2.7.1 Friends network-based	46
2.7.2 Trust network-based	47
2.8 Datasets	51
2.9 Summary	52
3. RESEARCH METHODOLOGY	53

3.1 Research Design	53
3.2 Research Framework	54
3.2.1 Phase 1: Literature review and data preparation.....	55
3.2.2 Phase 2: Integrating explicit, implicit social relations and explicit feedback information into SVD.....	61
3.2.3 Phase 3: Exploiting social relations and rating feedback information and integrating them into SVD++.....	62
3.2.4 Phase 4: Clustering community detection	63
3.2.5 Phase 5: Overlapping community detection.....	64
3.2.6 Phase 6: System evaluation	65
3.3 Experimental Preparations.....	67
3.3.1 Comparisons with the proposed approach.....	68
3.3.1.1 Comparison with rating-based methods.....	68
3.3.1.2 Comparison with the state-of-the-art methods.....	68
3.4 Parameters Setting.....	70
3.5 Summary	72
4. EXPLOITING IMPLICIT SOCIAL RELATIONSHIPS VIA DIMENSION REDUCTION TO IMPROVE RECOMMENDATION SYSTEM PERFORMANCE.....	73
4.1 Data Sources.....	73
4.1.1 Link prediction.....	75
4.1.2 Multi-steps resource allocation	75
4.1.3 Singular value decomposition (SVD)	78
4.2 The MSRA-SVD Model	80
4.3 Experimental Preparation.....	83
4.3.1 Description of datasets	84
4.3.2 Experimental setup.....	84
4.3.3 Results and discussion.....	85
4.3.3.1 Statistic accuracy results	85
4.3.3.2 Impact of different dimension values	87
4.3.3.3 Impact of changing lambda values	91
4.3.3.4 Impact of changing beta values	94
4.3.3.5 Impact of changing gamma values	96
4.3.3.6 Decision support accuracy results	99
4.4 Summary	101
5. INTEGRATING SOCIAL RELATIONS AND FEEDBACK INFORMATION INTO SVD++ TO ENHANCE PREDICTION ACCURACY.....	102
5.1 Data Sources.....	103
5.2 Multi-step resource allocation (MSRA)	104
5.3 The SVD++ method	105
5.4 The MSRA-SVD++ model	107
5.5 Experimental Preparation.....	110
5.5.1 Description of datasets	110
5.5.2 Experimental setup.....	111
5.5.3. Results and analysis	112
5.5.3.1 Statistic accuracy metrics.....	112
5.5.3.2. Impact of different dimensions values.....	114
5.5.3.3 Impact of changing beta values	119
5.5.3.4 Impact of changing gamma values	122

5.5.3.5 Decision support accuracy results	125
5.6 Summary	127
6. SVD++ AND CLUSTERING APPROACHES TO ALLEVIATE THE COLD-START PROBLEM FOR RECOMMENDATION SYSTEMS	129
6.1 An Overview of the Proposed Model	129
6.2 Methods and Data Sources.....	130
6.2.1 Extraction of social relations	130
6.2.1.1 Extraction explicit social relations	131
6.2.1.2 Extraction implicit social relations	131
6.2.2 Confidence values	132
6.2.3 Partitioning Around Medoids (PAM) Algorithm	135
6.2.4 SVD++ method	136
6.3 Proposed Model.....	136
6.4 Experimental Evaluation.....	140
6.4.1 Description of the datasets.....	140
6.4.2 Experimental setup.....	140
6.4.3 Results and discussion.....	141
6.4.3.1 Determining the best cluster.....	141
6.4.3.2 Benchmarking with our previous models	142
6.4.3.3 Benchmarking with state-of-the-art methods.....	143
6.4.3.4 Impact of implicit social relations	145
6.4.3.5 Impact of different lambda values	146
6.4.3.6 Impact of using confidence values	148
6.4.3.7 Decision support system results	150
6.5 Summary	151
7. A NOVEL OVERLAPPING METHOD TO ALLEVIATE THE COLD-START PROBLEM IN RECOMMENDATION SYSTEMS	153
7.1 An Overview of the Proposed Model	153
7.2 Methods and Data Sources.....	154
7.2.1 Extract explicit social relations.....	154
7.2.2 Extract implicit social relations	155
7.2.3 Confidence values	155
7.2.4 Partitioning Around Medoids (PAM) algorithm	156
7.2.5 Novel overlapping method	157
7.2.6 SVD++ method.....	158
7.3 Proposed Model.....	159
7.4 Experimental Evaluation.....	161
7.4.1 Description of the datasets.....	161
7.4.2 Experimental setup.....	162
7.4.3 Results and discussion.....	163
7.4.3.1 Determining the best cluster.....	163
7.4.3.2 Benchmarking with our previous models	164
7.4.3.3 Benchmarking with state-of-the-art methods.....	165
7.4.3.4 Impact of different lambda values	166
7.4.3.5 Decision support system results	168
7.5 Summary	170
8. CONCLUSIONS AND FUTURE WORK.....	172
8.1 Conclusions of the Study	172
8.2 Future Work	172
REFERENCES	174

CURRICULUM VITAE	187
-------------------------------	------------



DOKTORA TEZİ

TAVSİYE SİSTEMİ İÇİN VERİ MADENCİLİĞİ TEKNİKLERİNİ KULLANARAK VERİ SEYREKLİĞİNİ VE SOĞUK BAŞLATMA SORUNLARINI HAFİFLETMEK İÇİN ÇEŞİTLİ MODELLER

Ali Mohsin Ahmed AL-SABAAWI

**Aksaray Üniversitesi
Fen Bilimleri Enstitüsü**

Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Öğr. Üyesi Dr. Yusuf YENICE and Doç. Dr. Hacer KARACAN

ÖZET

Web 2.0'ın geliştirilmesi ve elde edilebilir verilerin hızlı artışı bilgi aşırı yüklemesinin üstesinden gelebilen çoklu sistemlerin ve onlar arasında da Tavsiye Sistemleri'nin evrilmesine yol açmıştır. Tavsiye Sistemleri'nin performansının büyük ölçüde seyreklik ve soğuk-başlatma problemleri tarafından sınırlandırılmış olması nedeniyle, mevcut araştırma bu problemlerin hafifletilmesini temel bir amaç olarak benimsemiştir. Bu amacın gerçekleştirilmesi için, şu dört veri madenciliği tekniği önerilmiştir: çok-aşamalı kaynak ayırma - tekil değer ayrışması (MSRA-SVD), MSRA-SVD++, kümelenen topluluk tespiti ve örtüşen topluluk tespiti. İlk iki model veri seyrekliği probleminin üstesinden gelme amacına adanmışken, kalan modeller ise soğuk-başlatma kullanıcılarının problemlerinin hafifletilmesi amacıyla benimsenmektedir. İlk iki modelin temel stratejisi sosyal ağdaki gizli ilişkilerin belirlenmesi için (MSRA) yönteminin kullanılmasıdır. MSRA yöntemi ilişki olasılığının belirlenmesi amacıyla uygulanır. Eğer olasılık bir eşik düzeyini aşarsa, bu durumda yeni bir ilişki kurulmuş olacaktır. İkinci model için (MSRA-SVD++), derecelendirme bilgileri yoluyla elde edilebilecek olan örtük geribildirim kaynağı ek bir bilgi kaynağı olarak kullanılır. Buna ek olarak, ve örtüşme modelleri de soğuk-başlatma kullanıcılarının problemlerinin hafifletilmesi amacıyla benimsenmektedir. Bu modellerin ardında yatan kümelenme temel düşünce kullanıcıların çeşitli topluluklar halinde kümelenmesi için bir kümeleme tekniğinin uygulanmasıdır. Bunun elde edilebilmesi için de, güven değerine sahip olan açık ve örtük sosyal ilişkiler entegre edilerek uzaklık değerleri hesaplanır. Buna ek olarak, medoidler etrafındaki bölüntülenme ile ilgili olarak da kümelenme algoritması benimsenir. Daha sonra, son model için tüm kümelerin ortalama uzaklıkları hesaplanır. Tüm kullanıcılar için, kullanıcılar ve belli bir kümenin merkez düğümü arasındaki mesafe hesaplanır. Eğer bu mesafe söz konusu kümenin ortalamasından daha azsa, o küme için yeni bir kullanıcı eklenecektir. Dahası, tahmin değerinin hesaplanması amacıyla her bir küme için SVD++ yöntemi kullanılır. Önerilen modeller üç adet gerçek-dünya veri kümesi kullanılarak değerlendirilir. Sonuç olarak, bulgular önerilen modellerin çeşitli en yeni çalışmaları tahmin doğruluğu bakımından nasıl geride bıraktıklarına dair kayda değer bir anlayışı sergilemiştir.

Anhtar Kalimler: Tavsiye Sistemleri, Seyreklik, Soğuk başlangıç, İşbirliğine dayalı filtreleme, veri madenciliği, Kümelenme, Örtüşme.

Şubat, 2021; 187 sayfa

Ph.D. THESIS

ADOPTING SEVERAL MODELS TO ALLEVIATE SPARSITY AND COLD-START PROBLEMS USING DATA MINING TECHNIQUES FOR RECOMMENDATION SYSTEMS

Ali Mohsin Ahmed AL-SABAAWI

**Aksaray University
Graduate School of Natural and Applied Science
Department of Electrical-Electronic and Computer Engineering**

**Supervisors: Asst. Prof. Dr. Yusuf YENICE
and
Assoc. Prof. Dr. Hacer KARACAN**

ABSTRACT

The development of Web 2.0 and the rapid growth of available data have led to the evolution of multiple systems among them the Recommendation Systems (RSs) which can handle the information overload. Due to the fact that, RSs performance is substantially limited by sparsity and cold-start problems; this research study takes the route for a major objective to attenuate these problems. To realize this objective, four data mining techniques are proposed namely: multi-steps resource allocation- singular value decomposition (MSRA-SVD), MSRA-SVD++, clustering community detection, and overlapping community detection. The first two models are dedicated to tackle the data sparsity problem whereas the rest models are adopted to alleviate cold-start users' problem. The core strategy of the first two models is to use the (MSRA) method to identify hidden relations in social network. the MSRA method is applied to determine the probability of their relation. If the probability exceeds a threshold, a new relationship will be established. For the second model (MSRA-SVD++), an implicit feedback source is exploited as an additional source of information, which can be extracted via rating information. Additionally, clustering and overlapping models are adopted to overcome cold-start user problem. The main idea of these models is to apply a clustering technique to group users into several communities. In order to attain that, explicit and implicit social relations with the confidence values are integrated to compute distance values. Additionally, the partitioning around medoids (PAM) clustering algorithm is adopted. Later, for the last model, the average distances of all clusters are computed. For all users, the distance between users and the center node of a particular cluster is computed. If the distance is less than the average of this cluster, a new user for this cluster will be added. Moreover, the SVD++ method is employed for each cluster to compute the prediction value. The proposed models are evaluated through the usage of three real-world datasets. Ultimately, findings exhibited a great deal of insights on how the proposed models outperformed a number of the state-of-the-art studies in terms of prediction accuracy.

Keywords: Recommendation systems, Sparsity, Cold-start, Collaborative filtering, Data mining, Clustering, Overlapping.

February, 2021; 187 pages

LIST OF FIGURES

Figure 2.1. Types of Recommender systems.	16
Figure 2.2. The architecture of the memory-based approach.	18
Figure 2.3. Decomposition process of SVD method.	21
Figure 2.4. A simple overlapping of two communities.	26
Figure 2.5. Content-Based approach architecture.	28
Figure 2.6. Hybrid system framework.	30
Figure 2.7. Sample of sparsity problem.	32
Figure 2.8. Cold-start problem.	33
Figure 2.9. Example of friends' relationships.	46
Figure 2.10. Example of directed relationships.	48
Figure 2.11. Sample of explicit relationships.	48
Figure 2.12. A user-user matrix for explicit relations.	49
Figure 3.1. Research framework.	55
Figure 3.2. Illustrating of 5-fold cross-validation.	60
Figure 3.3. General view of the proposed models.	60
Figure 3.4. Summary of phase 2.	61
Figure 3.5. Summary of phase 3.	63
Figure 3.6. An overview of the clustering community detection.	64
Figure 3.7. An overview of overlapping community detection.	65
Figure 4.1. An example of data sources for the proposed model.	74
Figure 4.2. Social network nodes.	77
Figure 4.3. An overview of MSRA algorithm.	78
Figure 4.4. The decomposition process of the SVD method.	79
Figure 4.5. Overview of the MSRA-SVD model.	83
Figure 4.6. MAE results for various numbers of dimensions for FilmTrust dataset.	88
Figure 4.7. RMSE results for various numbers of dimensions for FilmTrust dataset.	89
Figure 4.8. MAE results for various numbers of dimensions for Ciao dataset.	89
Figure 4.9. RMSE results for various numbers of dimensions for Ciao dataset.	90
Figure 4.10. MAE results for various numbers of dimensions for Last.Fm.	90
Figure 4.11. RMSE results for various numbers of dimensions for Last.Fm.	91
Figure 4.12. Impact of different lambda values on MAE results for FilmTrust.	92
Figure 4.13. Impact of different lambda values on RMSE results for FilmTrust.	92
Figure 4.14. Impact of different lambda values on MAE results for Ciao.	92
Figure 4.15. Impact of different lambda values on RMSE results for Ciao.	93
Figure 4.16. Impact of different lambda values on MAE results for Last.Fm.	93
Figure 4.17. Impact of different lambda values on RMSE results for Last.Fm.	93
Figure 4.18. Impact of different beta values on MAE results for FilmTrust.	94
Figure 4.19. Impact of different beta values on RMSE results for FilmTrust.	95
Figure 4.20. Impact of different beta values on MAE results for Ciao.	95

Figure 4.21. Impact of different beta values on RMSE results for Ciao.....	95
Figure 4.22. Impact of different beta values on MAE results for Last.Fm.	96
Figure 4.23. Impact of different beta values on RMSE results for Last.Fm.	96
Figure 4.24. Impact of different gamma values on MAE results for FilmTrust.....	97
Figure 4.25. Impact of different gamma values on RMSE results for FilmTrust.....	97
Figure 4.26. Impact of different gamma values on MAE results for Ciao.....	98
Figure 4.27. Impact of different gamma values on RMSE results for Ciao.....	98
Figure 4.28. Impact of different gamma values on MAE results for Last.Fm.	98
Figure 4.29. Impact of different gamma values on RMSE results for Last.Fm.	99
Figure 5.1. An example of data sources for the proposed model.	104
Figure 5.2. The impact of explicit and implicit relations in SVD++ method.....	108
Figure 5.3. Overview of MSRA-SVD++.	110
Figure 5.4. MAE results for various numbers of dimensions for Ciao.....	116
Figure 5.5. RMSE results for various numbers of dimensions for Ciao.....	117
Figure 5.6. MAE results for various numbers of dimensions for Last.Fm.	117
Figure 5.7. RMSE results for various numbers of dimensions for Last.Fm.	118
Figure 5.8. MAE results for various numbers of dimensions for FilmTrust.....	118
Figure 5.9. RMSE results for various numbers of dimensions for FilmTrust.	119
Figure 5.10. Impact of different beta values on MAE results for Ciao.....	120
Figure 5.11. Impact of different beta values on RMSE results for Ciao.....	120
Figure 5.12. Impact of different beta values on MAE results for Last.Fm.	120
Figure 5.13. Impact of different beta values on RMSE results for Last.Fm.	121
Figure 5.14. Impact of different beta values on MAE results for FilmTrust.....	121
Figure 5.15. Impact of different beta values on RMSE results for FilmTrust.....	121
Figure 5.16. Impact of different gamma values on MAE results for Ciao.....	123
Figure 5.17. Impact of different gamma values on RMSE results for Ciao.....	123
Figure 5.18. Impact of different gamma values on MAE results for Last.Fm.	123
Figure 5.19. Impact of different gamma values on RMSE results for Last.Fm.	124
Figure 5.20. Impact of different gamma values on MAE results for FilmTrust.....	124
Figure 5.21. Impact of different gamma values on RMSE results for FilmTrust.....	124
Figure 6.1. An example of explicit relations.	133
Figure 6.2. The extraction process of implicit relations.....	133
Figure 6.3. A sample of a user-item matrix.....	134
Figure 6.4. An overview of the proposed model.	137
Figure 6.5. The impact of adding implicit relations.....	146
Figure 6.6. The MAE results for different lambda values for Ciao.....	147
Figure 6.7. The RMSE results for different lambda values for Ciao.....	147
Figure 6.8. The MAE results for different lambda values for FilmTrust.....	148

Figure 6.9. The RMSE results for different lambda values for FilmTrust.....	148
Figure 7.1. An overview of the proposed model.	161
Figure 7.2. The MAE results for different lambda values for Ciao.....	167
Figure 7.3. The RMSE results for different lambda values for Ciao.....	167
Figure 7.4. The MAE results for different lambda values for FilmTrust.....	168
Figure 7.5. The RMSE results for different lambda values for FilmTrust.....	168



LIST OF TABLES

Table 2.1. Examples of the user-item rating matrix.	13
Table 2.2. Examples of the users' profile.	13
Table 2.3. Examples of the Items' profile.	14
Table 2.4. Incorporated CB and CF.	31
Table 2.5. An example of a shilling attack.	36
Table 2.6. Similarity measures periodically used for CF approach.	38
Table 2.7. Rating matrix.	39
Table 2.8. Matching objects.	40
Table 2.9. Two binary data vectors.	40
Table 2.10. A confusion matrix.	43
Table 2.11. Summary of the common datasets in the RSs.	52
Table 3.1. Statistical of the sample data of Last.Fm.	58
Table 3.2. Statistical of the sample data of Ciao.	58
Table 3.3. Statistical of FilmTrust.	59
Table 3.4. Outline of the thesis framework.	67
Table 3.5. Summarizing of the comparisons of the proposed method.	70
Table 3.6. Parameters setup for MSRA-SVD model.	71
Table 3.7. Parameters setup for MSRA-SVD++ model.	71
Table 3.8. Parameters setup for clustering community detection models.	71
Table 4.1. Statistics of the datasets.	84
Table 4.2. Parameter setup.	84
Table 4.3. Evaluation performance for non-cold-start users for FilmTrust. Boldface indicates the best results and (*) indicates second-top results.	86
Table 4.4. Evaluation performance of non-cold-start users for Ciao. Boldface indicates the best results and (*) indicates second-top results.	87
Table 4.5. Evaluation performance of non-cold-start users for Last.Fm. Boldface indicates the best results and (*) indicates second-top results.	87
Table 4.6. Precision results.	100
Table 4.7. Recall results.	100
Table 4.8. F-measure results.	101
Table 5.1. Statistics of the datasets.	111
Table 5.2. Parameter setup.	111
Table 5.3. Evaluation performance of non-cold-start users for Ciao. Boldface implies best results and (*) indicates the second-best results.	114
Table 5.4. Evaluation performance of non-cold-start users for Last.fm. Boldface implies best results and (*) indicates the second-best results.	114
Table 5.5. Evaluation performance of non-cold-start users for FilmTrust Boldface implies best results and (*) indicates the second-best results.	114
Table 5.6. Precision results.	126
Table 5.7. Recall results.	126
Table 5.8. F-measure results.	127
Table 6.1. Statistics of the FilmTrust and Ciao.	140

Table 6.2. Parameter setup.	140
Table 6.3. Accuracy results of various numbers of clusters for Ciao.	142
Table 6.4. Accuracy results of various numbers clusters for FilmTrust.	142
Table 6.5. Performance comparison for cold-start users for Ciao. Boldface implies best results and (*) indicates the second-best results.	143
Table 6.6. Performance comparison for cold-start users for FilmTrust. Boldface implies best results and (*) indicates the second-best results.	143
Table 6.7. Performance comparison for cold-start users for Ciao. Boldface implies best results and (*) indicates the second-best results.	144
Table 6.8. Performance comparison for cold-start users for FilmTrust. Boldface implies best results and (*) indicates the second-best results.	144
Table 6.9. The clustering results with and without confidence values, for Ciao dataset.	149
Table 6.10. The clustering results with and without confidence values, for FilmTrust dataset.	150
Table 6.11. Results of precision.	151
Table 6.12. Results of recall.	151
Table 6.13. Results of F-measure.	151
Table 7.1. Statistics of the FilmTrust and Ciao.	162
Table 7.2. Parameter setup.	162
Table 7.3. Accuracy results of various numbers of overlapping clusters for Ciao.	163
Table 7.4. Accuracy results of various numbers of overlapping clusters for FilmTrust.	164
Table 7.5. Performance comparison with previous models for cold-start users for Ciao. Boldface implies best results and (*) indicates the second-best results.	165
Table 7.6. Performance comparison with previous models for cold-start users for FilmTrust. Boldface implies best results and (*) indicates the second-best results.	165
Table 7.7. Performance comparison with state-of-the-art for cold-start users for Ciao. Boldface implies best results and (*) indicates the second-best results.	166
Table 7.8. Performance comparison with state-of-the-art for cold-start users for FilmTrust. Boldface implies best results and (*) indicates the second-best results.	166
Table 7.9. Results of precision.	169
Table 7.10. Results of recall.	169
Table 7.11. Results of F-measure.	170

LIST OF ABBRIVIATIONS

CB	Content Based
CCS	Complete Cold-start
CF	Collaborative Filtering
COS	Cosine Similarity
CPM	Clique percolation method
DGC	Discount cumulative gain
FN	False Negative
FP	False Positive
GN	Girvan and Newman
ICS	Incomplete Cold-start
KB	knowledge-based
K-NN	K-nearest neighbors
LP	Link Prediction
MAE	Mean Absolute Error
MF	Matrix Factorization
MSRA	Multi-steps resource allocation
NMF	Non-negative MF
PAM	Portioning Around Medoid
PCC	Pearson's Correlation Coefficient
PMF	Probabilistic MF
RAI	Resource Allocation Index
RBP	rank-biased precision
RC	Rate coverage
RMSE	Root Mean Square Error
RS	Recommendation System
SGD	Stochastic Gradient Descent
SVD	Singular Value Decomposition
SVD++	Singular Value Decomposition plus
TN	True Negative
TP	True Positive

1. INTRODUCTION

1.1 Overview

The amount of available data is growing rapidly and it is extremely difficult for users to find their preferences in it. In addition, the increased use of technology in our lives has boosted peoples' demand for intelligent systems that execute this task. These are called recommendation systems (RSs). The main objective of an RS is to predict items that might be of interest (e.g., movies, music, books, and newspapers) to users based on enormous amounts of data (Resnick and Varian, 1997; Ricci et al., 2011).

A basic RS relies on several technologies such as information filtering, user modeling, and classification learning (Sun et al., 2015). Generally, recommendation systems include collaborative filtering systems (CF), content-based systems (CB), and hybrid systems (Jamali and Ester, 2010). A CF learns a user-item matrix for recommending items. A CB utilizes users' properties to find new items (Liu et al., 2013). A hybrid system integrates CF and CB systems to recommend items.

CFs is the most prevalent type of RS that can be further be subdivided into memory-based and model-based systems. A memory-based system computes the similarity of users/items to find the nearest n of users/items for recommendation. The neighbor-model CF is a very common method of this type that depends on human rating as the main feedback. The recommended items are identified by computing the similarities among users. By contrast, a model-based system learns the users' behavioral patterns to recommend items to them by dividing the dataset into two parts (training and testing) and applying data mining methods or machine learning techniques. Recently, several methods have been executed in this type such as clustering and Matrix Factorization (MF) (Ma et al., 2009). MF has become a very prevalent method in RSs since it yields highly accurate results and can reduce the user-item matrix to a small number of latent factors from which the most important attitudes can be extracted.

CB system is another type of RS. This type returns the recommendations by analyzing the content of textual information and finding the regularities. The essential difference between CF and CB systems is that CF system depends on user activity (rating and

browsing) to find users' preferences. Whereas, CB system relies on the profile of the users and items, then uses this information to compute the prediction (Yu et al., 2017).

However, RS faces some challenges that have their negative effects on accuracy. Moreover, the huge size of the data, which amounts to millions of items on some websites such as Netflix, in such data the sparse is increased inside it. In addition, users do not usually rate most of the items. Therefore, the recommendation accuracy is passively affected. This problem is called sparsity (Lam et al., 2008). Another issue is the cold-start problem. It refers to the problem of making predictions for a new customer or item. The RS cannot recommend items to new users because there is no history regarding their tastes. Likewise, the RS lacks information regarding the rate or score of a cold-start item (Kim and Kim, 2016). Cold-start is classified into two types: incomplete cold start (ICS) and complete cold start (CCS). ICS refers to the case in which the cold-start user/item has poor feedback (3 ratings or less). Whereas CCS refers to the case in which the user/item does not have any score and, hence, the RS lacks relevant items/users (Suryana et al., 2018).

The main elements in RS are users, items, and rating values. These elements represent a user-item matrix. The traditional methods utilize these elements as basic knowledge to recommend preference items to users (Yang et al., 2012; Leskovec, et al., 2014). However, the results are often inaccurate since the growth in the amount of data worsens the problems discussed above. In response, researchers tend to use social information in RS since this information has provided an additional source of data (Sun et al., 2015; Dang and Viennet, 2013; Si and Jin, 2003; Unger et al., 2016).

Recently, the use of social networks to enhance the recommendation accuracy has attracted many researchers in recommendation systems. They applied explicit relations directly (Sun et al., 2015; Ma et al., 2009; Yu et al., 2017; Dillon, 1983; Newman, 2001). The main factor of the social network information is explicit friends. The recommendation system employs the preferences of the user's friends to explore new items. However, the sparsity and accuracy problems are still encountered. Previous studies assume that connected users have almost the same references whereas unconnected users most probably have different tastes. In practice, users have similar preferences even with users who are far away in the network. This type of relation is called implicit relation (Newman, 2001; Ma et al., 2011). Implicit relations increase

the prediction accuracy by alleviating the sparsity problem. To effectively exploit the social network information, two types of friendship (explicit and implicit friends) should be exploited.

In the social information, users or items can be partitioned into numerous communities or clusters. A community means a group of entities that are close to each other compared to other groups. In RS, user's interests are varied, thus, a user belongs to many communities concurrently. For instance, a user may at the same time belong to many clusters such as college, school, friends, and family. Such communities (clusters) are called overlap communities. Overlapping community detection can simply recognize a user's interest in many communities. Many algorithms adopted Overlapping community in the social network (Lancichinetti et al., 2009; Zhi-Xiao et al., 2016).

Up to now, many approaches exploiting CFs have been applied to predict items to users. The most common methods are Matrix Factorization (MF) and clustering. Several methods have been employed in the MF technique such as Singular Value Decomposition (SVD), SVD++, Probabilistic Matrix Factorization (PMF), and non-negative matrix factorization (NMF). Furthermore, prediction can be made by clustering techniques (Tsai and Hung, 2012). A clustering process means distributing users into sets of groups/clusters according to the distance among them. An RS can exploit this technique by recommending items that were rated by users of a particular cluster to which the targeted user belongs. Various clustering approaches have been adopted in the literature such as k-means (Wasid and Ali, 2018), Partitioning Around Medoid (PAM) algorithm (Hussein et al., 2016), and Fuzzy C-means (Koochi and Kiani, 2016). In this research, MF and clustering techniques are applied to enhance the predict accuracy, namely, SVD++, PAM, and novel PAM overlapping methods. The details of these methods can be seen in chapter two.

The aim of this research is to promote the prediction accuracy by utilizing the Singular Value Decomposition technique that exploits Explicit and Implicit relationships to reduce the sparsity problem. Another aim is to alleviate the cold-start problem by employing clustering and overlapping community detection techniques. AS for the datasets, they include Last.fm which is released in HetRec workshop (Cantador et al., 2011), FilmTrust gathered by (Guo et al., 2013) and Ciao collected by (Tang et al.,

2012). Mean absolute error (MAE) and root mean square error (RMSE) are employed to measure the accuracy of the prediction. Furthermore, precision, recall, and f-measure are used to evaluate the accuracy of top-n recommended items.

1.2 Problem Background

The goal of recommendation systems (RS) is to provide items of interest to users. Many studies have been conducted on realizing this objective. However, one of the biggest challenges is accuracy. The main causes of low accuracy are sparsity and cold-start. As the data size grows rapidly, users cannot rate most of the items, which results in data sparsity. Cold-start is an additional recommendation system problem, in which the user does not rate items and, hence, the system cannot distinguish the user's preferences. Both problems negatively affect the performance of recommendation system. To mitigate these problems, many studies have been proposed on all RS types.

A traditional recommendation system involves users, items, and relations between users and items. A user-item rating matrix is used as a source of knowledge for the RS (Yang et al., 2012). Some RSs methods employ users' ratings as the only source of information and they do not consider any additional information. Most of these methods depend on expert ratings, user-generated feedback, and crowdsourcing (popularity) to produce a list of recommended items for each particular user (Son 2016; Ku et al., 2016; Wei et al., 2017; Gupta et al., 2017; Zhang and Piramuthu, 2018). Recently, in addition to rating information, modern RSs have used many other information sources such as social information, forums, and social bookmarking (Dang and Viennet, 2013). The modern Web 2.0 provides an opportunity for users to establish their relationships to create a social network. Thus, RSs can utilize these relationships as an extra information source to predict items (Pham et al., 2011). In practice, people are influenced by their friends. For example, when someone decides whether to buy, watch or listen to an item, they will check the items chosen by their friends. Therefore, social network information is useful in RSs. Social networks are satisfactory information sources for tackling RS problems. Hence, many researchers have utilized social network information to enhance the performance of RSs (Unger et al., 2016; Sun et al., 2015; Wang et al., 2015). In a social network, relations can be classified into two kinds: friend relationships and trust relationships.

Friend relationships, which are represented as an undirected graph, are mutual interactions between users in a social network. RSs permit customers to establish friend relationships. By contrast, trust relationships (which are represented as a directed graph), are one-side relations. For example, user 1 trusts user 2 but user 2 does not necessarily trust user 1. In a trust relationship, some RSs allow users to give trust values to other users.

Various related works incorporate a social network into MF to improve the RS performance. (Unger et al., 2016) proposed a regularization-based method that utilized MF with social information as an additional factor. They incorporated two social regularizations: friendship relations and the correlation between users and items. (Guo et al., 2016) proposed a method that used transition in the trust relationship to find new trustees, which can improve the performance of the RS. Other researchers such as (Xie et al., 2015) used social network information in a typical way, but they enhanced the accuracy of the prediction by increasing the number of learning loop.

Two social recommendation methods are proposed by (Ma et al., 2011): average-based and individual-based regularization (SR1 and SR2). In this study, both kinds of regularization are incorporated into the matrix factorization to compute the prediction of a missing rating. In both methods, the authors used explicit relations and integrated them into the matrix factorization method. The results revealed that the SR model is more accurate than the state-of-the-art models, such as User-Mean, Item-Mean, and Non-negative Matrix Factorization (NMF). (Reafee et al., 2016) proposed a new model (EISR) that exploited both implicit and explicit relationships to enhance the recommendation performance. The authors used the link prediction (LP) technique to discover implicit relations by computing the similarity between the user and the new link and integrating both explicit and implicit relations into Probabilistic Matrix Factorization. The Link Prediction technique is used to discover new relationships from the explicit relations. LP indicates the assignment of predicting new edges for the missing edges, where it uses many techniques to portend these missing edges. LP is utilized in many social network datasets such as scientific co-authorship networks (Newman, 2001) and e-commerce like RS (Chen et al., 2005; Liu and Kou, 2007). (Reafee et al., 2016) They used Resource Allocation Index (RAI) as a similarity technique of LP to find implicit relationships. However, RAI just collects the

neighbors of one-step and ignore multi neighbor's steps. Therefore, it misses some resources and it leads to missing substantial information. In this study, LP is employed but with the multi-steps resource allocation (MSRA) similarity technique. The prime reason for utilizing LP is its ability to find the network structure hidden relations and that it does not require any additional information.

Community-based RS is a method proposed by (Dang and Viennet, 2013). This method included two-steps. In the first step, they applied the clustering method to extract the users' communities. In the second step, they used the rating values of the users in a particular community to recommend items to other users in the same community. The method achieved high performance comparing with traditional RS that exploit users' rating only. (Lam et al., 2008) proposed method using social information to solve the cold-start problem by depending on the cold-start users' friend relationships of, and using these relationships to predict items for them. Moreover, (Yang et al., 2012) proposed a method based on the notion of the model of graph theory. They used a semi-supervised probabilistic model to recognize the sparsity problem, where they integrated between the impact of the user's friends and user preference. In another related work (O'Doherty et al., 2012), a new model was produced that utilizes trust relationships between pairs of actor users depending on the structural information of the bipartite graph. They employed the shared items to compute the degree of trust between pairs. The clear limitation of this model is using the classical collaborative filtering approach. Thus, the results were not accurate. (Jamali and Ester, 2010; Hussein et al., 2016) proposed a method that used community detection to determine the influential users to decrease the cold-start problem.

In a social network, users often belong to multiple interest communities to create an overlapping community. An overlapping community is a kind of community structure. Clustering users into overlapping communities contributes to alleviating not only the sparsity problem, but the cold-start problem as well via recommending items to users from different communities (Fortunato, 2010; Psorakis et al., 2011). Many algorithms and methods have been proposed and implemented to recognize the communities. Girvan and Newman (GN) (Girvan and Newman, 2002) proposed the first overlapping algorithm, in which the edges between the communities are removed from the original

graph. (Chen and Yuan, 2006) enhanced the GN algorithm by exploiting the weighted users in the graph to determine the functional modules.

Having reviewed the research that has been conducted in previous studies, it is concluded that the traditional methods still lack sufficient prediction accuracy. Common users exploited as much information as possible to enhance accuracy (Shi et al., 2020). In this approach, the author did not utilize the social information which has an obvious impact on RSs. In addition, (Ma et al., 2011) proposed two models SR1 and SR2. Both models made use of explicit friends and neglected implicit friends. (Zhang et al., 2013) proposed a model that exploited trust relations. However, the authors used explicit trust relations, which are limited and don't provide enough information. In (Reafee et al., 2016), the authors proposed an explicit and implicit social regularization (EISR) model. The implicit relations are extracted using the resource allocation index. However, the EISR model utilized the explicit social relations and the first step degree of the implicit relations. Yet, it ignored the second-degree of implicit relations. Regarding the clustering techniques, many studies have achieved as mentioned above. The most recent clustering study is (Hussein et al., 2016). The authors adopted both explicit and implicit social relations, and then applied the clustering technique to find the influences nodes in each cluster. They finally ranked the top-n influence nodes to serve the cold-start-users. The authors concluded implicit relations by using second-degree relations trust. However, they assumed that all relations between each pair of users have similar values which negatively affect the prediction accuracy.

In this research, four models are proposed to reduce RS problems. The Singular Value Decomposition (SVD) and SVD++ methods are adopted to produce the first two models MSRA-SVD and MSRA-SVD++ respectively. These models can exploit the social network (explicit and implicit relationships) and user-item matrix to predict values for the missing rating to reduce the sparsity problem. SVD is a low dimensional factor that plays a vital role in the RS since it allows one to find latent features between users and items. The main intuition of the SVD is the ability to discover latent features by constructing a user-feature matrix and item-feature matrix from ratings feedback (explicit feedback). It then uses the small part of this data and aborts the other to predict new items. (Sarwar et al., 2000) proposed the SVD method and they compared the

performance of his model with the CF method. (Badrul et al., 2002) proposed the SVD-base and they made the RS extremely scalable. Another model was proposed by (Zhang et al., 2005). They simplified the cost of the computation in SVD by keeping the accuracy of the recommendation stable. (Reddy and Adilakshmi, 2014) incorporated SVD with CF. They reduced the dimensionality and used the Euclidian distance to compute the similarity to find the user's neighbor. (Shi et al., 2014) devised a new model by dividing the users into groups and then recommending items to the users of each group. However, the SVD method suffers from the "cold-start" problem for new users or items in the absence of observed ratings (Dai et al., 2019). For example, if a user u in vector U (latent factor of users) and an item i in vector V (latent vector in items) have missing values, the algorithm cannot predict items. The SVD++ method is expanded of SVD. This method uses implicit as well as explicit feedback. The implicit feedback indicates purchasing or browsing an item, which reflects a user's preferences.

Additionally, social networks are exploited in different techniques such as clustering and overlapping to alleviate the cold-start problem. The users are clustered according to the distance between each user and the center of a cluster. The distances are computed by utilizing users' social relations and their confidence values. The PAM clustering algorithm is employed to produce the third model of this research. Finally, a novel overlapping PAM algorithm is proposed to produce the fourth model.

In all proposed models, the Stochastic Gradient Descent (SGD) optimization method is utilized. It is one of the most prevalent methods in the machine learning techniques employed to minimize the regularization error during the training stage. SGD has adopted in many applications such as image classification, image recognition, and recommendation systems.

1.3 Problem Statements

Having understood the studies conducted in the previous researches, it can be inferred that the traditional methods still suffer from low prediction accuracy. This research aims to enhance prediction accuracy. Therefore, the proposed model addressed the following Research Questions (RQ):

- RQ1 How can the division of the dataset into more than one part can increase the accuracy of the recommendation system?
- RQ2 How can social relations and explicit rating feedback information be incorporated into the SVD method?
- RQ3 How can social information and explicit/implicit rating feedback information be incorporated into the SVD++ method?
- RQ4 How can user clustering based on social relations and confidence values can be created using the PAM algorithm?
- RQ5 How can detecting users' communities who belong to several clusters be established using a novel proposed overlapping method?

1.4 Objectives of the Research

This research aims to produce a new model that can increase the prediction accuracy. To realize this goal, the MF technique is exploited, namely, SVD and SVD++ with valuable sources of information. Subsequently, social relations and confidence values are incorporated into the SVD and SVD++ methods. As mentioned earlier in the previous section, four models are applied to cope with the aforementioned problems. To accomplish the aim of this study, the following objectives have been specified.

- 1 Due to such challenges as cold start and sparsity, accuracy is negatively affected. Therefore, dividing a dataset into several parts and producing a separate model for each part can increase the accuracy of the RS.
- 2 To develop rating explicit feedback information, explicit social relations, and implicit social relations depending on the MF technique (SVD) used to reduce data sparsity.
- 3 To develop explicit and implicit rating feedback information, explicit social relations, and implicit social relations relying on the MF technique (SVD++) applied to alleviate data sparsity.
- 4 To evolve a clustering approach that can detect users' communities more accurately using the PAM clustering algorithm to decrease the cold-start problem.
- 5 To improve an overlapping method that can detect users who belong to different communities to alleviate the cold-start problem.

1.5 Scope of the Study

To achieve the aforementioned goal of this research, some aspects are determined. Thus, the scope of this research is as follows:

- 1- The dataset that should be used in this proposal includes social information and rating information. Hence, the datasets are Last.FM, FilmTrust, and Ciao, which are considered standard datasets that contain social information with the user-item matrix.
- 2- The challenges that are going to be solved are cold-start and sparsity.
- 3- The evaluation metrics are MAE, RMSE, precision, recall and F-measure, which are used to check the performance of the proposed models
- 4- Gephi software is utilized to show the social information in explicit and implicit relations.
- 5- The Matlab programming language is used to perform the program of the proposed models.

1.6 Significance of The Study

RSs aim to supply users with their items of interest. Therefore, in this study, the prediction accuracy is improved by integrating social relations and rating information into the matrix factorization methods (SVD, SVD++). Social relations have a vital role to boost the prediction process. Thus, this research considers this point and utilizes explicit and implicit social relationships. Regarding implicit relationships, the link prediction method is applied to extract hidden relations in the social network. Although many studies have been conducted to enhance RSs, they still lack sufficient prediction accuracy. Therefore, this research aims to overcome these limitations and improve the prediction accuracy. To achieve this goal, the dataset is divided into two parts: users with accepted rating history and cold-start users. For the first part, MSRA-SVD++ and MSRA-SVD models are proposed to exploit the social information which involves explicit and implicit relationships, as well as the rating information which also includes explicit and implicit information. For the second part, the users are distributed into sets of clusters according to their relationships. The prediction for the cold-start user is computed by taking the items rated in this community only. Finally, we believe that users may have various interests. For example, a user tends to watch different kinds of

movies such as action and comedy, consequently; a user may belong to several communities at the same time. Therefore, overlapping community detection is adopted to simulate this fact.

1.7 Contributions of the Study

Several contributions are made in this study. These contributions are illustrated as follows:

- i- Enhancing the prediction accuracy of a recommendation system for non-cold-start users by integrating three sources of information: explicit social relations, multi-steps resource allocation method to extract implicit social relations, and users' ratings into singular value decomposition (SVD) method to produce the MSRA-SVD model.
- ii- Enhancing the prediction accuracy of a recommendation system for non-cold-start users by integrated four sources of information: explicit social relations, multi-steps resource allocation method to extract implicit social relations, explicit users' ratings, and implicit users' ratings into singular value decomposition plus (SVD++) method to present the MSRA-SVD++ model.
- iii-Enhancing the prediction accuracy of a recommendation system for cold-start users by applying clustering community detection techniques (using PAM algorithm) to group users on the basis of their distances and the SVD++ method to predict items.
- iv-Enhancing the prediction accuracy of a recommendation system for cold-start users by applying overlapping community detection techniques (using a novel overlapping algorithm) to group users on basis of their distances and the SVD++ method to predict items.

1.8 Thesis Outline

The thesis is organized in eight chapters as follow:

Chapter 1 includes an introduction of the topic. The problems are highlighted with the research questions. The objectives and significance of the study are illustrated.

Chapter 2 is devoted to the literature review. In this chapter, the approaches and challenges of the recommendation system are addressed. Moreover, similarity methods, evaluation metrics and social relations are explained as well as their relation with the RSs. Finally, some of the common methods such as SVD, SVD++ and PAM algorithm are clarified.

Chapter 3 Research Methodology: This chapter exhibits all the practical theoretical issues and data required to conduct the research models. An overview is made of the four models of this research: MSRA-SVD, MSRA-SVD++, clustering community detection and overlapping community detection. Finally, the parameters setting and values are illustrated as well.

Chapter 4 The MSRA-SVD Model: This model is utilized to mitigate the data sparsity problem by integrating explicit social relations, implicit social relations, and user-item matrix into the SVD method.

Chapter 5 The MSRA-SVD++ Model: This model is used to tackle the data sparsity problem and overcome the limitation of the MSRA-SVD model.

Chapter 6 A Clustering Community Detection Model: In this model, the cold-start problem is alleviated by grouping the users into a set of clusters using the PAM algorithm. Afterward, the prediction process is achieved via the SVD++ method.

Chapter 7 An Overlapping Community Detection Model: In this model, the cold-start problem is defeated using a novel overlapping algorithm. Furthermore, this model is utilized to overcome the limitation of the previous clustering technique.

Chapter 8 Conclusions and Future Work: This chapter presents the finding of the study. The limitations and the expanded view for future work are illustrated as well.

2. LITERATURE REVIEW

2.1 Overview of Recommender System

With the expanded use of Web 2.0, RSs have spread in numerous applications, such as book stores, webpages, interested places, movies, clothes, accessories, and so forth. Recommendation systems have been independently researched since the mid-1990s. Many studies have been performed in this area and there are many issues that demand implementations to improve the quality of recommendation. RSs rely on explicit feedback information that can be acquired through observation of a user's behavior (Ma et al., 2011) or by utilizing users' ratings of data sources such that users can determine their preferences by rating items. Moreover, user profiles and item profiles consider data sources; these kinds of data sources are called explicit information (Adomavicius and Tuzhilin, 2005). A user profile includes personal information about the user, such as age, gender, and level of education. Item profiles involve their type, year of production, and director. Tables 2.1, 2.2, and 2.3 present users' prediction matrices, users' profiles, and items profiles, respectively.

Table 2.1. Examples of the user-item rating matrix.

	Item1	Item2	Item3	Item4	Item5
User1	5	3	?	1	?
User2	4	2	5	2	?
User3	3	?	?	2	4

Table 2.2. Examples of the users' profile.

	Age	Gender	Educational level
User1	21	F	Bsc
User2	25	M	Master
User3	30	M	Phd

Table 2.3. Examples of the Items' profile.

	Type	Name	Director	Year of the production
Movie1	Action	The dark night rises	Christopher Nolan	2012
Movie2	Comedy	Coming to America	John Landis	1988
Movie3	Romantic	Me Before You	Thea Sharrock	2015

Recommendation systems help users to find their preferences items by tracking users' behavior in the system. Many researchers have been produced various definitions of RS. For example, (Ricci et al., 2011) presented RS as “software tools and techniques providing suggestions for items to be of use to a user”. (Yu and Gao, 2018; Resnick and Varian, 1997) defined RS as “systems that obtain the opinions about items from a community of users and then use those opinions to guide other users within that community to those items that are interesting for them”. The following sections are presented the literature of RSs which involve related work of the RS, approaches, challenges, similarity measures, evaluation metrics, social network information, and datasets.

2.2 Feedback Information on the Recommender Systems

As in any system, recommender systems require inputs to implement the prediction process and then produce an output (items). These inputs are acquired by creating a user profile that should contain information related to user's preferences. The recommender system can apply many techniques to infer users' preferences implicitly, such as by monitoring users' behavior. Moreover, the recommender systems produce suggestions for users and ask them explicitly to present their preferences (Jannach et al., 2010; Ricci et al., 2011).

Several methods are utilized by recommender systems to collect user information. Such information is gathered explicitly through items evaluations. There are many of these that evaluate items including 5-star ratings, 1-10 ratings, or binary ratings (by clicking like or dislike). Another way to collect users' preferences is implicit information, which can be acquired by recording the time on a webpage page, the speed of mouse scrolling when a user browses a webpage, or the number of times that

a user listens to a certain song. Moreover, elements of a user profile, such as age, gender, hobbies, and/or education level, can be exploited by a recommender system. Another source of information may include content that has been shared or posted by the user on social media (Lee et al., 2010; Choi et al., 2012).

2.2.1 User-Item rating matrix

The main source of the information in recommendation systems is the user-item matrix, without these ratings, of which it becomes impossible to recommend items for users (Leskovec et al., 2014).

A rating expresses a user's preference for several items, which can be exploited by the RS model to predict new items for the user. The rating term refers to the preference of a user for a particular item. Many kinds of rating are in existence in an RS, such as 5-star ratings, like/dislike, and some systems collect ratings by asking users a number of questions and through their answers, the system can infer a rating value or user interest (Ricci et al., 2011).

Table 2.1 shows an example of the user-item matrix, where the rating type is a 5-star rating. The matrix involves users and items, with the rows indicating users (u1, u2, u3) and the columns referring to items (I1, I2, I3, I4, I5). As shown in the table there are some cells without ratings and other cells containing values between 1 and 5. Negative feedback about an item is indicating by 1 or 2, whereas the values 3, 4, 5 indicate positive feedback (Yang et al., 2012). The task of the recommendation system is to predict items for users in the empty cells.

2.3 Types of The Recommender Systems

In the modern e-commerce applications, data has grown rapidly; thus, it becomes very difficult to navigate in these applications to determine one's preferences. This problem is called information overload. The aim of recommender systems (RSs) is to overcome information overload and recommend items to users without browsing through every items. Several taxonomies have been produced to illustrate the RS and its types. The common taxonomy of RSs includes three main types: Collaborative Filtering (CF),

Content-Base (CB), and Hybrid system. Figure 2.1 depicts the types of RS. The following sections display RS types in further detail.

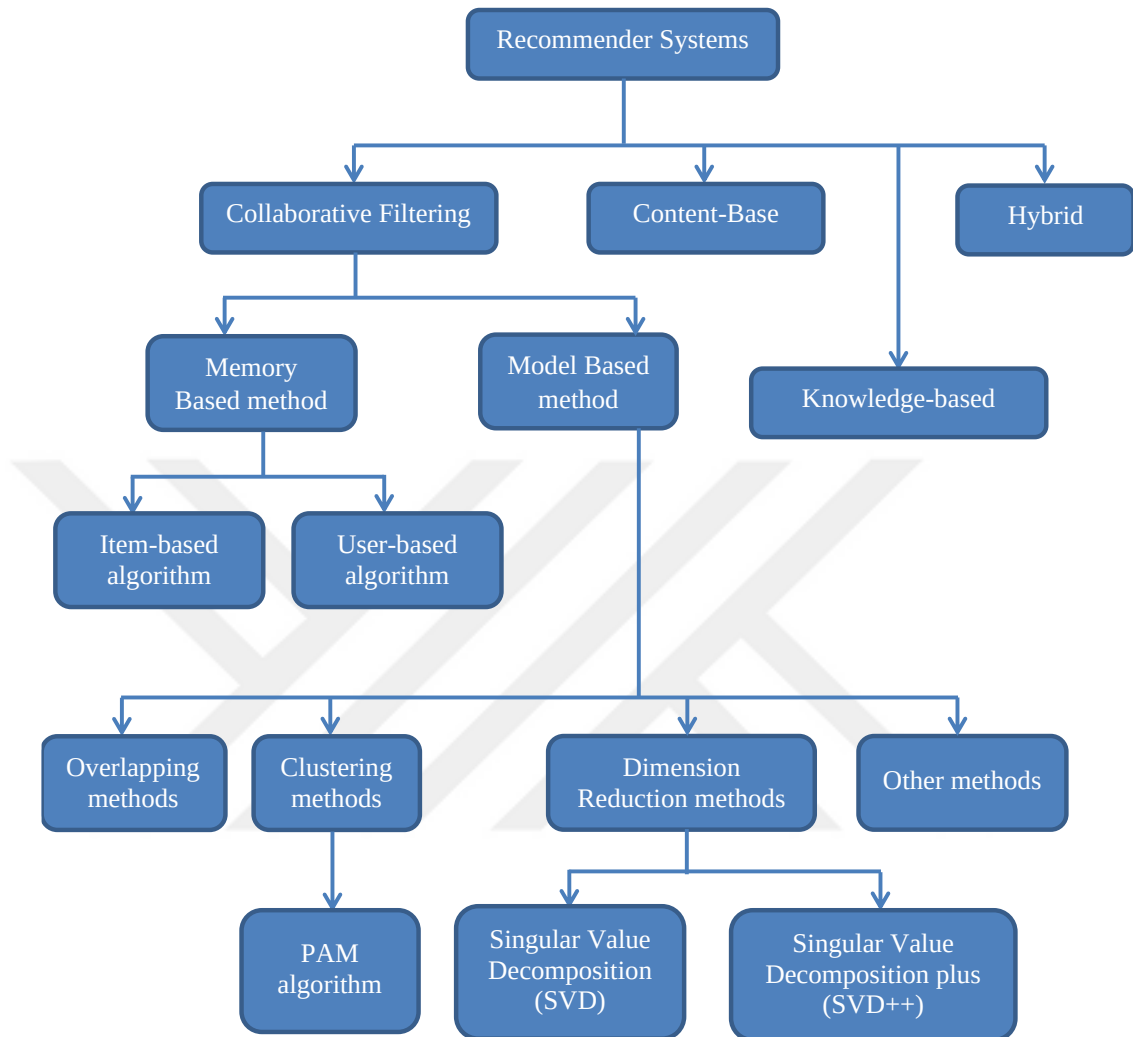


Figure 2.1. Types of Recommender systems.

2.3.1 Collaborative filtering approach

One of the most common types of recommendation systems is Collaborative Filtering (CF), which employs the user's historical preference where the system records the user's behavior while browsing a website. The system then predicts items for the user based on user's interests (Adomavicius and Tuzhilin, 2005). CF is utilized in many E-commerce websites such as Amazon (Linden et al., 2003), TiVo, and Yahoo! (Koren et al., 2008). Moreover, CF is considered a vital type of RS approach and is still active in several applications domains, such as E-learning, E-resources, and E-tourism (Lu et al., 2015).

CF is divided into two categories: Model-base and Memory base (Bergner et al., 2012). Model-base utilizes the user-item matrix to employ it in machine learning techniques to train the model, followed by predicting items for users. While Memory-base exploits users' ratings and computes the neighbor of the active user to predict new items (Al Falahi et al., 2012).

A memory base can predict items ideally when the user-item matrix has high density. Nevertheless, because of increases in the size of data on a daily basis, it is very difficult to produce data with less sparsity. Therefore, the accuracy of predictions is negatively affected. On the other hand, the model can overcome the shortcoming of the memory base CF because of its capability to reduce the effect of the sparsity problem thereby producing more accurate results (Basu et al., 1998).

2.3.1.1 Memory-based approach

In this approach, the core process to predict items for users is similarity. This approach is divided into two types: the user-based algorithm and the item-based algorithm (Yang et al., 2012; Sarwar et al., 2001). The user-based algorithm computes the similarity between each pair of users to find the user's neighbors. The item-based algorithm applies the same procedure, the similarity is calculated between each pair of items to find the item's neighbors. The main source of information is a user-item matrix. The entire process of the memory-based approach involves three steps:

- i. Calculating the similarity among users or items.
- ii. Determining the neighbors and generating top-N computed items using k-nearest neighbors.
- iii. Creating recommendations.

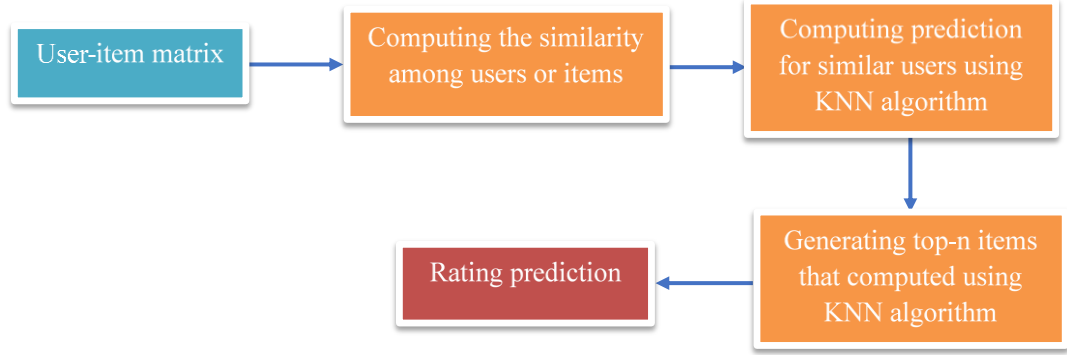


Figure 2.2. The architecture of the memory-based approach.

- Computing the similarity

As shown in the Figure above, the first step of memory-based is computing the similarity among users or items. A user-user matrix is created, and then a certain similarity method is applied. For each pair of users, the common items between them are counted and the similarity is computed by applying the similarity formula. Pearson correlation coefficient and Cosine similarity are the most similarity methods used in recommender systems. The outcome of this step is the similarity value of each pair of users. Section 2.5 shows the similarity process in more detail.

- K-nearest neighbors (K-NN)

K-NN is one of the prevalent classification algorithms used in machine learning techniques. Additionally, it is used in the recommendation systems field, specifically in the memory-based approach (Bobadilla et al., 2011). After acquiring similarity values, the similarity values are ranked according to their similarity for the active user, starting from the closest users. Then, k-nearest neighbors are determined. Finally, top-N users are selected to compute a new rating value for the active user based on their ratings. K-NN algorithm is utilized in both the user-based and item-based approaches. In both algorithms, several prediction formulas can be applied as follows:

a- Weighted sum of others' ratings

$$p_{ik} = \bar{r} + \frac{\sum_{j \in n(i)} sim_{i,j} * (r_{i,j} - \bar{r}_j)}{\sum_{j \in n(i)} sim_{i,j}} \quad (2.1)$$

where, p_{ik} is the predicted rating for user i and item k , $\bar{r}$ refers to the mean rating for user i , $sim_{i,j}$ indicates the similarity between users i and j , and $n(i)$ is the neighborhood of user i .

b- Simple Weighted Average

This method does not use an average rating value of the user/item. The following formula computes the rating value for the item-based prediction:

$$p_{u,i} = \frac{\sum_{n \in N} (sim_{i,n} * r_{u,n})}{\sum_{n \in N} |sim_{i,n}|} \quad (2.2)$$

Where $sim_{i,n}$ is the similarity between items i and n , $r_{u,n}$ is the rating of the user u for item n ; and N is the total number of items.

However, the drawbacks of K-NN are its memory consumption and time. It takes a very long time especially for a large database, and it requires high-capacity memory to handle the number of users to compute similarity. Moreover, the similarity values are computed based on common items between users, and when there is sparse data, there will be very few common items. Subsequently, the values of the prediction will be unreliable. Accordingly, the model-based approach has been used to handle these limitations. The following subsection explains the model-based approach.

2.3.1.2 Model-based approach

In this approach, prediction occurs by extracting pattern behavior between users, while applying machine-learning algorithms. User ratings are trained in a particular technique in the training part and are then used to predict new items in the testing part. Machine-learning algorithms have succeeded in overcoming the flaws of memory-based and have produced more accurate results. Several algorithms have been able to apply the model-based, such as clustering (Xue et al., 2005), Bayesian models (Yang et al., 2012), aggregating regression (Demir et al., 2017) (Zhao and Guo 2017), probabilistic soft logic (Kouki et al., 2015; Bach et al., 2017), and dimension reduction methods (Hofmann, 2003). Singular Value Decomposition (SVD) and SVD plus (Sarwar et al., 2000) are used to extract any hidden information inside the user-item matrix.

i. Dimension reduction

Dimension reduction methods are used to enhance the quality and efficiency of the neighborhood-based approach. Sparsity is a common problem in RSs, such that users keep most items without ratings. As a result, similarity computation becomes difficult. In terms of the latent factor, dimension reduction produces low-dimension impersonation. Dimension reduction can assume important information of the data even though there exists little rating and unimportant data can be removed without sacrificing accuracy. The distance of two users can be calculated by their low-dimensional latent factor even when they have a small number of ratings.

Many researchers have applied dimension reduction methods in their studies including Singular Value Decomposition (SVD) (Golub and Reinsch, 1971), Latent Dirichlet Allocation (LDA) (Blei et al., 2003; Lee and Seung, 1999), and Probabilistic Matrix Factorization (PMF) (Mnih and Salakhutdinov, 2008). In this research, the SVD algorithm was utilized to extract the latent information of the user-item matrix.

Singular Value Decomposition SVD is a low dimensional factor that plays a vital role in the RS since it allows one to find latent features between users and items. The main intuition of the SVD is its ability to discover latent features by constructing the user-feature matrix. The item-feature matrix then uses the small part of the dataset that contains the rich information and aborts the other data, after which it predicates new items. [30], proposed the SVD method and compared the performance of his model with the CF method. (Badrul Sarwar et al., 2002) proposed an SVD-base model and made the RS extremely scalable. Another model was proposed by (Zhang et al., 2005), which simplified the cost of the computation in the SVD by keeping the accuracy of the recommendation stable. (Reddy and Adilakshmi, 2014) incorporated SVD with CF and reduced the dimensionality and used Euclidian distance to compute the similarity in order to find the user's neighbor. (Shi et al., 2014) proposed a new model by dividing users into groups and recommending items to the users of each group.

A particular matrix with m rows and n columns decomposes into three vectors U , S , and V . SVD constructs two orthogonal matrices $U_{m \times r}$, is right orthogonal, where m refers to the number of users and k nonzero eigenvalues. $V_{n \times r}$ is left orthogonal, where n refers to the number of items. $S_{r \times r}$ is a diagonal matrix, which has k nonzero values.

SVD can find the best low rank linear of the user-item matrix. Multiplication of the three vectors produces a new matrix similar to the original before the decomposition process. Figure 2.3 shows the decomposition process of the SVD method.

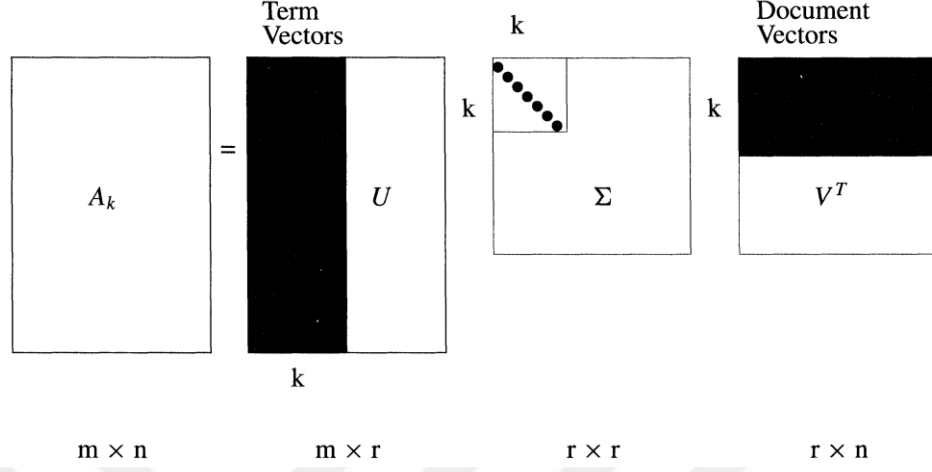


Figure 2.3. Decomposition process of SVD method.

The prediction is carried out by multiplying the user vector by the item vector followed by adding it to the baseline.

$$SVD(A) = U * S * V^T \quad (2.3)$$

$$\bar{r}_{ui} = b_{ui} + V_i^T U_u \quad (2.4)$$

where b_{ui} is the baseline predictor for the unknown items. The baseline predictor can be computed by Equation (2.5) (Lu et al., 2015).

$$b_{ui} = \mu + b_u + b_i \quad (2.5)$$

$$b_i = \frac{(\sum_{u \in R(i)} r_{u,i} - \mu)}{|R_i|} \quad (2.6)$$

$$b_u = \frac{(\sum_{u \in R(u)} r_{u,i} - \mu - b_i)}{|R_u|} \quad (2.7)$$

μ is the mean of global rating, b_u is to the user biased derived from μ , and b_i is the item bias that is derived from μ . (Ricci et al., 2015) proposed two regularization parameters (λ_1 and λ_2) which can be added to $|R_u|$ and $|R_i|$ to compute the user bias and item bias. Equations (2.6) and (2.7) have been adopted in this study and modified thus:

$$b_i = \frac{(\sum_{u \in R(i)} r_{u,i} - \mu)}{\lambda_1 + |R_i|} \quad (2.8)$$

$$b_u = \frac{(\sum_{i \in R(u)} r_{u,i} - \mu - b_i)}{\lambda_2 + |R_u|} \quad (2.9)$$

Singular Value Decomposition plus SVD++: Basically, the SVD++ method derived from SVD, which has utilized by many researchers (Jiao et al., 2019) and (Shi et al., 2020). However, the SVD method exploits the explicit information (votes or rating) that is available in the user-item matrix and ignores the implicit information, such as clicking, browsing, and purchasing (Jiao et al., 2019). SVD++ proposed by (Koren and Bell, 2015). For example, assume that the 2-D user-item matrix (M), the rows U indicates the users, and the columns V represent the items. The rating value that the users can rate the items with a value can be expressed as $r_{u,v}$.

The purpose of the RS is to predict values for the particular set of empty cells in the user-item matrix. In general, the dataset is huge and many of the users leave the items without a rating, thus RS faces data sparsity problem. However, SVD++ decomposes the user-item matrix into two matrices P and Q , which are orthogonal matrices to represent the users and items respectively. The decomposition process is to convert the high dimensional user-item matrix that contains the whole data into low dimensional data $P^{u,f}$ and $Q^{i,f}$ where u is the user number, I is the items number, and f refers to the dimension after reduction. In other words, SVD++ reduces the size of the P and Q matrices to a specific value and finds the best low dimensional data representation of the user-item matrix by removing the unimportant data.

Furthermore, SVD++ is an expanded method for SVD. The difference is that the implicit feedback is taken into consideration when it comes to SVD++, whereas the SVD method depends only on the users' rating (explicit feedback). Consequently, a Binary matrix is constructed to represent the implicit feedback, where the new matrix is separated into left and right orthogonal matrices. The formula depicts the prediction of the SVD++.

$$r_{u,i} = \mu + b_u + b_i + q^T \left(p_u + \frac{1}{\sqrt{|N_u|}} \sum_{j \in N_u} y_j \right) \quad (2.10)$$

Where μ is the average value of the whole dataset, b_u and b_i are the observed deviations from the mean value of user u and item I , respectively. $|N_u|$ is the number of items rated by user u , $|R_u|$ is the number of users that rated a specific item, $|R_i|$ is the number of items that were rated by a set of users, and y_j is the left orthogonal matrix of the implicit matrix.

ii. Clustering Models

Clustering or unsupervised learning is the assignment of items of similar properties into a group; hence, items in the same group differ from items in other groups. The purpose of clustering is to reduce searching times for particular items, so instead of searching in an entire dataset for an item, it becomes only necessary to check the general properties of each cluster and find a more similar cluster for a desired item. Clustering can be divided into two categories: hierarchical and partitioned (Ricci et al., 2011). Hierarchical algorithms utilize the tree concept to cluster items. Partitioned clustering algorithms compute the similarity the item features followed by setting close items together in one cluster.

Several studies have used clustering algorithms in RSs. A clustering technique for RSs is proposed by (Dang and Viennet 2013), and includes two-steps. In the first step, the clustering method was applied to extract the communities of users. In the second step, the rating values of users in the communities was used to recommend items to other users in the same community. The method achieved high performance in comparison with traditional RSs that exploit only user ratings. (Xue et al., 2005) used the K-means algorithm to allocate the neighborhood. They computed the distance between a user and the centers of the cluster to determine which cluster should contain the user. In addition, they utilized smooth technology to improve the clustering process, which is better than the K-NN CF approach. (Sarwar et al., 2002) proposed to expand the K-NN classifier, in which they employed K-means clustering twice to divide users into created clusters. The results were enhanced in terms of efficiency; however, in terms of accuracy, the results decreased to 5%. (Gose, 2005) proposed another method, but here they suggested to cluster items rather than users. In that study, the authors used the Pearson Correlation to compute similarity between items. They applied four different algorithms: robust clustering, average link hierarchal agglomerative, h-Metis,

and k-Metis. The accuracy deteriorated in comparison with the state of the art, while the efficiency was enriched. (Li and Kim, 2003) and (Ungar and Foster, 1998) produced a comparable idea by using K-means to resolve the probabilistic model interpretation of the RS.

Community detection produces a social network as a graph, users, products, or items. Networks are represented by vertices and edges of the graph, indicating the relationship among vertices. Community detection has become a complicated task since the size of the network and the interacting of the vertices have increased.

Many algorithms have been proposed in the literature and these algorithms are categorized into three types: similarity-based algorithms, optimization-based algorithms, and heuristic algorithms (Yang et al., 2010). Newman proposed the Fast Newman algorithm (Newman, 2004). The modularity function is utilized for global optimization. K-means and K-medoids are two common algorithms that have adopted partitioning clustering (Bedi and Sharma, 2016). (Bi et al., 2016)(Hussein et al., 2016) proposed a method using the PAM algorithm to determine how an influential user to decrease the cold-start problem. This study divided users into clusters in the community followed by finding the centrality of each cluster and then recommending popular items of each cluster to new users. However, the author here considered all the nodes (users) to have the same weight as the center of the cluster, which would affect the accuracy of the prediction as each node has a different weight depending on the closeness of the cluster center.

The following subsection presents Partitioning Around Medoid (PAM) algorithm, which uses the K-medoid technique.

Partitioning Around Medoid (PAM) Algorithm

The clustering process is gathering objects almost the same as each other in one community or cluster. K-means is the common example of the clustering and it's used in many applications. K-means applied Euclidean distance to compute the distance between the cluster's center and the objects in the cluster (Leskovec et al., 2014).

K-medoid is another technique of clustering and PAM is the essential algorithm in the K-medoids. PAM is achieved depending on k-means and shifting the medoids algorithm. Both K-means and PAM are implementing the clustering by decreasing the distance between objects in a particular cluster and the central object of the same cluster. However, k-means sensitive to the outliers because the distribution of data is corrupted if an object has extremely value. PAM algorithm overcame this limitation where it less sensitive to the outliers and large values than K-means (Jiawei and Micheline, 2006).

PAM algorithm firstly determines the k objects as centers (medoids) of clusters and the rest objects assigned to the closest clusters. During the execution of the PAM algorithm, the algorithm tries to find a new medoid by a random selection of non-medoid objects, and if the new cost smaller than the current medoid, the new selection will swap with the previous medoid. This process is repeated until reaching the minimum cost (Jiawei and Micheline, 2006).

iii. Overlapping techniques

Individuals interact with each other to create a social network. With the development of Web 2.0, online interactions among users have grown rapidly. Several applications such as Facebook and Twitter, have facilitated user interactions. Due to the increase in the number of interactions, recognizing these interactions has become difficult and complex as has the process of grouping similar individuals into a cluster community for detection. Community detection can be extracted using the characteristic properties of social networks (Bedi and Sharma, 2016). Community detection regardless of clustering or overlapping, is utilized in many applications, including pattern recognition, text mining, and recommendation systems. In real networks, users may belong to several communities, and accordingly, these communities may overlap. Figure 2.4 shows an example of the overlapping of two communities, such as when two nodes (5 and 6) overlap in the two communities.

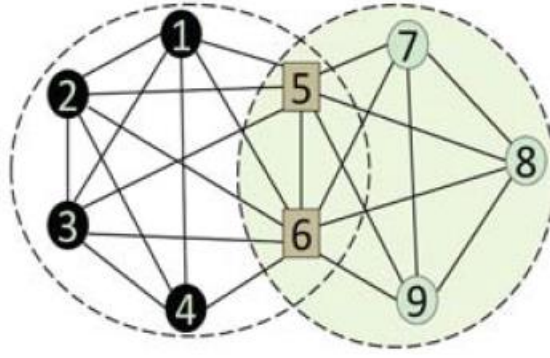


Figure 2.4. A simple overlapping of two communities.

The advantage of overlapping community detection is that there is provision for users' interests from different communities rather than a unique community. Recently, the overlapping technique has attracted many researchers due to its ability to handle the limitations of RSs, such as data sparsity and cold-start problems. Many studies have applied overlapping community detection. (Palla et al., 2005) proposed the first overlapping community detection, namely the algorithm Clique Percolation Method (CPM), which achieves good results, especially in complex networks. In more recent studies, local optimization methods have been proposed to detect overlapping communities (Lancichinetti et al., 2009). Local fitness maximization (LFM) algorithm proposed by (Lancichinetti et al., 2009; De et al., 2013), This algorithm exploits the advantage of overlapping and hierarchy communities. The authors of (Feng et al., 2015) enhanced the overlapping detection by adjusting links weights by adding pre-processing implementation.

Another effort uses an overlapping community detection method for time-weighted recommender systems. In this approach, the authors ignored users' similarities while determining community centers and instead used association rules (Feng et al., 2015). A hybrid method was proposed by (Rezaeimehr et al., 2018), wherein three main steps were implemented. First, missing predictions were computed based on similarity and reliability measures. Second, overlapping community detection was utilized based on the time-weighted approach. Finally, rule-mining was applied to suggest n-items.

In this research, overlapping community detection is exploited based on explicit and implicit trust relationships to overcome the cold-start problem.

2.3.2 Content-based approach

The Content-Based approach (CB) recommends items that depend on analyzing the content of textual information. This content could be documents, URLs, item descriptions, or a profile of a user's preferences (Pazzani, 1999). The textual content can observe users rating items in the past, so the similarity between these items may be considered to be textual content (Zhu et al., 2003), and the observed browsing features of the pages. Heuristic methods or classification algorithms are utilized in this type of RS to predict items (Pazzani and Billsus, 1997).

Unlike CF, which relies on the user-item matrix and exploits these ratings for a recommendation, CB techniques use features associated with the object they recommend. The limitation of CB, is that the features are not available and data descriptions are insufficient in number. Subsequently, the recommendation process is difficult to execute. In addition, CB utilizes another source of information that uses a user's profile, such as gender, education level, occupation, hobbies, and so forth (Krulwich, 1997). Figure 2.5 shows the details of the CB approach. Three main modules can be seen in this approach.

- i- Content analyzer: In this step, particular processing steps are applied to extract features from unstructured source information such as text. Then, it represents these features in an appropriate form.
- ii- Profile learner: The outcome of the content analyzer is exploited by this module. The representative data are collected and learned using a machine learning technique to create the user profile.
- iii-Filtering component: The user profile is utilized to recommend a set of relevant items by computing the similarity between the profile of the targeted user with the system's items.

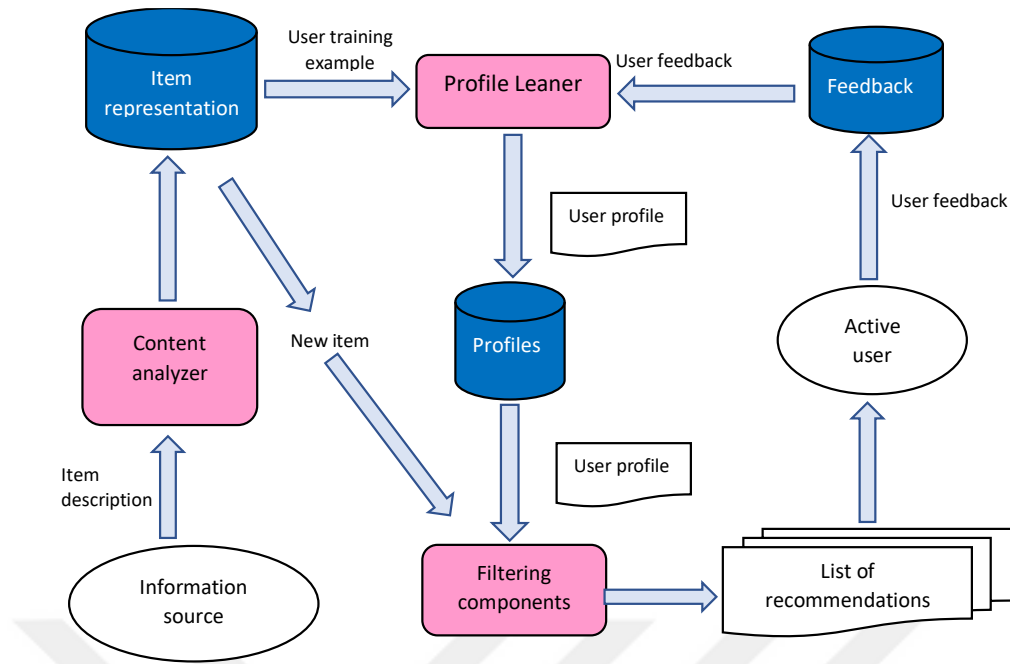


Figure 2.5. Content-Based approach architecture (Lops et al., 2011).

The CB approach uses two classes for recommendations: the first-class uses information retrieval methods (Lops et al., 2011; Semeraro et al., 2009). This class depends on the similarity measure to compute the similarity between a query and a set of unrated items in the system. The second-class utilizes machine learning methods, such as K-Nearest Neighbor (KNN) (Cover and Hart, 1967), Decision Trees (DT) (Quinlan, 1986), or Support Vector Machine (SVM) (Cristianini and Shawe-Taylor, 2000). Additionally, the CB approach has been exploited in many applications, such as AVATAR (Fernández et al., 2006) and PTV (Smyth and Cotter, 2000), Foaling the Music system (Celma, 2006), and Music FX (McCarthy and Anagnost, 1998). Moreover, the CB approach has been used in the book recommendation domain. (Mooney and Roy, 2000) proposed a method called Learning Intelligent Book Recommending Agent (LIBRA), in which many features are extracted to create the book profile such as book titles, authors, reviewers, and synopses. All these features are exploited by the Naïve Bayes classifier.

2.3.3 Knowledge-based system

A knowledge-based (KB) system exploits the knowledge of users to offer a set of items to those users. KB usually utilizes the relationships between a user's requirement and available recommendations. In other words, KB determines how to meet a user's needs

with a particular item. One of the widespread expressions of KB is case-based. In this expression, the similarity between a user's profile or query is employed to generate a list of items for the targeted user (Lu et al., 2015).

2.3.4 Hybrid Approach

As mentioned previously, both CF and CB have a number of limitations. Thus, a new approach is applied by incorporating both CF and CB to overcome the mentioned limitations known as the Hybrid Recommender System. This approach takes advantage of CF and CB to combine their features in one model to recommend items. Many studies have employed a hybrid approach and each study has a different scenario. For example, the Naïve Bayesian classification algorithm uses Content-Based information to predict missing items in the user-item matrix, with note keeping values of the original user-item matrix without touching (combining the output rating by applying CB with the source of a user-item matrix). This is followed by applying the weighted Pearson Correlation, which gives higher weights for common items (items rated by most users) and awards a higher rate to the active user (Melville et al., 2002) Table 2.4 shows details of this scenario. As we have seen in Table 2.4 how the Hybrid approach reduced the sparsity problem by filling the missing rated using CB information.

In addition, the hybrid approach can alleviate the cold-start problem and when the user does not have a sufficient rating, the approach can exploit the user's profile and recommend items to the user. The results of the hybrid approach are better than the results of applying pure CF or pure CB in terms of MAE and RMSE (Su et al., 2007). (Bi et al., 2016) proposed a group-specific latent factor model by utilizing missingness-related characteristics to accommodate new users or items without any observed ratings. Figure 2.6 shows the generic framework of the hybrid system.

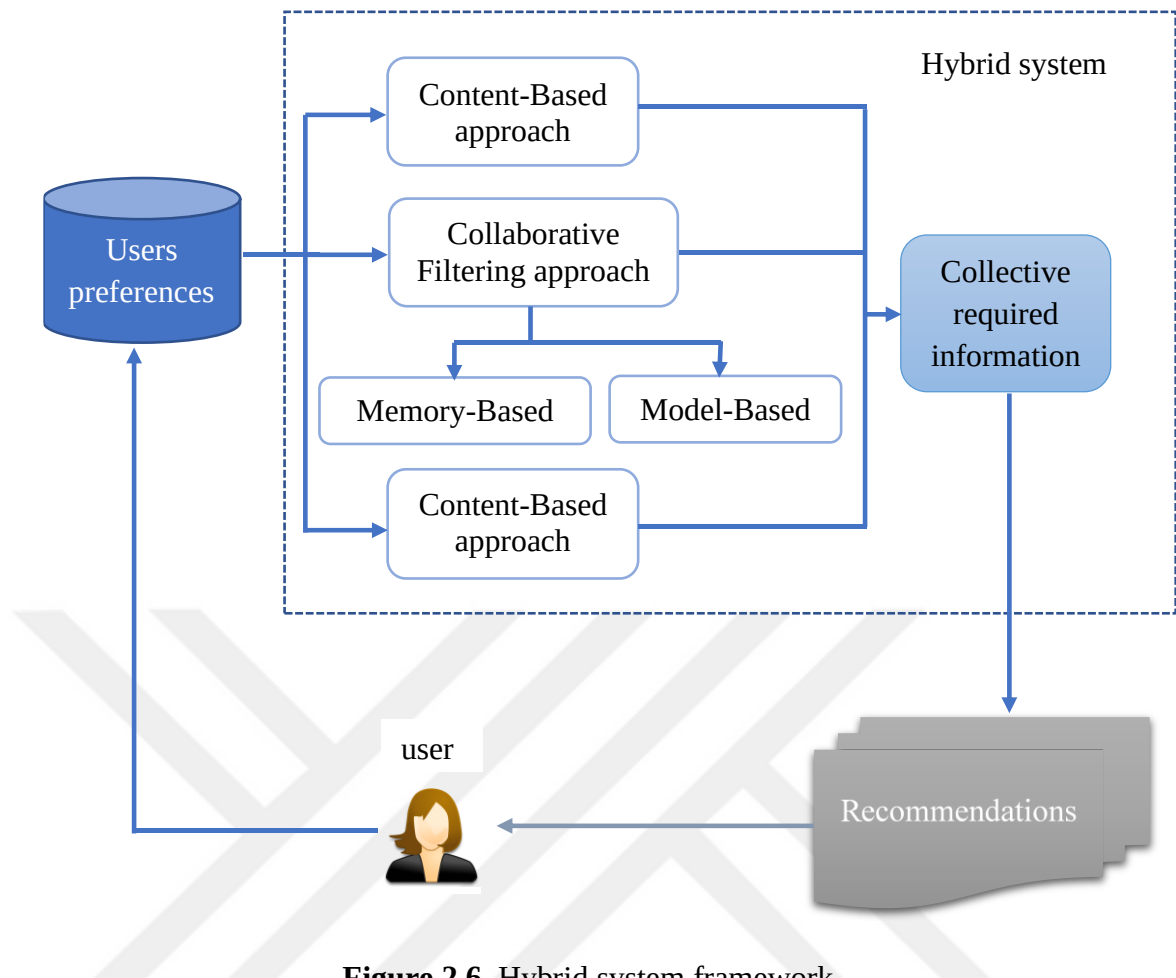


Figure 2.6. Hybrid system framework.

Table 2.4. Incorporated CB and CF (a) content data and original user-item matrix (b) pseudo rating filled by CB technique (c) predictions using weighted Pearson on pseudo rating data.

(a)

Content information					Rating matrix				
	Age	Sex	Career	ZIP	I1	I2	I3	I4	I5
U1	32	F	Writer	22904			4		
U2	27	M	Student	10022	2		4	3	
U3	24	M	Engineer	60402		1			
U4	50	F	Other	60804		3	3	3	3
U5	28	M	Educator	85251	1				

(b)

Pseudo rating data					
	I1	I2	I3	I4	I5
U1	2	3	4	3	2
U2	2	2	4	3	2
U3	3	1	3	4	3
U4	3	3	3	3	3
U5	1	2	4	1	2

(c)

Pearson-CF Prediction					
	I1	I2	I3	I4	I5
U1	2	3	4	3	2
U2	3	4	2	2	3
U3	3	3	2	3	3
U4	3	3	3	3	3
U5	1	3	1	2	2

2.4 Challenges of Recommendations Systems

Although E-commerce has been used in many applications by large online companies, the increasing amount of data causes real challenges for developers and researchers. In this section, several of the common challenges of CF are explained.

2.4.1. Data sparsity

Due to the large amount of online data that reaches many thousands of users and items, most users typically leave a high proportion of items without a rating, which is a problem known as sparsity. When the system has a huge number of items and the number of user ratings is very small, the recommender system is unable to recommend items that meet users' preferences. Moreover, finding the similarity between similar users' tastes does not reflect users' behavior if they rate a small number of items and none of these items is similar (Su and Khoshgoftaar, 2009). Subsequently, the sparsity problem affects prediction performance negatively. Several studies have adopted dimension reduction methods such as SVD, SVD++, and PMF to alleviate the sparsity problem (Al-Sabaawi et al., 2020; Yu et al., 2017). Another approach uses demographic profiles, in which similarities between two users can be computed if they have similar profiles, such as age, gender, education, area code, etc. (Pazzani, 1999).

	T1	T2	T3	T4	T5	T6	T7	T8
U1	4		3	4		2		5
U2		3	4	2	5			
U3		4		4		2	2	4
U4	3		5		2	4		
U5	2	5		2		4		4

Figure 2.7. Sample of sparsity problem.

2.4.2 Cold-start

One of the most common challenges is the cold-start problem which includes cold-start users and cold-start items. A cold-start item is a new item that has been introduced to a system. It is difficult to predict an item as long as the item has not been rated by

users, or the item will be at the end of the prediction list due to there being no feedback given on this item (as it is still new). However, to handle the cold-start item partially, staff members system rates this item. Similarly, the cold-start user problem occurs when a new user registers in the system or when a user has few ratings (less than a threshold) (Son, 2016). A cold-start user is divided into two types: incomplete cold start (ICS) and complete cold start (CCS). ICS refers to the scenario in which the cold-start user/item has poor feedback (3 ratings or less), whereas CCS refers to the scenario in which the user/item does not have any score and, hence, the RS lacks relevant items/users (Suryana et al., 2018). Figure 2.8 depicts the cold-start problem.

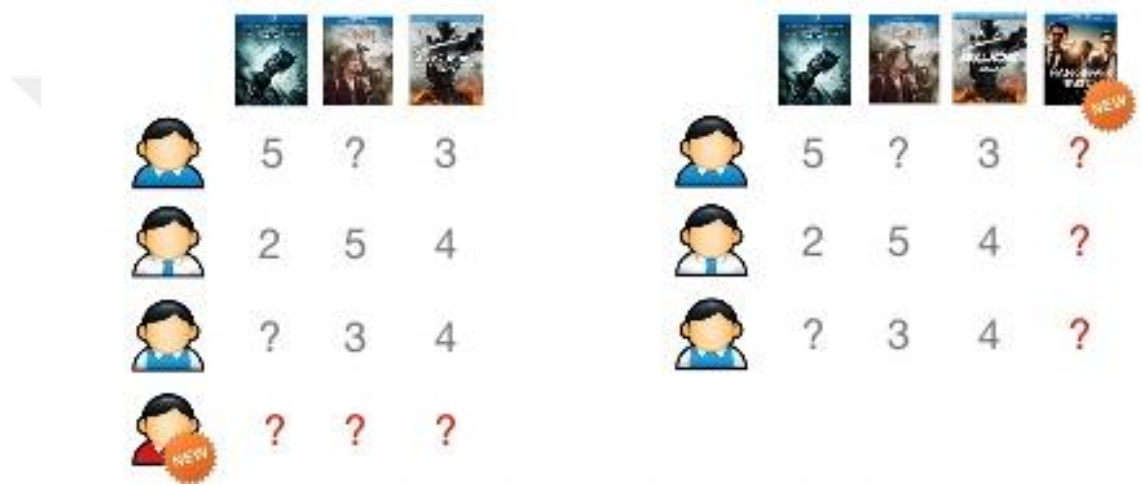


Figure 2.8. Cold-start problem.

2.4.3 Scalability

Scalability indicates computation complexity as long as the numbers of users and items grow at an extreme rate. Traditional CF faces a real scalability challenge, such that computation resources will become insufficient. For instance, for millions of users (M) and items (N), the computation complexity of CF is approximately $O(M+N)$ to $O(M^2N)$, which is too large (Papagelis et al., 2005). Additionally, ideal online systems should respond to a user request immediately regardless of users' activities, which requires high scalability of the CF approach. Furthermore, all computations are achieved at the prediction time, thus the scalability issue becomes the main challenge of the memory-based CF. However, an item-based approach is more robust than a user-based approach in terms of scalability, where computations of the item-based approach

occur offline. On the other hand, online predictions are required for the user-based approach (Wang et al., 2006).

Several studies have been proposed to solve the scalability problem such as (George and Merugu, 2005; Xue et al., 2005; Altingovde et al., 2013). In addition, dimension reduction methods are utilized to overcome the scalability problem (Sarwar et al., 2002), in which the authors applied the SVD method to compute SVD decomposition via exiting users.

2.4.4 Synonymy

Synonymy is a problem of items that have different names. Most RS cannot distinguish such a problem and deal with these items differently. For example, two movies: “children’s movie” and “children’s film” are the same product, but an RS works with them differently. The prevalence of synonyms affects accuracy prediction negatively (Su and Khoshgoftaar, 2009).

Some studies have proposed an intelligence technique that adopts automatic term expansion. The limitation of this technique is that the added terms may lead to different meanings, and subsequently regression of the prediction results (Jones, 1972). Another study was proposed latent semantic index by (Deerwester et al., 1990), In this technique the similar items or products are placed near to one another. However, the latent semantic index provides a partial solution to the synonymy problem, where the main idea of this method is that most words have several meanings (Patel et al., 2017).

2.4.5 Privacy preserving

Privacy pertains to the personal information collected from users during registration on a system. Perfect systems should keep the personal information of their customers secure and avoid violation of users’ privacy. Normally, an RS needs to collect as much data as possible to enhance prediction performance (Anand, 2011). However, using personal information for malicious purposes leads to a number of users either refusing to share their personal information on a system or at least providing misinformation. Accordingly, researchers have proposed solutions to mitigate this limitation that can use personal information carefully and sensibly (Ramakrishnan et al., 2001), one of

these studies is (Bilge and Polat, 2012). (Huo et al., 2020) proposed a method to protect users' information by adopting two algorithms, namely geographical location privacy-preserving (GLP) and friend relationship privacy-preserving (FRP) algorithms. By applying these algorithms, recommendation performance improved in terms of precision and recall. In another study, cross-platform privacy information was protected using the locality sensitive hashing (LSH) algorithm which was used to protect some private information and the MinHash technique to protect information related to services that were utilized by users (Qi et al., 2018).

2.4.6 Gray sheep

Gray sheep is a term used to describe users with inconsistent opinions such that they neither agree nor disagree with a group of people. These kinds of users do not make a contribution to simplifying the function of a CF approach. Another term is 'lack sheep', which refers to users with opposing opinions. For example, most users would rate an item with five stars, but black sheep users would rate the same item one or two stars. Subsequently, the prediction of preferred items for this type of user is almost impossible (McCrae et al., 2004).

A hybrid recommendation system has been proposed to alleviate the Gray sheep problem by integrating the CB and CF approaches, as proposed by (Miranda et al., 1999), in whose study the weights of CB and CF were computed, and for each user, these weights were utilized differently according to their behavior (Miranda et al., 1999). (Ghazanfar and Prügell-Bennett, 2014) proposed a method to cluster gray sheep users and isolate them. Then, based on their content-based file and rating behavior, a hybrid method was applied to predict reliable recommendations. Moreover, (Sánchez-Moreno et al., 2016) proposed a different technique to address Gray sheep users by used a playing coefficient to determine Gray sheep users. Then one of the CF approaches was adopted by using counts of the play to predict items.

2.4.7 Shilling attack

A shilling attack is a group of users cooperating to give positive or negative ratings for particular items. This type of user has been determined to use a clustering technique, with which shilling attack users not only have common items but also give these items

similar ratings. Normally, such users inject a database with fake profiles and ratings. Subsequently, the system removes them from the database, after having identified them using an empirical threshold. However, occasionally normal users with real profiles and honest ratings may cooperate with the same items; thus, simple shilling attack detection methods are unable to recognize such real users (Jannach et al., 2010).

Several studies have been proposed to recognize these users. In (Lam and Riedl, 2004), the authors found that the item-based approach was less sensitive to this attack than the user-based approach. In another study proposed by (O'Mahony et al., 2004), in that approach, the robustness was analyzed to resistant potentially malicious perturbations attacks. Moreover, a normalization technique of neighbor-based CF was applied to remove the global effects and selected neighbors based on the residual of global effects (Bell and Koren, 2007). In (Zhou et al., 2016), the shilling attack problem was detected using the support vector machine (SVM) and target item analysis (TIA) methods. The SVM method was adopted to classify shilling attack users, whereas the, TIA method was exploited to decrease false-positive values. The SVM-TIA model failed in terms of recall, but it achieved good results in terms of precision. Table 2.5. shows an example of a shilling attack, such that the attackers have almost similar ratings and they rated every item with a similar rating.

Table 2.5. An example of a shilling attack (Zhou et al., 2016).

	I1	I2	I3	I4	I5		In
User₁	5	2	3	0	0		5
User₂	2	0	4	1	2		3
User₃	4	2	3	0	5		0
User₄	0	3	0	3	4		3
.....	...	...	...	...	...		...
User_m	2	0	4	1	2		3
Attacker₁	2	1	0	0	5		4
Attacker₂	2	2	0	0	5		3
Attacker₃	1	2	0	0	5		2
.....	...	...	...	...	...		...
Attacker_p	2	0	0	0	5		4

2.5 Distance and Similarity Measurement

Distance and similarity metrics have been utilized in many information systems fields, including information filtering, information retrieval, and pattern recognition. Both

metrics use a set of features to compute the closeness or matching of a particular pair of users. In an RS, a well-known classifier algorithm called the K-Nearest Neighbors (K-NN) algorithm is used to classify users by computing the similarity between each pair of users based on their ratings. In some cases, the distance is required to implement a certain task such as the clustering technique, where the largest value indicates the highest similarity leading to zero distance. Accordingly, the distance can be computed as follows (Cohen et al., 2003):

$$distance = 1 - similarity \quad (2.11)$$

In the literature, many similarity algorithms have been used and proposed for different types of data, such as numeric and binary. The concept of computing similarity and the distance between a pair of users/items depends on creating two vectors of features identified by non-zero values (Nilashi, 2016).

In recommender systems, similarity is the mainstay of the memory-based CF and model-based CF approaches. Many similarity methods have been used to compute similarities between two items/users, such as Cosine Similarity (CarlKadie, 1998), Pearson Correlation (Resnick et al., 1994), Spearman's Correlation Coefficient (Herlocker et al., 2017), and Jaccard Coefficient (Finch, 2005). Table 2.6 shows equations of some similarity algorithms that most widely used in the CF approach.

Table 2.6. Similarity measures periodically used for CF approach (Ahn, 2008).

Measures	Definition
Pearson's Correlation Coefficient (PCC)	$sim(u_x, u_y) = \frac{\sum_{h=1}^{n'} (r_{u_x, i_h} - \bar{r}_{u_x}) (r_{u_y, i_h} - \bar{r}_{u_y})}{\sqrt{\sum_{h=1}^{n'} (r_{u_x, i_h} - \bar{r}_{u_x})^2} \sqrt{\sum_{h=1}^{n'} (r_{u_y, i_h} - \bar{r}_{u_y})^2}}$ where $r_{u,i}$ are the ratings of item i by user u, $\bar{r}_u$ is the average rating of user u, and n' is the number of common items for both users.
Cosine (COS)	$sim(u, v) = \frac{\sum_{h=1}^{n'} r_{u, i_h} r_{v, i_h}}{\sqrt{\sum_{h=1}^{n'} r_{u, i_h}^2} \sqrt{\sum_{h=1}^{n'} r_{v, i_h}^2}}$ where $r_{u,i}$ is the rating of item i by user u and n' is the number of common items for users u and v
Adjusted Cosine (ACOS) for Similarity between items	$sim(i_x, i_y) = \frac{\sum_{j=1}^{m'} (r_{u_j, i_x} - \bar{r}_{u_j}) (r_{u_j, i_y} - \bar{r}_{u_j})}{\sqrt{\sum_{j=1}^{m'} (r_{u_j, i_x} - \bar{r}_{u_j})^2} \sqrt{\sum_{j=1}^{m'} (r_{u_j, i_y} - \bar{r}_{u_j})^2}}$ Where $r_{u,i}$ is the rating of item i by user u, $\bar{r}_u$ is the average rating of user u for all the items rated by the user, and m' is the number of users who rated both of the items
Constrained Pearson's Correlation Coefficient (CPCC)	$sim(u_x, u_y) = \frac{\sum_{h=1}^{n'} (r_{u_x, i_h} - r_{med}) (r_{u_y, i_h} - r_{med})}{\sqrt{\sum_{h=1}^{n'} (r_{u_x, i_h} - r_{med})^2} \sqrt{\sum_{h=1}^{n'} (r_{u_y, i_h} - r_{med})^2}}$ where $r_{u,i}$ is the rating of item i by user u, r_{med} is the median value in the rating scale (e.g. 3 in the rating scale of 5 and, it is 4 in a scale from 1 to 7.), and n' is the number of items co-rated by both users
Spearman's Rank Correlation (SRC)	$sim(u, v) = 1 - \frac{6 \sum_{h=0}^{n'} d_h^2}{n'(n'^2 - 1)}$ Where d_h is the difference in the ranks of item h by the two users and n' is the number of common items between users u and v
Jaccard coefficient	$jac_{u,v} = \frac{x}{x+y+z}$ Where x is the number of common trusted users between users u and v. y and z are the number of trusted users of users u and v, respectively.
Dice similarity	$Dice_{u,v} = \frac{2x}{2x+y+z}$

PCC and COS are widely used to compute numeric similarities in the memory-based approach, while Jaccard and Dice similarity are used to compute the binary similarities. For PCC and COS, the range of the similarity lies on a scale between -1 to 1, -1 meaning the objects are entirely dissimilar, 1 indicating complete matches between them, while 0 indicating that there is no relationship between them.

Although many studies have used traditional similarity methods to compute similarity in RSs, many researchers have produced new similarity methods, such as Inner Product (IP) (Lee et al., 2007), Proximity, Impact, and Popularity (PIP) (Ahn, 2008), and Bayesian Similarity (BS) (Guo et al., 2013). Other approaches improved traditional similarity methods by overcoming their limitations, such as Effective Missing Data Prediction (EMDP) (Ma et al., 2007), Rated Item Pools (RIP) (Shi et al., 2009), and Jaccard Uniform Operator Distance (JacUOD) (Sun et al., 2012). Other studies transferred the similarity method concept from particular domains to recommender system domains, such as Electric Circuit Analysis (ECA) (Yang et al., 2012).

Different studies have integrated traditional similarity methods to alleviate the challenges of recommender systems. PCC was incorporated into the probabilistic matrix factorization algorithm to reduce data sparsity (Ma et al., 2011; Cantador et al., 2011), (Guo et al., 2013) incorporated PCC into the SVD method to reduce the data sparsity and cold-start problem (Guo et al., 2015). Mohsin et al. exploited the Jaccard coefficient to cluster users according to their relationships. (Hussein et al., 2016). Parvin et al. proposed a method that adopted PCC into the nonnegative matrix factorization method to enhance prediction accuracy (Parvin et al., 2019).

Due to this study is used PCC and Jaccard similarity, the following examples show the computing similarity between two users using PCC. Table 2.7 depicts the rating matrix for 4 users and 5 items.

Table 2.7. rating matrix.

	I1	I2	I3	I4	I5
U1	4	3	?	2	?
U2	2	?	?	4	4
U3	5	?	4	3	4
U4	3	3	?	2	4

To find the similarity between u3 and u4, the first step computes the average rating of u3 and u4 as follows:

$$\text{Avg (u3)} = (5+4+3+4) / 4 = 4$$

$$\text{Avg (u4)} = (3+3+2+4) / 4 = 3$$

$$\text{Sim}(u3, u4) = \frac{(5-4)(3-3)+(3-4)(2-3)+(4-4)(4-3)}{\sqrt{(5-4)^2+(3-3)^2+(4-3)^2} \sqrt{(3-3)^2+(2-3)^2+(4-4)^2}} = \frac{1}{\sqrt{2} \sqrt{1}} = \frac{1}{1.414} = 0.707$$

The result shows that u3 and u4 have an almost a similar taste to each other.

Another example computes the similarity using the Jaccard coefficient. Table 2.8 shows the matching of two objects X and Y, where each object has a set of features, $X = \{x_1, x_2, x_3, \dots, x_n\}$ and $Y = \{y_1, y_2, y_3, \dots, y_n\}$. In this table, a is the number of match positive features for both objects, b is the number of positive features of x_i , and the number of negative features of y_i , c represents the number of negative features of x_i and the number of positive features of y_i . and d indicates the number of matched negative features for both objects.

Table 2.8. Matching objects.

		Y	
		1 (Presence)	0 (Absence)
X	1 (Presence)	A	B
	0 (Absence)	C	D

Table 2.9 shows an example of binary data vectors for two objects.

Table 2.9. Two binary data vectors.

	D ₁	d ₂	d ₃	d ₄	d ₅	d ₆	d ₇	d ₈	d ₉
X	1	0	1	0	1	1	1	0	1
Y	1	1	0	0	1	1	1	0	0

The Jaccard and Dice similarity are computed as follows:

$$jac_{X,Y} = \frac{a}{a+b+c}$$

Based on Table 2.9, $a = 4$, $b = 2$, $c = 1$, thus the similarity is

$$jac_{X,Y} = \frac{4}{4+2+1} = 0.57$$

$$Dice_{X,Y} = \frac{2a}{2a+b+c} = \frac{8}{8+2+1} = 0.73$$

The same concept can be applied to compute the similarity among users that have social relationships. The binary user-user matrix is created according to the users'

relations, where the value is 1 if there is a relation between two users; otherwise, the value is zero. Then, the similarity can be computed using the Jaccard coefficient.

2.6 Evaluation Metrics

Various metrics in the RS field are utilized to evaluate the recommendation performance of a number of algorithms (Herlocker et al., 2004). In this section, the most widely metrics are demonstrated. According to (Herlocker et al., 2004), there are two types of evaluation: offline and online. Offline evaluation is the division the dataset into two parts: testing and training. A particular algorithm computes a new rating or ranks unseen items. The evaluation process is achieved by comparing the testing part and the outcomes of the training part to measure the quality of the predictions. The speed of computing is one advantage of offline evaluation, and one limitation is its inability to determine which items are truly of interest to users.

A 5 cross-validation is most common in the offline evaluation. A dataset is divided randomly into five equal parts. To guarantee test the whole data, the cross-validation is performed 5 times, where each part can be used for testing and the rest for training. Secondly, online evaluations are implemented in live observing users' behavior, measuring users' satisfaction after recommending a set of items. However, online evaluation is so rare due to it is expensive and impossible.

Many matrices are used to evaluate recommendation performance. Some of these matrices are used to compute top-N recommendations and others are used to evaluate the prediction accuracy. Evaluation metrics can be divided into accuracy metrics, rank-based metrics, and rate coverage metrics.

2.6.1 Accuracy metrics

According to (Sarwar et al., 2001), evaluation accuracy can be classified into statistical accuracy metrics and decision support accuracy metrics.

2.6.1.1 Statistical accuracy metrics

Performance is assessed by comparing a predicted value with an actual user rating in a dataset. The mean absolute error (MAE) (Goldberg et al., 2001) and root mean square

error (RMSE) (Smyth and Cotter, 2000) are widely used metrics to compute prediction accuracy.

$$MAE = \frac{1}{N} \sum_{i=1}^N |r_{u,i} - \bar{r}_{u,i}| \quad (2.12)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (r_{u,i} - \bar{r}_{u,i})^2} \quad (2.13)$$

where N is the total number of predictions, $r_{u,i}$ the actual value in the dataset for item I that is specified by user u , and $\bar{r}_{u,i}$ is the result of the production system. If the values of the MAE and RMSE are close to zero, the results are highly accurate, whereas if the values are large, the results are of low accuracy.

2.6.1.2 Decision support accuracy metrics

These metrics measure the successful classification of recommended items. In other words, decision support accuracy metrics measure the number of times that the system can classify relevant items as positive or irrelevant items as negative. Relevant items refer to items that users are liked in the actual rating. This is type of metric success for binary ratings, where statistical accuracy metrics cannot evaluate a system using binary ratings. However, for numeric ratings, decision support accuracy metrics can measure the performance of such systems by converting numeric ratings to binary ratings. For each user, if the user has rated an item with a value more than the average user's rating, the user is considered to have liked the item. The drawback of these metrics is that items with no ratings are considered irrelevant because users may not have seen them at all (Herlocker et al., 2004).

Several metrics are used to evaluate a prediction by distinguishing items as relevant or irrelevant. Precision, Recall, and F-measure is the common frequent measures in decision support accuracy metrics (Cleverdon et al., 1966). Precision is the fraction of recommended items that are relevant to the user. Recall is the ratio of relevant items to the part of the recommended items. F-measure is the translation of precision and recall into a single metric.

These metrics divide items into four groups, which is called a confusion matrix as shown in Table 2.10.

Table 2.10. A confusion matrix.

	Relevant	Irrelevant
Recommended	TP	FP
Not recommended	FN	TN

A TP (true positive) is the number of relevant items that is recommended by the system. An FP (false positive) is the number of irrelevant items that are recommended by the system. A TN (true negative) indicates the number of irrelevant items that are not recommended. The relevant items that are correctly recommended are placed in the FN (false negative) category.

$$precision = \frac{TP}{TP+FP} \quad (2.14)$$

$$recall = \frac{TP}{TP+FN} \quad (2.15)$$

$$f_measure = \frac{2 \text{ precision } . recall}{precision+recall} \quad (2.16)$$

2.6.2 Rank-based metrics

Unlike accuracy metrics, rank-based metrics order or rank items that are recommended to a user. Several metrics apply the scheme, such as the Discount Cumulative Gain (DCG) metric (Järvelin and Kekäläinen, 2002), the Rank-Biased Precision (RBP) metric (Moffat and Zobel, 2008), and the recovery rate metric (Jalili et al., 2018).

Discount cumulative gain (DCG) ranks recommended items according to their evaluation. A higher rank is given to the more relevant item.

$$DCG_{b_u} = \sum_{i=1}^b R_i \sum_{i=b+1}^n \frac{R_i}{\log_b r_i} \quad (2.17)$$

where u denotes a user, b controls the degree of items' shares in the DCG, (which is set to 2), n is the particular rank position, r is the ranking score at position i , which is 1 if the last position of the list, R_i is 1 when the item is relevant and 0 otherwise.

The DCG is normalized by dividing the DCG value over the perfect ranking of the user's preference, which matches the order of the items with the actual rating list in the test part.

$$NDCG_u = \frac{DCG_u}{DCG_{max}} \quad (2.18)$$

The average of all users is calculated by dividing the summation of the NDCG of all users by the total number of all users as follows:

$$NDCG_{total} = \frac{\sum_{u=1}^m NDCG_u}{m} \quad (2.19)$$

The optimal value is 1, which refers to the best recommended for relevant items.

Rank-biased precision (RBP): RBP is similar to DCG, where for each user more shares are given to highly ranked relevant items. Unlike DCG, however, RBP uses a geometric sequence:

$$RBP_u = (1 - p) \sum_{i=1}^N (R_i * p^{r_i-1}) \quad (2.20)$$

$$RBP_{total} = \frac{\sum_{u=1}^m RBP_u}{m} \quad (2.21)$$

where r is the ranking score at position i , which is 1 if the last position of the list. R_i is 1 when the item is relevant and 0 otherwise. p is a probability value between 0 and 1, a small value of p means the user examines the top-ranked items, while a large value of p refers to the user testing the items in lower ranks.

Recovery rate: In this metric, the performance is evaluated based on the correct ranking of items. The recovery rate of user u and the average recovery rate are computed as follows:

$$Recovery_u = \frac{1}{|N_R|} * \sum_{i \in N_R} \frac{r_i}{c_i} \quad (2.22)$$

$$Recovery_{total} = \frac{\sum_{u=1}^m Recovery_u}{m} \quad (2.23)$$

Here, N_R is the number of relevant items in the test part of user u , r_i the rank of item i in the recommended items list, c_i is the number prospective items that are going to be recommend to user u .

2.6.3 Rate coverage

An efficient recommender system focuses not only on recommending popular items, it also focuses on a wide ratio of items that the system provides or the number of users that receive the recommending service. In other words, when a system recommends more items or serves more users, the performance of the system is deemed efficient (Herlocker et al., 2017). Normally, the rate coverage (RC) metric of users is utilized to measure how many cold-start users are served in a particular algorithm or to measure the diversity of predicting items by an algorithm.

$$\text{User- RC} = \frac{\text{number of cold start users that correctly reached}}{\text{total number of cold start users}} \quad (2.24)$$

$$\text{Item} - \text{RC} = \frac{\text{number of relevant items in top}_n \text{ list}}{\text{total number of items in the system}} \quad (2.25)$$

2.7 Social Network

With the development of Web 2.0, social media have been applied in many applications such as blogs, social bookmarking, and social networks (Dang and Viennet, 2013). Social networks are the most prevalent application of Web 2.0, permitting users to share images, videos, special events, and to create relationships.

Social network information has utilized widely in RSs. The motivation to use social networks in recommendation systems is that users are affected by their friends' or trustees' tastes. For example, when someone is deciding whether to buy, watch, or listen to an item, he/she will check which items have been selected by his/her friends. Therefore, social network information is useful in RSs.

Many studies have adopted social networks as a support factor to alleviate the problems of RSs and boost the accomplishment of recommendations by exploiting information from the social networks (Xie et al., 2015; Reafee et al., 2016). Two

categories of social network are available, namely those that are friends' network-based and trust network-based.

2.7.1 Friends network-based

Friends network-based method means friendship relations can be engaged to find mutual friends where the RS can use friends' ratings to recommend items. The undirected graph clarifies a friend's relationships. For example, user1 has a friendly relationship with user2, so concurrently user2 is a friend of user1. Figure 2.9 shows an example of friends' relationships.

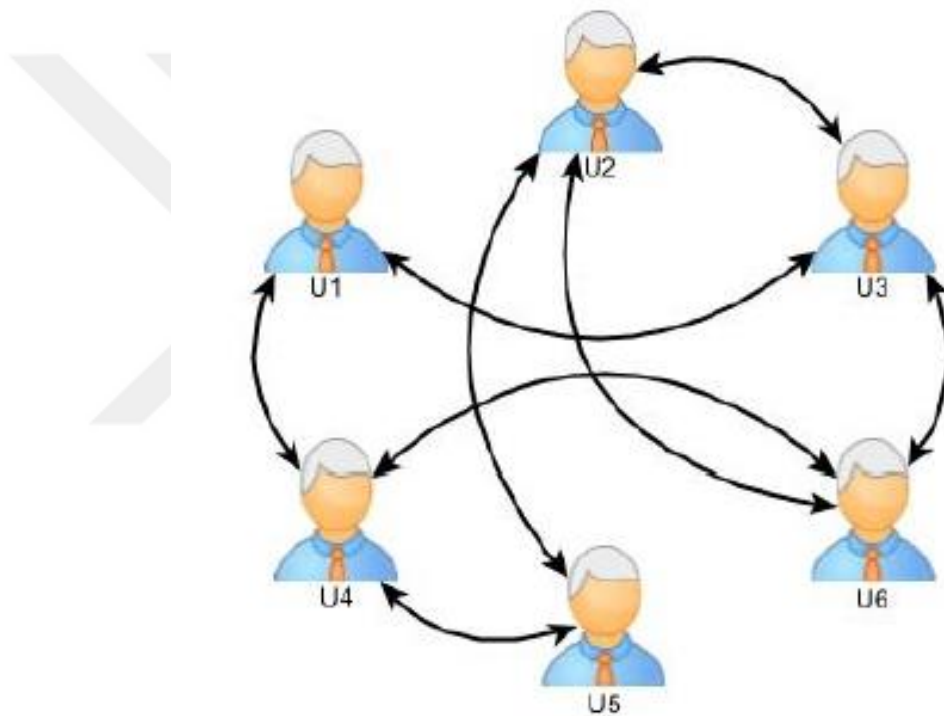


Figure 2.9. Example of friends' relationships.

In the literature, several studies have employed social relationships in a model-based CF approach in order to overcome problems of RSs, especially the sparsity and cold-start problems (Unger et al., 2016; Wang et al., 2015). The modern Web 2.0 facilitates users' utilization of their relations in different applications; thus, social information can be integrated with MF to enhance accuracy further. (Guo et al., 2016) produced a model that exploited trust relationships to determine new friends, thereby enhancing prediction accuracy.

In another approach, (Xie et al., 2015) utilized social network information in a typical manner, but learning repetitions contributed to the improvement of RS performance. (Ma et al., 2011) produced two models using social information, which integrated explicit social relations into the probabilistic matrix factorization (PMF). Furthermore, (Reafee et al., 2016) proposed EISR, the acronym for explicit-implicit social recommendation. Thus, four models were produced based on implicit relations that are extracted using the link prediction technique. The EISR method utilized three information sources that are incorporated into the PMF method.

After reviewing the research that has been conducted in previous studies, some of them ignored implicit relationships, other studies such as EISR did not exploit implicit rating feedback. It can be concluded that the traditional methods still lack sufficient prediction accuracy.

2.7.2 Trust network-based

Similar to friends' relationships, trust relationships are used to solve RS problems, such as sparsity and cold-starts. The trust network-based approach employs trust relationships to present new items. Here, trust user values are given to other users. Unlike friend relations, a trust statement means directed relations, such as when user 1 trusts user 5, but, user 5 may not trust user 1. Figure 2.10 illustrates an example of direct relationships.

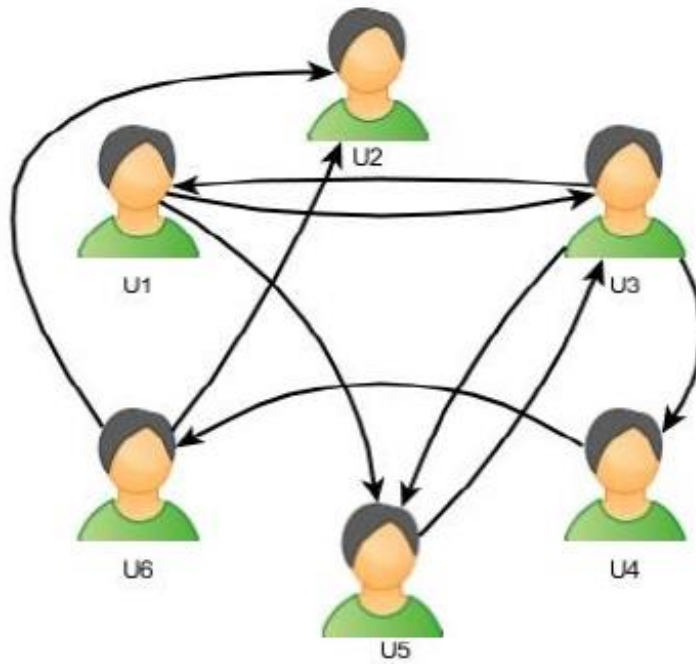


Figure 2.10. Example of directed relationships.

Also, Figures 2.11 and 2.12 show how explicit relations are represented in an user-user matrix.

The screenshot shows a Notepad window titled "Trust Relation File". The content is a list of pairs of IDs, where the first ID is the trustor and the second ID is the trustee. The pairs are listed as follows:

trustorID	trusteeID
1	3
1	5
1	6
1	1
2	7
2	8
2	9
2	5
3	7
3	9
3	1
3	1
4	9
4	1
5	7
5	8
5	9
5	9
6	9
7	9
9	1
9	7

Figure 2.11. Sample of explicit relationships.

	U1	U2	U3	U4	U5	U6	U7	U8	U9
U1			1		1	1			
U2	1						1	1	1
U3	1				1		1		1
U4	1								1
U5	1						1	1	1
U6									1
U7									1
U8									
U9	1						1		

Figure 2.12. A user-user matrix for explicit relations.

There are two types of trust relationships: explicit and implicit. Explicit trust relationships are obtained directly from the dataset and are mostly binary values. For example, when user x trusts user y , the value is 1, while 0 denotes that there is no relationship between users. An implicit trust relationship is extracted based on explicit relations; it provides more information than explicit relationships. However, binary explicit relations cannot differentiate between the tastes of users (Son et al., 2020). Binary explicit trust relations consider every user to have the same weight when there is a relation between them. Accordingly, explicit trust expresses relationships inaccurately. For instance, user A decides to mark user B , user A trusts user B in IT issues, but not in clothes, for example.

To overcome this problem, implicit trust relations can be used. Implicit trust gives different weights for each item based on the user's rating. Several properties are considered to develop implicit trust relations, such as asymmetry, transitivity, dynamicity, and context-dependence (Son et al., 2020; Guo et al., 2012). Asymmetry refers to a directional or weighted graph to express the relations among users. Transitivity denotes that the relations must be spread. For example, when user x trusts user y and user y trusts user z , there is a high likelihood that user x will trust user z . Trust dynamicity means that the trust is changeable over time. Finally, context-dependence implies that a trust relationship is under area control, which varies based on the current area. Implicit trust relationships are computed using users' ratings, which can be calculated as follows:

$$T_{u,v} = A_{u,v}/A_u \quad (2.26)$$

where A_u is the number of items that are rated by user u , and $A_{u,v}$ is the set of common items rated by users u and v .

Due to the sparsity problem of the user-item matrix, the combination between similarity of users and implicit trust may be helpful to alleviate data sparsity. Therefore, the adjusted weight between users u and v is computed as follows:

$$w_{u,v} = \begin{cases} \frac{2*sim_{u,v}*T_{u,v}}{sim_{u,v}+T_{u,v}}, & sim_{u,v} * T_{u,v} \neq 0, \text{ and } sim_{u,v} + T_{u,v} \neq 0 \\ sim_{u,v}, & sim_{u,v} \neq 0, \text{ and } T_{u,v} = 0 \\ T_{u,v}, & T_{u,v} \neq 0, \text{ and } sim_{u,v} = 0 \\ 0, & otherwise \end{cases} \quad (2.27)$$

Where $sim_{u,v}$ denotes the similarity between users u and v , which is computed using the Pearson Correlation Coefficient (PCC) or any numeric similarity method in Table 2.6.

Implicit trust relationships have been adopted in several studies. (Guo et al., 2012) proposed a method that incorporated CF and explicit trust relations directly by combining the ratings of the active user with trusted neighbors using Equation 2.28.

$$\tilde{r}_{u,i} = \frac{\sum_{v \in TN_u} t_{u,v} r_{v,i}}{\sum_{v \in TN_u} t_{u,v}} \quad (2.28)$$

$\tilde{r}_{u,i}$ is the computed rating, TN_u the number of trust users of user u , $r_{v,i}$ is the rating value of the item i rated by user v .

The SocialMF method was proposed by Jamali and Ester (Jamali and Ester, 2010), integrated trust social relations into matrix factorization to provide suggestions to active users. However, this method did not exploit implicit trust relations. TrustSVD was a common method proposed by (Guo et al., 2015), who exploited both explicit and implicit trust relations and integrated them into the SVD method. The TrustSVD method improved the prediction performance in terms of accuracy. The major limitation of this method was that transitivity implicit trust did not apply. Additionally, (Guo et al., 2014) proposed two models of neural networks to overcome the limitations of explicit trust relationships. Other studies would filter out some of the unnecessary

information that affected prediction accuracy negatively by using reliability measurements (Hernando et al., 2013; Moradi and Ahmadian, 2015). (Hernando et al., 2013) utilized implicit trust relationships and computed the reliability values to remove unnecessary information before the prediction process. However, this method was not adaptable for RS having explicit trust relations. (Moradi and Ahmadian, 2015) proposed a method that used implicit trust relations and users' similarities to alleviate the data sparsity problem.

Although previous studies improved RSs performance, all of these studies considered that users are one community (cluster) regardless of dividing them into many groups based on their tastes or their social relations. We think it is unreasonable to consider all users as one community. Another point is that they did not utilize the SVD++ method, which is a very successful method when it combined with valuable sources.

2.8 Datasets

Several free datasets are available for researchers in the RS field. Various domains of datasets are available, including movies, music, books, documents, e-commerce, applications in markets, and web searches. Some of these datasets contain rating information only, some of them hold social information such as followers and those followed. Others have demographic information (age, gender, habits, and nationality), and some datasets focus on the Internet of Things, such as GPS and real-time health signals (Durante et al., 2013). Table 2.11 shows some of the common datasets that are used in most studies with some basic information such as the number of users and items as well as the domain.

Table 2.11. Summary of the common datasets in the RSs.

Datasets	# users	# items	Density ratio	Domain	Available data	
					Rating-feedback	Social network
MovieLens 1K	943	1682	6.30%	Movie	1-5	None
Jester	59132	140	21.28%	Joke	-10 To +10	None
Bookcrossing	278858	271379	0.001%	Books	1-10	None
Netflix	480189	17770	0.117%	Movies	1-5	None
FilmTrust	1508	2071	1.14%	Movie	0.5 – 4	Trust
Ciao	7375	99,749	0.0379%	general items	1-5	Trust
Last.Fm	2100	18,744	0.0027%	Music	Listens counts	Friendship
Flixster	147,612	48,794	0.1138%	Movie	0.5-5	Trust
Douban	129,490	58,541	0.22%	movie, books, music	1-5	Friendship
Epinions 665K	40,163	139,738	0.0118%	movie, books, music, etc	1-5	Trust
Delicious	1867	69,226	-----	Tagging to other users	0 & 1	Trust

2.9 Summary

In this chapter, an overview of the concepts and methods of recommendation systems was presented. In the beginning, we first explained the concept of RS, how the represented in a matrix. Afterward, the types of RSs which are also expanded to subtypes were demonstrated. Furthermore, the common methods that were used in different studies, challenges of RSs were cleared. Due to RS depends mainly on similarity measurement, most main of the similarity methods were shown. Moreover, a set of common machine learning methods that are widely used in RS was displayed. A part of the chapter was dedicated to explaining social networks and their relation with the RS. Evaluation metrics that consider one of the substantial factors in the RS to evaluate the performance of a particular method were displayed. Finally, the common datasets in this filed with their statistics were mentioned in this chapter as well. In the next chapter, the methodology steps to implement this study will be presented.

3. RESEARCH METHODOLOGY

By studying the previous researches, the recommender system problems are addressed. These problems are related to the social network. Thus, the present social recommender system models that depend on the dimensional reduction methods need to be improved. The objective of this research is to integrate social relationships into four machine learning techniques namely, singular value decomposition (SVD), SVD++, clustering technique, and overlapping community detection method to enhance prediction accuracy.

In this chapter, the methodology steps are produced. The methodology includes all the stages that are essential to execute the research systematically. These stages refer to the generic framework of the research that discusses the phases, methods and tools. This chapter consists of four sections, starting with the introduction in section 3.1. Section 3.2 demonstrates the research design. The research framework is illustrated in section 3.3. Finally, section 3.4 depicts the evaluation metrics.

3.1 Research Design

The methodology is portrayed in this chapter to realize the objectives of this research. It includes various aspects such as the methods, data sources, tools, and evaluation metrics. Four actions should be taken to obtain the answers of the research questions. First, the MSRA-SVD model is proposed. It involves incorporating social relationships and explicit feedback information into the singular value decomposition (SVD) method. Social relationships include explicit and implicit relationships. Implicit relationships are hidden information extracted from the social networks. This information can increase the prediction accuracy. Second, the MSRA-SVD++ model is produced by integrating explicit and implicit social relationships and explicit and implicit feedback information into the singular value decomposition plus (SVD++) method. These two models are employed to mitigate the data sparsity problem. Detecting user's community is a critical issue to determine the user's preferences. Clustering techniques can be used to group users on the basis on their relationships. Hence, the cold start problem can be mitigated and the third objective is realized. Eventually, for the final objective, a novel method of overlapping community detection is adopted to exploit users' options and overcome the cold-start problem. To

summarize, this research takes four main objectives, all of which use dimension reduction methods. The first two objectives aim to alleviate the data sparsity problem. Whereas the third and fourth utilize community detection in two techniques clustering and overlapping. Both techniques are employed to overcome the cold-start problem.

3.2 Research Framework

In this section, the research framework is illustrated. It includes six phases, starting from literature review and ending with system evaluation. The research framework displays the steps for implementing this research. Figure 3.1 depicts the phases of the research framework.

The operational framework includes six phases. Namely, phase 1: Literature review, problem statements, and data preparation, phase 2: Integrating Explicit, Implicit social relations, and explicit feedback information into SVD. Phase 3 is exploiting social relations and rating feedback information, and integrating them into SVD++. Moreover, phase 4 is Clustering community detection. Phase 5 includes overlapping community detection. Finally, system evaluation indicates showing the metrics that are used to evaluate the proposed methods. The details of all phases are demonstrated in the following subsections.

Phase 1:	Literature review and data preparation
Problem formulation, literature review and dataset preparation	
Phase 2:	Integrating Explicit, Implicit social relations, and explicit feedback information into SVD
Exploiting explicit relationships, implicit relationships that computed by Multi-steps resource allocation (MSRA), and explicit feedback information, afterwards incorporate all these sources into SVD	
Phase 3:	Exploiting social relations and rating feedback information and integrating them into SVD++
Exploiting explicit and implicit social relationships, with explicit and implicit feedback information, then incorporate all these sources into SVD++	
Phase 4:	Clustering community detection
Using clustering algorithm to detect users' community where each group of users belong to a particular group, which can be used to alleviate the cold start problem	
Phase 5:	Overlapping community detection
A novel overlapping community detection is proposed to mitigate cold-start problem	
Phase 6:	System evaluation
The metrics are explained, which include MAE, RMSE, precision, recall, and F-measure	

Figure 3.1. Research framework.

3.2.1 Phase 1: Literature review and data preparation

Here, the first phase of this research is devoted to clarify the initial steps of the research. This phase includes four parts problem formulation, literature review, determining the dataset, and the preprocessing.

a. Problem formulation

As usual in any research, the first task is highlighting the problems of the research, where the problems are addressed from general issues until specifying the problems that occur in a particular research field. In this study, the problems are labeled in social networks and community detection. These problems are forward to present in the research questions.

b. Literature review

The literature review involves studying related work. In chapter 2, the definition and types of the recommendation systems (RSs) are explained. Additionally, some techniques utilized in RSs, such as the similarity, are demonstrated. The challenges of RSs are described in detail. Furthermore, previous and recent related work that includes social networks and users' community is reviewed. Moreover, the methods that are used in this research are also demonstrated, such as Link Prediction (LP) and machine learning techniques. Finally, the evaluation metrics of RSs used to measure different methods are clarified.

c. Dataset selection

The selection of the dataset is a vital issue in RSs area because each dataset contains different information and each kind of these datasets can be used in different techniques. Dataset selection should meet the research questions and study objectives for an efficient evaluation of the proposed models. Additionally, several datasets were used in previous studies. Some of them contain users' rating feedback such as MovieLens, Jester, etc. While other datasets encompass friend relationships such as Last.Fm, Flixter, and Douban. Another kind of datasets has trust social relationships information such as Ciao, FilmTrust and Epinion. Friend social relationships and trust social relationships are appropriate for this study as they include explicit and implicit social relations and users' rating information, which are fit to cover the objectives of this research. Accordingly, three real-world datasets, namely, Last.Fm, FilmTrust and Ciao are utilized in this research.

Last.Fm is a standard dataset that encompasses social network and undirected graph specifically. Last.Fm is released in the framework of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (Cantador et al., 2011). Last.Fm is the United Kingdom's online music system. The users can listen to

several kinds of music, and it enables the users to create their friends' relationships. The dataset consists of 2100 users and 18744 items. The dataset contains the number of listens for each artist.

Ciao is a general shopping dataset used as a standard dataset in RSs. It contains a social network (directed graph). This dataset is released in The Fifth ACM International Conference on Web Search and Data Mining (Tang et al., 2012). The Ciao dataset enables users to establish relationships. The dataset contains two files. The first file is for rating, which is a 5-star rating. Whereas the second file is trust relationships and involves a pair of values. For example, (1,2) means user 1 trust in user 2. Ciao dataset contains 7375 users and 99746 items with 278483 user ratings.

Finally, FilmTrust is the third dataset. It is released on The 23rd International Joint Conference on Artificial Intelligence in 2013 (Guo et al., 2013). FilmTrust is a movie rating and it has social trust relations information. Additionally, this dataset includes 35497 ratings distributed among 1508 users and 2071 items. The rating includes 8 levels from 0.5-4 with a step of 0.5.

d. Data preprocessing

After selecting the dataset, the next step is to prepare this data to employ it and apply the proposed method. Data preparation involves three sub-steps data mapping, selecting useful data, and dividing the data.

Regarding data mapping, Last.Fm contains the listening count for each user to the set of artists. Due to the 5-stars rating is the common rating system, we need to map these values into rating values between 1-5. According to (Liu et al., 2013) proposed formula 3.1 to convert listening count to the rating values.

$$r = \begin{cases} \lfloor \log_{10} l \rfloor + 1, & \text{if } \lfloor \log_{10} l \rfloor + 1 \leq 5 \\ 5, & \text{otherwise} \end{cases} \quad (3.1)$$

Where r is the rating integer value, l is the listening count, $\lfloor x \rfloor$ is the operator of rounding towards zero.

After data mapping, now the dataset is filtered out to extract the useful information. In the Last.Fm dataset, users having 5 or more friend relationships are selected randomly. The number of selected users is 1,122 with all items (18,744). The total number of user ratings is 55,140 with 99.73 % sparsity. The number of friend relations is 11,064.

For the Ciao dataset, all users with more than one item rating and at least one relation are chosen. And due to memory consumption, the items with one or zero ratings are removed. After applying these conditions, 6767 users and 22229 items remain. The overall detecting rating is 177,331 with 99.96% sparsity. The number of trust relations is 111780. Finally, FilmTrust dataset involves 1508 users and 2071 items, with 35,494 ratings and 98.86% sparsity. the number of trust relations is 1853. All users and items are selected in this study. Tables 3.1, 3.2 and 3.3 depict the basic statistics for Last.FM, Ciao and FilmTrust datasets respectively.

$$sparsity = 1 - \frac{overall\ ratings}{\#users * \#items} \quad (3.2)$$

Table 3.1. The Last.Fm sample Data Statistics.

Statistic	Amount
# Users	1,122
# Items	18,744
# Ratings	55,140
% Sparsity	99.73
Friend relations	11,064

Table 3.2. The Ciao sample Data Statistics.

Statistic	Amount
# Users	6,767
# Items	22,229
# Ratings	177,331
% Sparsity	99.88
Friend relations	111780

Table 3.3. The FilmTrust sample Data Statistics.

Statistic	Amount
# Users	1508
# Items	2071
# Ratings	35,494
% Sparsity	98.86
Friend relations	1853

In this study, the users are split into two groups on the basis of their ratings. Users with more than four ratings are classified as non-cold-start users, while the rest are cold-start users. For cold-start users, two models are proposed. On the other hand, for the non-cold-start users, different two models are presented.

Additionally, the machine learning procedure works by dividing the dataset into two parts: training and testing. Some studies divide datasets into a particular percentage for training and testing. The training part is used to learn the model for the behavior patterns of the users, while the testing part is used to evaluate the model by comparing the predicted values computed on the basis of the training part with actual values in the testing part. The standard division is 80% for training and 20% for testing. However, dividing a dataset into parts does not guarantee testing all the data. Therefore, the f-fold cross-validation provides an optimal option for efficient evaluation. For example, 5-fold cross-validation divides a dataset into 5 folds: 4 folds (80%) for training and one-fold (20%) for testing. The cross-validation implements five times, each time the testing part is switched with one part of the other four. Consequently, all the data will be evaluated.

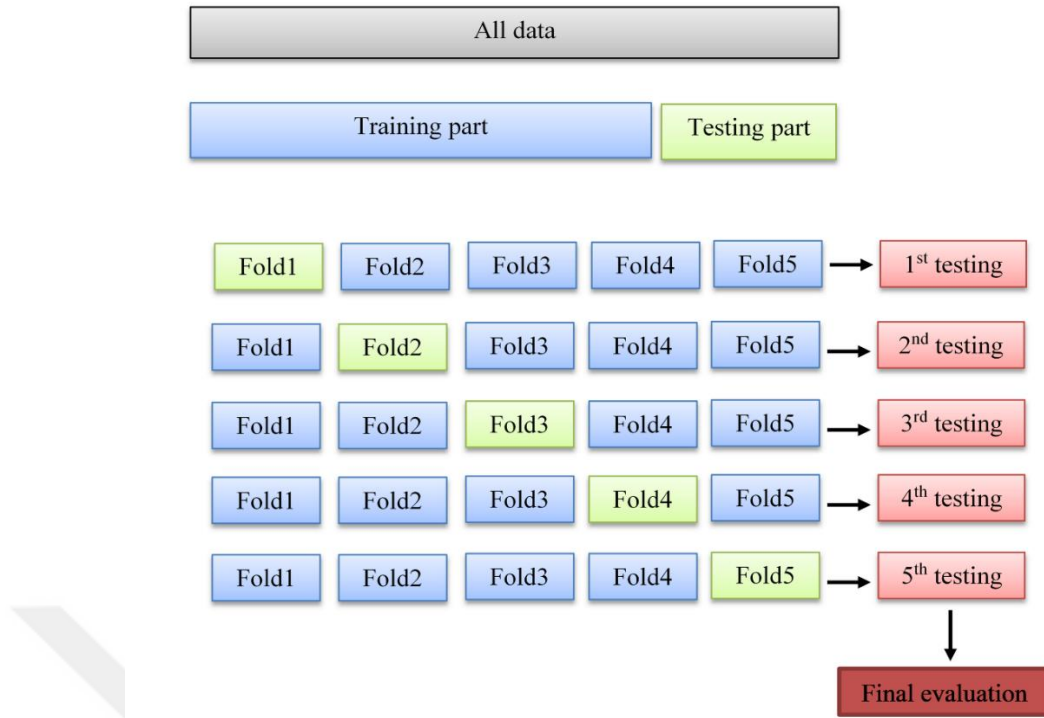


Figure 3.2. Illustration of 5-fold cross-validation.

After reviewing the first phase, the remaining phases are gathered in Figure 3.3. The general design of the proposed models encompasses four phases.

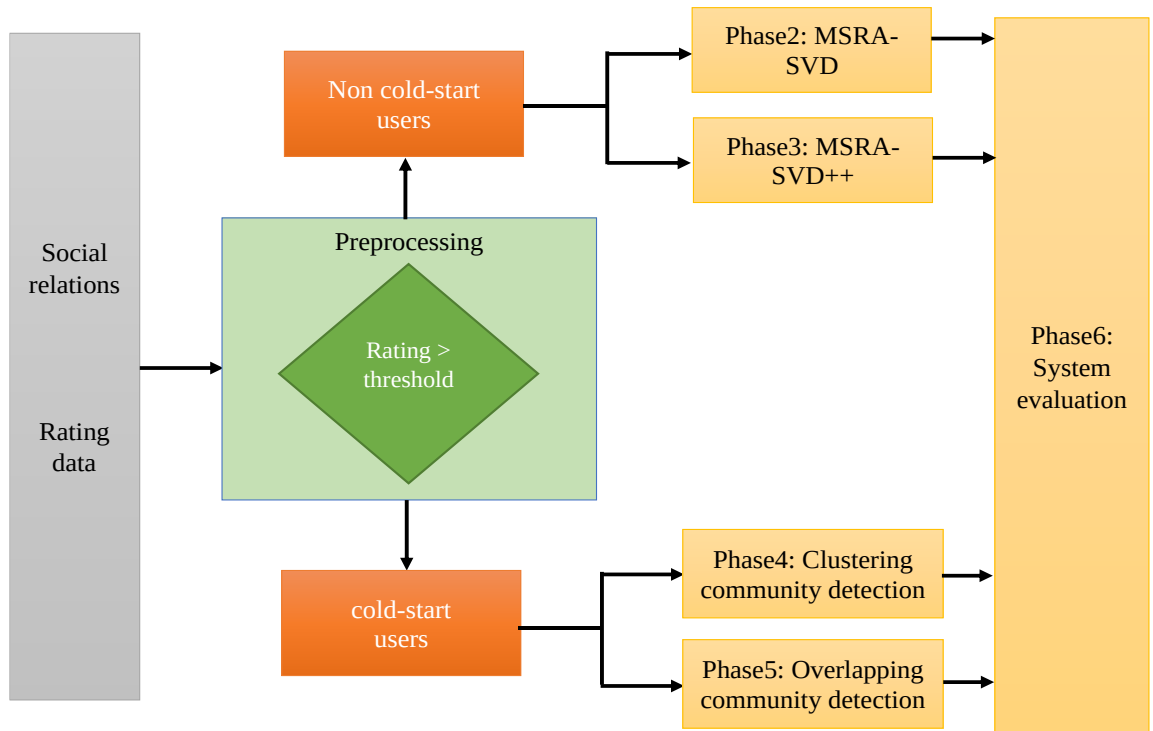


Figure 3.3. General view of the proposed models.

3.2.2 Phase 2: Integrating explicit, implicit social relations and explicit feedback information into SVD

In this phase, the first model of this study is produced. This model involves three sources of information: explicit social relationships, implicit social relationships and explicit feedback information. Explicit social relationships and explicit feedback information can be obtained directly from the dataset. Whereas, implicit social relationships are extracted using a link prediction technique based on explicit social relationships. Furthermore, all the sources of information are incorporated into the SVD method to produce the MSRA-SVD model. Chapter 4 presents the details of this phase. Figure 3.4 shows a summary of this phase.

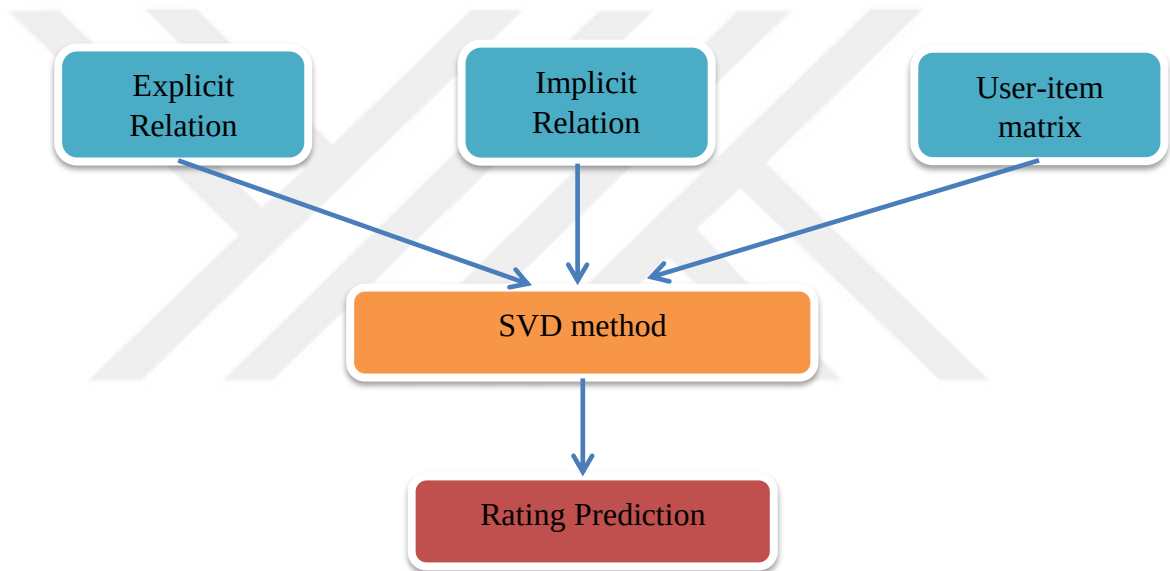


Figure 3.4. Summary of phase 2.

Three of the recommender systems datasets are utilized: Last.Fm, Ciao and FilmTrust. All of these datasets include explicit social relationships and rating information. The main tasks in this phase are extracting implicit social relations and integrating the sources into the SVD method.

Implicit relations are hidden information that can be extracted from explicit relations to find a set of friends by applying link prediction techniques used in different similarities methods. In this research, the Multi-Steps Resource Allocation (MSRA) similarity method is used to find out implicit relations.

Explicit social relations are the input to extract implicit relationships. The first step reads the social relations data to determine the users' friends and non-friends. Afterward, the multi-steps resource allocation is applied to extract new friends of a particular user. The similarity is computed between the new users and a target user. A given threshold is determined to compare the similarity value with it. If the value exceeds the threshold, a new relation will be established.

After acquiring implicit social relationships, the prediction process starts by exploiting previous ratings and social relations (explicit and implicit). To optimize the prediction accuracy, the stochastic gradient descent (SGD) algorithm is adopted. Additionally, five metrics are utilized to evaluate this model. They are MAE, RMSE, precision, recall and f-measure. The whole process is explained in detail in Chapter 4.

3.2.3 Phase 3: Exploiting social relations and rating feedback information and integrating them into SVD++

This phase is similar to the previous one. Four sources of information rather than three are used in this phase: explicit social relations, implicit social relations, explicit feedback information and implicit feedback information. As mentioned in phase 2, explicit social relations and explicit feedback information can be attained directly from a dataset. Implicit social relations are extracted using the MSRA method. The last source of information in this phase is implicit feedback information, which refers to information that can be extracted from users' behavior. Implicit feedback information is extracted from browsing, clicking or purchasing items. All the sources are combined into the SVD++ method.

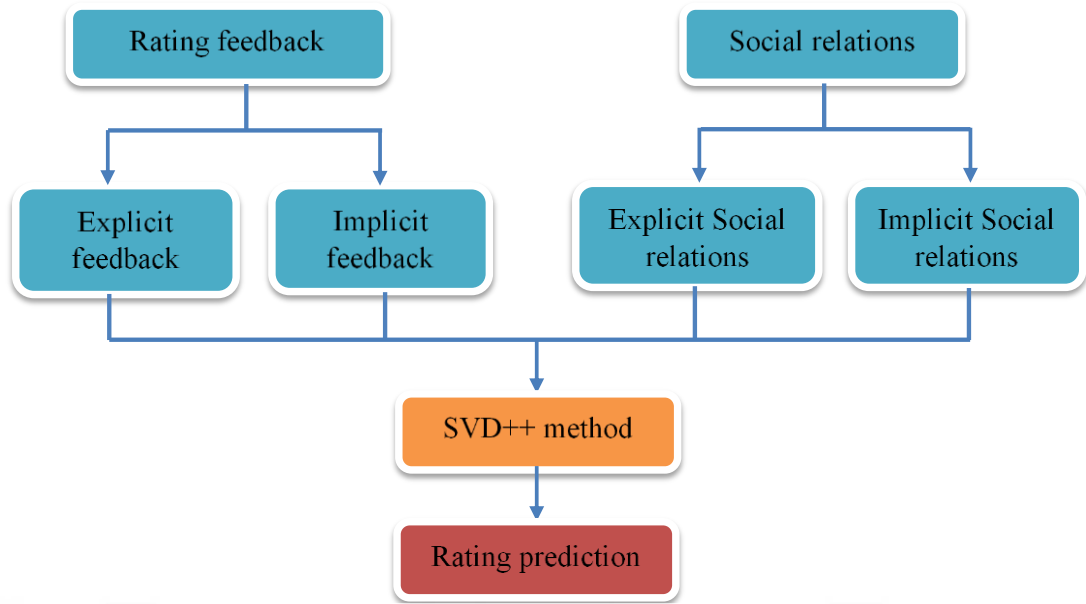


Figure 3.5. Overview of phase 3.

As in the previous phase, three datasets are used: Last.FM, Ciao and FilmTrust. Moreover, the SGD algorithm is adopted to reduce the regularization square error. Finally, the performance evaluation is achieved by using MAE, RMSE, precision, recall and f-measure. In chapter 5, the whole process of this phase is explained in detail.

3.2.4 Phase 4: Clustering community detection

Clustering is a very common technique used to group similar individuals based on their properties in one cluster and dissimilar individuals in other clusters. Clustering can reduce implementation time by seeking the required individual in one cluster instead of the whole dataset. In this phase, the clustering technique is utilized to group users into several clusters to overcome the cold-start problem. Partitioning around medoid (PAM) is adopted in this study. The distance between each pair of users is computed based on their explicit and implicit social relationships. Moreover, implicit trust relationships are exploited to predict more relations. Afterward, the outcomes of explicit and implicit social relationships and implicit trust relations are combined in weight values. Later, users are clustered into groups according to the computed weights. The clustering process can be summarized as follows:

- 1- Selecting a set of medoids randomly
- 2- Determining the medoids correctly
- 3- Grouping the users according to last medoids

After obtaining the clusters, the SVD++ method is applied for each cluster to compute the prediction values. Eventually, the performance evaluation of the model is achieved using five metrics: MAE, RMSE, precision, recall, f-measure and user coverage. The complete process of this model can be seen in chapter 6. Figure 3.6 shows an overview of this model.

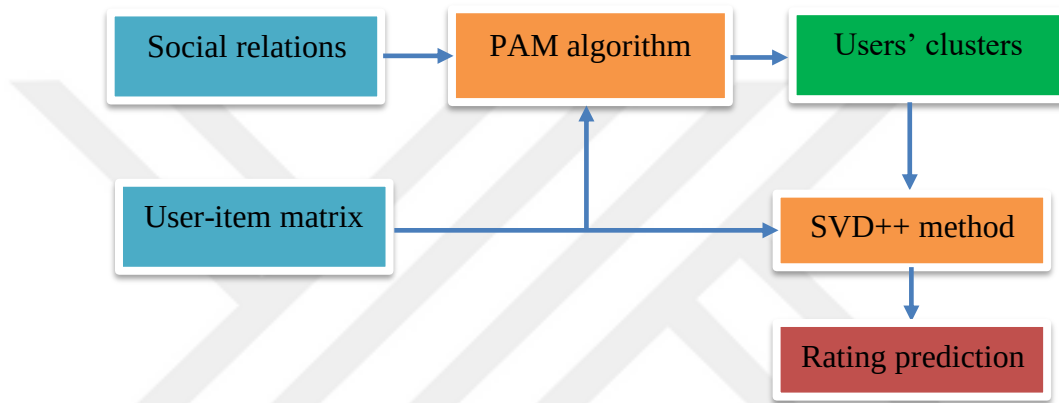


Figure 3.6. An overview of the clustering community detection.

3.2.5 Phase 5: Overlapping community detection

The previous phase assumed that the users are disjointed. In the real world, a user engages in several items at the same time. Accordingly, grouping a user into many clusters can overcome the clustering limitation and alleviate the cold-start problem. In this phase, an overlapping technique is proposed to tackle the aforementioned problems. As in the previous phase, social relations and implicit trust relations are utilized to compute the distances between each pair of users. Afterward, the PAM algorithm is applied to cluster users into a set of groups. Later, the clusters are merged. The overlapping steps are summarized as follows:

- 1- Identifying initial clusters
- 2- Determining overlapped communities according to the distance
- 3- Overlapping the clusters that have nodes belonging to different clusters

Moreover, in each community, the SVD++ method is adopted to predict the missing values. The performance evaluation is accomplished using MAE, RMSE, precision, recall, f-measure and user coverage. This phase is explained in chapter 7. Figure 3.7 depicts an overview of this phase.

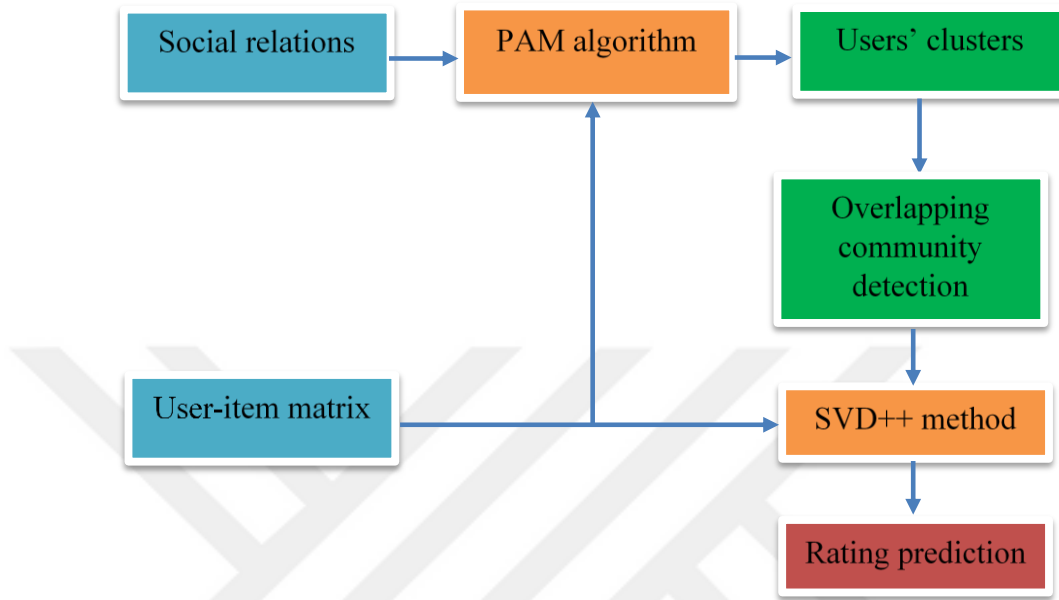


Figure 3.7. An overview of overlapping community detection.

3.2.6 Phase 6: System evaluation

In this phase, the evaluation metrics are demonstrated. Many matrices are used to evaluate the performance of the recommendation. Some of these matrices are used to compute top-n recommendation while others are used to judge the accuracy of the prediction. In this study, four different models are proposed. The first two models are exploited to alleviate the data sparsity problem, whereas the last two models are dedicated to addressing the cold-start problem. In all models, the datasets are divided into two parts: training and testing. In the training part, the data is used to build and learn the model. Moreover, after learning the model, new ratings are computed and subsequently compared with the actual ratings in the testing part. To evaluate the performance of the proposed models, accuracy metrics are adopted which involves statistical accuracy and decision support accuracy metrics. For statistical accuracy metrics, mean absolute error (MAE) and root mean square error (RMSE) are used. Furthermore, decision support accuracy metrics include precision, recall and f-measure.

MAE and RMSE are two accuracy predictions well known in the literature (Chen et al., 2018). They are defined as equations 3.3 and 3.4 for MAE and RMSE respectively.

$$MAE = \frac{1}{N} \sum_{i=1}^N |r_{u,i} - \bar{r}_{u,i}| \quad (3.3)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (r_{u,i} - \bar{r}_{u,i})^2} \quad (3.4)$$

Where N is the total number of the prediction $r_{u,i}$ is the actual value of the dataset for items i that are given by user u , and $\bar{r}_{u,i}$ is the result of the production system. When the value of MAE and RMSE is close to zero, this means high accuracy. When the value is high, however, this indicates low accuracy.

Precision, recall and f-measure metrics measure the successful classification of recommended items. These metrics are computed on the basis of the confusion matrix. Four factors are extracted from this matrix: True positive (TP), True negative (TN), False negative (FN) and False positive (FP). For more details, see chapter 2 (2.6.1). The following equations are used to compute the aforementioned metrics.

$$precision = \frac{TP}{TP+FP} \quad (3.5)$$

$$recall = \frac{TP}{TP+FN} \quad (3.6)$$

$$f_measure = \frac{2 \cdot precision \cdot recall}{precision+recall} \quad (3.7)$$

Table 3.4 shows the outline of the phases including the research questions, the objectives, the aims of the study, the metrics applied in each model, and the comparisons with other studies. All these issues are highlighted in this table.

Table 3.4. Outline of the thesis framework.

Phase	Research objective	Aim	Research question	Performance measure
2	To develop rating explicit feedback information, explicit social relations, and implicit social relations depending on the MF technique (SVD) used to reduce data sparsity.	To reduce the data sparsity problem and improve prediction accuracy	How can social information and explicit/implicit rating feedback information be incorporated into the SVD method?	MAE, RMSE, precision, recall, and f-measure
3	To develop explicit and implicit rating feedback information, explicit social relations, and implicit social relations relying on the MF technique (SVD++) applied to alleviate data sparsity.	To mitigate data sparsity and overcome the limitations of the previous models	How can social information and explicit/implicit rating feedback information be incorporated into the SVD++ method?	MAE, RMSE, precision, recall and f-measure
4	To evolve a clustering approach that can detect users' communities more accurately using the PAM clustering algorithm to decrease the cold-start problem.	To highlight the cold-start problem	How can user clustering based on social relations and confidence values can be created using the PAM algorithm?	MAE, RMSE, precision, recall and f-measure
5	To improve an overlapping method that can detect users who belong to different communities to alleviate the cold-start problem.	To tackle the cold-start problem and defeat the clustering problem	How can detecting users' communities who belong to several clusters be established using a novel proposed overlapping method?	MAE, RMSE, precision, recall and f-measure

3.3 Experimental Preparations

In this section, the evaluation metrics and parameters setting applied in this study are illustrated. The experiments are implemented on three real-world datasets: Last.Fm, Ciao, and FilmTrust for more detail see section 3.2.1. This study includes four models. All the models and the performance evaluation are implemented in Matlab 2018b. Moreover, Gephi software is utilized to visualize and present the clustering process in

chapters 6. Finally, the experiments are performed on Win 10 with intel ® core (TM) i7 CPU @ 2.8 GHz and 24GB RAM.

3.3.1 Comparisons with the proposed approach

In the literature, several methods have been applied in RSs field. Comparisons process is used to evaluate a proposed method to check whether it outperforms other methods. Therefore, in this study, comparison is divided into two parts: basic methods and state-of-the-art methods.

3.3.1.1 Comparison with rating-based methods

Here two methods are compared with the proposed models namely, SVD and SVD++. Both methods use a user-item matrix source as an input without including any additional sources. The implementing process steps are done to predict new values for the missing rating (for more detail read section 2.3.2.1). The comparisons are achieved with the models MSRA-SVD (chapter 4) and MSRA-SVD++ (chapter 5).

In the literature, there are no results that used decision support accuracy metrics in Last.Fm and Ciao datasets. Therefore, we chose the SVD method to compare with our models in terms of precision, recall and F-measure.

3.3.1.2 Comparison with the state-of-the-art methods

State-of-the-art methods are standard methods in a particular field. In RSs, several methods are considered state-of-the-art. In this study, methods that used social relations are selected to benchmark their results with the results of the proposed models. The methods are explained as follows:

- 1- RSTE: This method was proposed by (Ma et al., 2009). The authors employed trust friends' opinions in the social trust network to recommend items. This study is tested by Ciao and FilmTrust datasets.
- 2- SocialMF was proposed by (Jamali and Ester, 2010). In this study, explicit trust social relations and implicit trust social relations were combined into the matrix factorization technique. This study is tested by Ciao and FilmTrust datasets.

- 3- SocRec which stands for “Social recommendation”: It was proposed by (Canny, no date) and (Ma et al., 2011). This method uses explicit social information along with the user-item matrix and is evaluated on Ciao and FilmTrust.
- 4- EISR was proposed by (Reafee et al., 2016). This approach used three sources, namely, explicit friend relations, implicit friend relations and rating information. All these sources are incorporated into the PMF method. This method is applied in Last.Fm dataset.
- 5- TrustSVD was proposed by (Guo et al., 2015). The explicit and implicit trust social relations were exploited by a weight λ -regularization technique. It is used for extended user-item specific latent feature vectors.
- 6- Trust ANLF proposed (Parvin et al., 2019), the authors integrated the MF technique with social trust information to mitigate the cold-start user problem.
- 7- RTARS was proposed (Ahmadian et al., 2019). In this study, the trust-aware RS was improved by enhancing rating profiles of cold-start users.

The following table summarizes the comparison methods. It also shows the evaluation metrics for each model. In this study, two models are proposed (MSRA-SVD and MSRA-SVD++) for non-cold-start users and other two models (clustering and overlapping community detection) for cold-start users. Table 3.5, the proposed models outperform the other studies in terms of MAE and RMSE. Moreover, MSRA-SVD++ outperforms the MSRA-SVD model in terms of MAE, RMSE, precision, recall and F-measure. Finally, the overlapping community detection model achieves better results than the clustering community detection model in all metrics used.

Table 3.5. Summarizing of the comparisons of the proposed method.

Phase	Chapter	Dataset	Approach	Compared study	Metrics
2	4	Last.Fm, Ciao, FilmTrust	MSRA-SVD	RSTE, PMF, SVD, EISR	MAE, RMSE, precision, recall and F-measure
3	5	Last.Fm, Ciao, FilmTrust	MSRA-SVD++	RSTE, SocialMF, SocRec, PMF, SVD, SVD++, and EISR	MAE, RMSE, precision, recall and F-measure
4	6	Ciao, FilmTrust	PAM clustering algorithm and SVD++	RSTE, SocialMF, SocRec, SVD++, TrustSVD, TrustANLF, RTARS	MAE, RMSE, precision, recall and F-measure
5	7	Ciao, FilmTrust	Overlapping community detection and SVD++	RSTE, SocialMF, SocRec, TrustSVD, TrustANLF, RTARS	MAE, RMSE, precision, recall and F-measure

3.4 Parameters Setting

A set of parameters are utilized to implement the proposed models. These parameters involve the size of the dimension of the latent features (d), regularization parameter λ , number of training, explicit relations β_e , implicit relations β_i , and the learning rate. The meanings of these parameters are depicted below. The best values of these parameters that guarantee the production of the best results are shown in Tables 3.6, 3.7, and 3.8.

- 1- The dimension of the latent features (d) is the size of vectors U and V , which refer to the user and item features, respectively.
- 2- Regularization parameter λ is a constant value that is used to regularize the factors and avoid over-fitting.
- 3- β_e and β_i are constant values greater than zero that used to control the impact of explicit and implicit relationships.
- 4- The learning rate α is a constant value that is used to control the learning ability by reducing the regularization error in each implementation.

Table 3.6. Parameters setup for MSRA-SVD model.

Dataset Parameter	Last.Fm	Ciao	FilmTrust
β_e	0.002	0.01	0.1
β_i	0.002	0.01	0.1
α	0.03	0.01	0.03
Λ	0.001	0.01	0.001
D	5, 10	5, 10	5, 10
Iterations	5	5	5

Table 3.7. Parameters setup for MSRA-SVD++ model.

Dataset Parameter	Last.Fm	Ciao	FilmTrust
β_e	0.002	0.001	0.005
β_i	0.002	0.001	0.005
α	0.01	0.006	0.01
λ_e and λ_i	0.001	0.0001	0.001
λ_1 and λ_2	10, 25	4, 8	7, 10
D	5, 10	5, 10	5, 10
Iterations	5	5	5

Table 3.8. Parameters setup for clustering community detection models.

Dataset Parameter	Ciao	FilmTrust
α	0.06	0.07
Λ	0.01	0.2
λ_1 and λ_2	6, 8	20, 15
D	5, 10	5, 10
Iterations	5	5

3.5 Summary

This chapter described the methodology of the study. Firstly, the research design, which includes an overall review of the study, is demonstrated. This is followed by the research framework that involves five crucial phases. These phases include theoretical and practical parts achieved in this study. In phase 1, the literature review and data preparation are highlighted and the statistics of the datasets were given. The data preprocessing was clarified as well in phase 1. Regarding phase 2, the MSRA-SVD model was shown, whereas phase 3 was devoted to producing the MSRA-SVD++ model. Furthermore, in phase 5, clustering community detection using the PAM algorithm was described. Overlapping community detection was presented in phase 5. This chapter ends up by with viewing the parameters that will be utilized to rule this research, the evaluation metrics and the benchmarking methods.

4. EXPLOITING IMPLICIT SOCIAL RELATIONSHIPS VIA DIMENSION REDUCTION TO IMPROVE RECOMMENDATION SYSTEM PERFORMANCE

Due to the rapid growth of social networks in many applications, it is substantial to exploit this source in a recommendation system to enhance prediction accuracy. Several studies have used social relations in the literature (Ma et al., 2009; Jamali and Ester, 2010; Ma et al., 2011; Reafee et al., 2016; Guo et al., 2016). However, there is a lack of sufficient prediction accuracy. Some of the previous studies ignore implicit relations. Other studies exploited implicit relations but they neglected the other steps neighbors which may lead to vital information. Hence, totally ignoring implicit relations or merely using part of it may negatively affect prediction accuracy.

In this chapter, an improved method is proposed by integrating explicit/implicit social relations and the users' ratings into the SVD method to overcome the data sparsity problem. The contributions of this chapter include the way implicit social relations are extracted, where a new method is adopted, namely, a multi-steps resource allocation (MSRA). This method is incorporated with the SVD method. Furthermore, by applying this way, the recommendation system performance improves.

This chapter is dedicated to addressing the second research question in this study.

RQ2 How can social relations and explicit rating feedback information be incorporated into the SVD method?

Answering this question is achieved by integrating both explicit and implicit social relations with users' ratings. Users' ratings denote the rating of items by users. Moreover, an explicit relationship is a real connection between a pair of users. A multi-steps resource allocation method is applied to extract implicit relations.

The following subsections illustrated all processes in this model.

4.1 Data Sources

The major function of this model is integrating three sources of information: explicit social relations, implicit social relations and users' ratings.

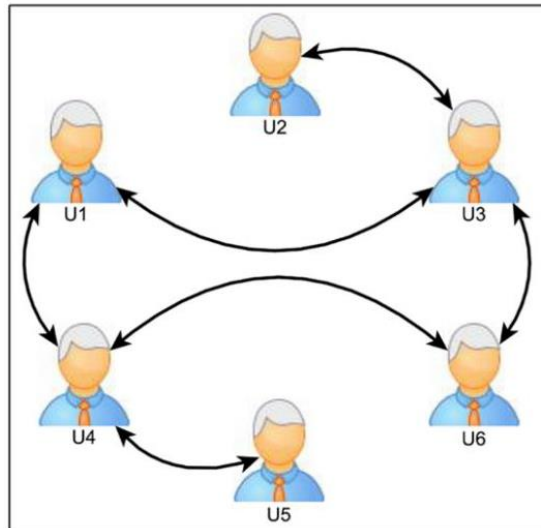
- i- Users' ratings: Let us assume M is the number of users and N is the number of items. A matrix called a user-item matrix (R) can be created with size $M*N$. r_{ij} is a rating value given to user i and item j .
- ii- Explicit social relations: Let us assume $G1 = (V, E1)$, where $(u, f) \in E1$ denotes explicit relations regardless of being direct or indirect.
- iii- Implicit social relations: let assume $G2 = (V, E2)$, where $(u, f) \in E2$ denotes implicit relations (friend-of-friend).

Figure 4.1 (A) shows an example of the user-item matrix, (B) depicts explicit social relations and (C) displays implicit relations.

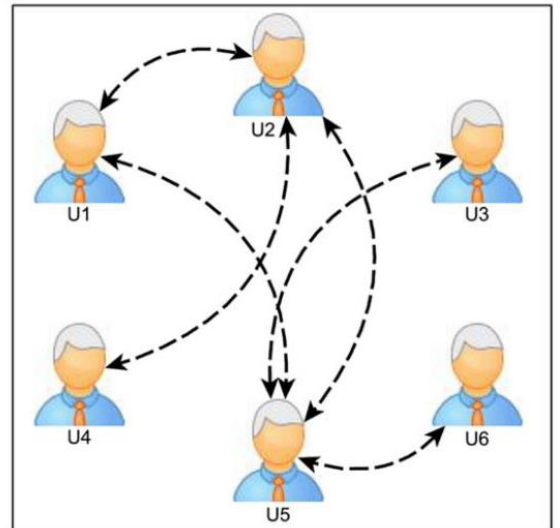
The first two resources are directly available from the dataset, whereas implicit relations are extracted from explicit ones.

	I1	I2	I3	I4	I5	I6
U1		4	3		5	
U2	3	5		4		
U3				3		5
U4	1		4			3
U5	4	3			4	

(A)



(B)



(C)

Figure 4.1. An example of data sources for the proposed model.

4.1.1 Link prediction

Link prediction is the task of predicting information missing from a social network. Three types of methods are utilized: link prediction based on features, link prediction based on the structure and a hybrid of these two methods. Many studies utilize these three types of methods, e.g, (Cantador et al., 2011; Ricci et al., 2011). These studies have attempted to improve the link prediction performance; however, the node characteristics are not always accessible. This study utilizes link prediction based on the structure to extract implicit relations in a social network.

In the social network structure, a graph is used in which the nodes and edges correspond to users and relations respectively. The social network is the main aspect in determining implicit relations. Therefore, the resource allocation index (RAI) (Dang and Viennet, 2013) method is used to predict new links. This algorithm is implemented by finding the direct common friends of x and y , where x and y are non-friends (no connection in the social graph) and computing the similarity between x and y via the following formulas:

$$sim(x, y) = \sum_{z \in |r(x) \cap r(y)|} \frac{1}{|r(z)|} \quad (4.1)$$

Where x and y are the two users; $r(x)$ and $r(y)$ represent the sets of the friends of x and y respectively.

However, RAI uses the first level of the common neighbors and ignores the other neighbors. This may lead to a failure to consider some vital information.

4.1.2 Multi-steps resource allocation

As mentioned before, a multi-steps resource allocation (MSRA) algorithm is used to extract implicit social relations. The MSRA method was proposed by Wu and Li (Wu and Li, 2014). The authors proposed two-steps resource transmission as $MSRA_2$, as expressed in Eq (4.2), and three-steps transmit resource transmission as $MSRA_3$. In this work, $MSRA_2$ is used to transmit resources and to predict new relationships.

$$s_{xy}^{MSRA2} = s_{xy}^{MSRA1} + \sum_{z1 \in Z1} \sum_{z2 \in Z2} \frac{1}{k_{z1}} \frac{1}{k_{z2}} \quad (4.2)$$

Where $Z1$ refers to $r(x) \cap r(y)$, while $Z2$ refers to $r(z1) \cap r(y)/Z1$. s_{xy}^{MSRA1} is computed using (4.1).

The MSRA is computed from explicit relationships. For instance, consider two unconnected nodes, say, x and y . The lists of friends for both nodes should be determined. The similarity of the common neighbors for both nodes is computed via inverse summation. If the value exceeds a threshold, a new relationship will be created.

Figure 4.2 presents an example of computing MSRA in social network nodes. In this figure, x and y are unconnected nodes and their common neighbors are A, B, and C. By using RAI, the similarity is computed by calculating $1/3 + 1/4 + 1/3 = 0.92$ since nodes A and C have 3 relationships each while B has 4 relationships. In Figures 4.2 (a) and 4.2 (b), we obtain the same result using RAI because it depends only on the directed connections between the targeted nodes. However, in Fig. 4.2 (b), there is more than one path from node x to node y through nodes A, B, and C. Thus, the score of Figure 4.2 (b) should be higher than that of Figure 4.2 (a). In two steps, MSRA computes the score of Figure 4.2 (b) as follows: First, it computes the score for the direct neighbors of x by applying the same approach as in RAI. That is, by multiplying the number of neighbors of node A with the number of neighbors of node D (since node D is a second-step neighbor of node x). The remaining nodes are multiplied in the same way. The full computation is $1/3 * 1/3 + 1/4 * 1/3 + 1/4 * 1/3 + 1/3 * 1/3 = 0.39$. Then, the result is added to the value that was computed previously (0.92).

This method increases the scores between unconnected nodes. Subsequently, it increases the number of relationships as demonstrated in the previous example. Now, the implicit relations of nodes x and y are A, B, C, D and E rather than A, B and C only, as computed using RAI. The MSRA algorithm is presented as Algorithm 1.

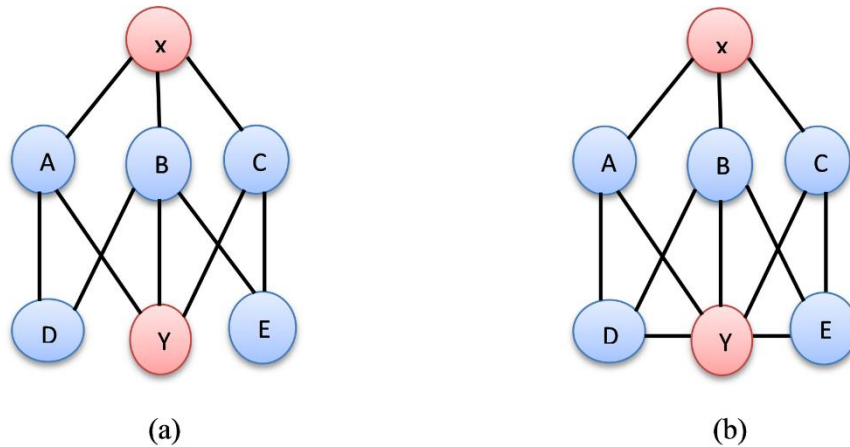


Figure 4.2. Social network nodes.

Algorithm1: computing MSRA

Input: social array, threshold

Output: MSRA array

Index=1

Foreach user i in social array

Foreach user j in social array

If i non-friend with j

Foreach user k in social array

If i friend k and j friend k

Foreach user t in social array

If j friend t and k friend t

$s1 = \text{no. of friends of each } k$

$s2 = \text{no. of friends of each } t$

$m = (1/s1) * (1/s2)$

endif

endfor

$s = s + m$

endif

endfor

temp array(index) = s

index = index + 1

endif

endfor

endfor

index = 1

foreach user i in social array

for each user j in social array

if i non-friend j

foreach user k in social array

If i friend k and j friend k

$z = \text{no. of friends for each } k$

$s = s + (1/\text{each } z)$

endif

endfor

```

temp array(index) = temp array(index) + s
    if temp array(index) > threshold
        msra(n) = temp array(index)
        n = n + 1
    endif
    index = index + 1
endif
endfor
endfor

```

Figure 4.3 depicts an overview of MSRA methods. In this figure, from social network structure, sets of friends and non-friend are determined. For each non-friend pair, the MSRA algorithm is applied to create a list of common friends between these nodes of this pair. An inverse similarity is computed for all common friends of each non-friend pair of users. Afterward, a comparison between the similarity result and a threshold is achieved. If the comparison is greater than a threshold, a new relation will be established between this non-friend pair.

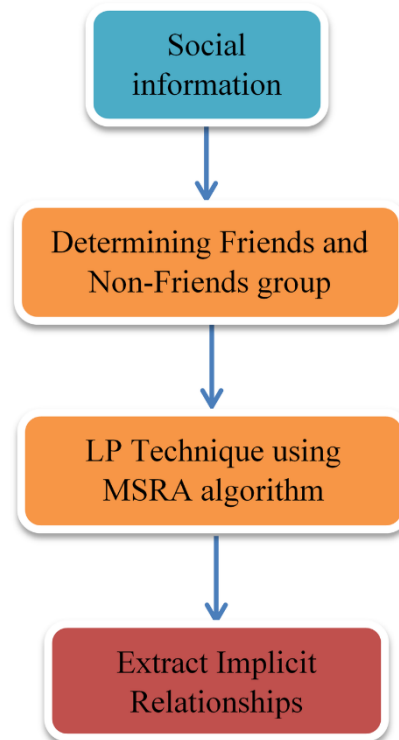


Figure 4.3. An overview of the MSRA algorithm.

4.1.3 Singular value decomposition (SVD)

The SVD method is a prevalent dimension reduction method used in many studies. It is a latent factor method. In this method, each user u and each item I are associated

with user factors vector (p_u) and item factors vector (q_i) respectively. Both vectors belong to R^d and they have d dimensions with real values R . Figure 4.4 depicts the decomposition process of the SVD method.

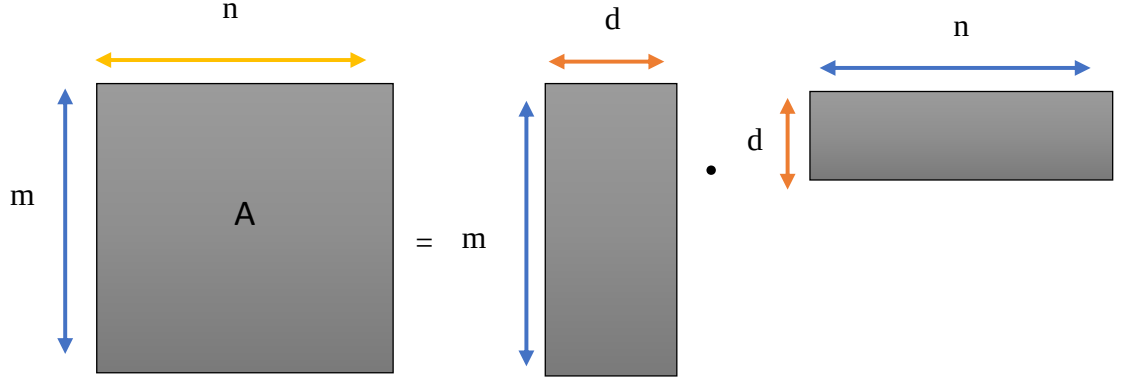


Figure 4.4. The decomposition process of the SVD method.

The prediction is computed on the basis of the inner product of vectors p and q as follows:

$$A = p_u \cdot q_i \quad (4.3)$$

The baseline is added to the previous equation to increase the accuracy. It is computed as follows:

$$baseline = \mu + b_u + b_i \quad (4.4)$$

μ is the mean of the whole ratings, while b_u and b_i are the user's bias and item's bias rating from the mean. b_u and b_i are computed using the following equations.

$$b_i = \frac{(\sum_{u \in R(i)} r_{u,i} - \mu)}{|R_i|} \quad (4.5)$$

$$b_u = \frac{(\sum_{u \in R(u)} r_{u,i} - \mu - b_i)}{|R_u|} \quad (4.6)$$

Where $r_{u,i}$ is a rating value given to the item i by user u , whereas R_i is the number of items that were rated by a particular user. R_u denotes the number of users who rated a particular item.

The aforementioned parameters are regularized using the following constraint:
 $\lambda(b_i^2 + b_u^2 + \|p_u\|^2 + \|q_i\|^2)$.

Regularization means that parameters' values are restricted to a specific threshold. Additionally, an optimization technique can be exploited with the SVD method to reduce the regularization error. The stochastic gradient descent (SGD) algorithm is a frequent algorithm used in dimension reduction methods. In this algorithm, the training involves several times of loop. The parameters (p_u , q_i , b_i , and b_u) are updated in each round to reduce the regularization errors as follows:

$$E = \sum_{(u,i) \in m} (r_{ui} - \mu - b_u - b_i - q_i^T p_u)^2 + \lambda(b_i^2 + b_u^2 + \|p_u\|^2 + \|q_i\|^2) \quad (4.7)$$

$$b_u = b_u + \alpha (E - \lambda b_u) \quad (4.8)$$

$$b_i = b_i + \alpha (E - \lambda b_i) \quad (4.9)$$

$$p_u = p_u + \alpha (E q_i - \lambda p_u) \quad (4.10)$$

$$q_i = q_i + \alpha (E p_u - \lambda q_i) \quad (4.11)$$

4.2 The MSRA-SVD Model

After the extraction of implicit relationships using the MSRA method, other sources are ready to be used. The following steps clarify the implementation of the proposed model.

Step 1: For explicit friends, a user-user matrix is created. The similarity between each pair of explicit friends is computed via the Pearson correlation coefficient (PCC) as follows:

$$sim(i, f) = \frac{\sum_{h=1}^n (r_{i,jh} - \bar{r}_i)(r_{f,jh} - \bar{r}_f)}{\sqrt{\sum_{h=1}^n (r_{i,jh} - \bar{r}_i)^2} \sqrt{\sum_{h=1}^n (r_{f,jh} - \bar{r}_f)^2}} \quad (4.12)$$

$r_{i,j}$ is the rating of item j by user i , while $r_{f,j}$ represents the rating of item j by user f . f is a friend of i regardless of an explicit or implicit friend. $\bar{r}_x$ refers to the average rating of a user. n is the number of common items between users i and f . In this method, the similarity value between i and f ranges from $[-1,1]$. Therefore, the higher value

means that the two users are more similar. The similarity values are placed in the created matrix.

Step 2: For implicit friends, another user-user matrix is created. The similarity between each pair of implicit friends is calculated via the PCC method (4.13).

$$sim(i, f^*) = \frac{\sum_{h=1}^n (r_{i,jh} - \bar{r}_i)(r_{f^*,jh} - \bar{r}_{f^*})}{\sqrt{\sum_{h=1}^n (r_{i,jh} - \bar{r}_i)^2} \sqrt{\sum_{h=1}^n (r_{f^*,jh} - \bar{r}_{f^*})^2}} \quad (4.13)$$

f^* denotes implicit friends. The similarity values are placed in the implicit friends' matrix.

Step 3: From the rating file, a user-item matrix is created. Then, non-cold-start users are separated according to their ratings, where a user with more than five ratings is kept.

Step 4: All these sources are employed in SVD to produce the new model. The computed similarity is utilized by the optimization technique to reduce the regularization error. Moreover, as the similarity is calculated for the users only, the regularization error is divided into two equations. The first equation reduces the regularization error for users' factors. This is achieved by adding the similarity values. The second equation reduces the regularization error for items' factors. Accordingly, the regularization error is modified by integrating the results of explicit and implicit similarity to produce equations (4.14) and (4.15).

$$\begin{aligned} \frac{\partial E}{\partial U_u} = & \sum_{(u,i) \in m} (r_{ui} - \mu - b_u - b_i - q_i^T p_u)^2 + \lambda(b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2) + \\ & \beta_e \sum_{i=1}^u \sum_{f \in F(i)} sim(i, f) \|p_i - p_f\|_{fro}^2 + \\ & \beta_i \sum_{i=1}^u \sum_{f^* \in F^*(i)} sim(i, f^*) \|p_i - p_{f^*}\|_{fro}^2 \end{aligned} \quad (4.14)$$

$$\frac{\partial E}{\partial V_i} = \sum_{(u,i) \in m} (r_{ui} - \mu - b_u - b_i - q_i^T p_u)^2 + \lambda(b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2) \quad (4.15)$$

β_e and β_i are constant values that are used to control the impact degrees of the explicit and implicit relationships respectively. $F(i)$ and $F^*(i)$ denote the number of user i 's explicit friends and implicit friends respectively. Via the stochastic gradient descent

optimization technique, the regularization square error is reduced. The following equations (equations (4.16), (4.17), (4.18) and (4.19)) are used to update the factors b_u , b_i , p and q .

$$b_u = b_u + \alpha (E - \lambda b_u) \quad (4.16)$$

$$b_i = b_i + \alpha (E - \lambda b_i) \quad (4.17)$$

$$p_u = p_u + \alpha \left(\frac{\partial U}{\partial E} q_i - \lambda p_u \right) \quad (4.18)$$

$$q_i = q_i + \alpha \left(\frac{\partial V}{\partial E} p_u - \lambda q_i \right) \quad (4.19)$$

Step 5: After optimizing the factors above, the SVD method is applied to compute the missing predictions as follows:

$$r_{u,i} = \mu + b_u + b_i + q_i^T p_u \quad (4.20)$$

Step 6: the performance evaluation is done by comparing the computed values in the training part by the proposed model with the actual values in the test part. Five metrics are used to judge the proposed model. They are MAE, RMSE, precision, recall and f-measure. Figure 4.5 displays an overview of the proposed model.

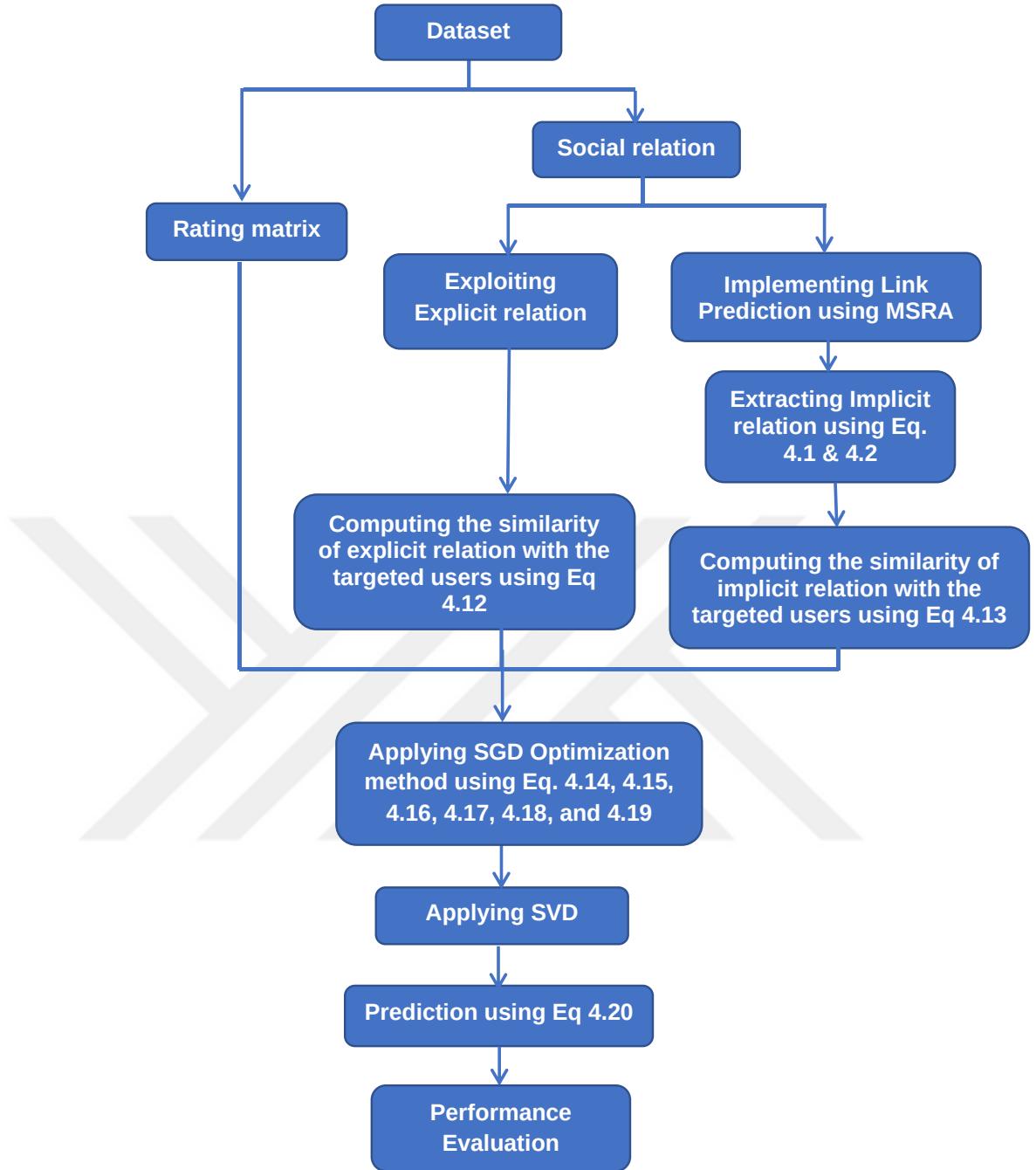


Figure 4.5. Overview of the MSRA-SVD model.

4.3 Experimental Preparation

In this section, the datasets used in this study are described and the evaluation metrics are presented.

4.3.1 Description of datasets

Three real-world datasets are tested in this model. They are FilmTrust, Ciao and Last.Fm. the description is explained in chapter 3. Table 4.1 shows the statistics of these datasets.

Table 4.1. Statistics of the datasets.

Dataset	Last.FM	Ciao	Film Trust
# users	1122	6767	1508
# items	18745	22229	2071
# rating	55,140	185759	35497
% Density	0.0026	0.0012	1.14
# explicit relation	11064	111780	1853
# implicit relation using MSRA	177987	610710	3138

4.3.2 Experimental setup

The implementation of this study depends on parameters for realizing the optimal results. These parameters are listed in Table 4.2.

Table 4.2. Parameter setup.

Dataset	Last FM	Ciao	Film Trust
β_e	0.002	0.01	0.1
β_i	0.002	0.01	0.1
α	0.03	0.01	0.03
λ_e and λ_i	0.001	0.01	0.001
d	5, 10	5, 10	5, 10
Iterations	5	5	5

To determine whether our proposed method enhances the prediction performance, state-of-the-art approaches are benchmarked against our study. These approaches include PMF (Mnih and Salakhutdinov, 2008), SVD (Paterek, 2007), SocReg (Ma et al., 2011) and EISR (Reafee et al., 2016).

The most similar studies in the literature to ours are EISR and SocReg. Thus, for equitable comparison, conditions such as the number of users, the explicit relationships for all users, the dimension of latent space and the number of iterations are used as listed in Table 4.2. The testing view in this study includes users that have at least one relationship and one rating. It is selected by distributing users on the training and testing parts randomly.

4.3.3 Results and discussion

A5-fold Cross-validation is adopted in this experiment. The datasets are divided randomly into five folds. Four-folds are used for training (80%) and one-fold for testing (20%). To guarantee test all the data, each execution is done five times, during which the average is computed to produce the outcome of each implementation. Moreover, the experiment is implemented several times and the mean is produced to display the eventual result.

4.3.3.1 Statistic accuracy results

Tables 4.3, 4.4 and 4.5 depict the accuracy results of non-cold-start users (those with more than five ratings) for FilmTrust, Ciao and Last.Fm datasets respectively. The accuracy results are computed using statistic accuracy metrics and decision support accuracy metrics. The following subsections are devoted to display the results of the proposed model using these metrics.

MAE and RMSE metrics are used in this type. For both metrics, high values indicate low accuracy. On the other hand, if the values are close to zero, the results are highly accurate. Two results are displayed in both tables when the number of dimensions is 5 and 10. The best results are written in Boldface.

The proposed models are compared with four state-of-the-art studies. PMF and SVD use rated information only, whereas SocReg and EISR utilize social relations. All these studies are shown in Table 4.3.

Table 4.3 shows the influence of social information on improving the prediction process. The effect of social information on non-cold-start users is obvious. Comparing the results of PMF and SVD against social relations models depicts how

social information has a vital role to boost accuracy. Although we can exploit the ratings of users who have enough rating history to predict new items, social relations assist in prediction as well. There is a big difference between methods that depend on rating values only and those that utilize social relations. For example, the SVD method achieved 0.709 in MAE and 0.925 in RMSE, for the FilmTrust dataset when $d = 5$. In contrast, the worst results in this table in the FilmTrust dataset are produced by PMF, where it achieved 0.714 in MAE and 0.949 in RMSE when $d = 5$.

According to Tables 4.4 and 4.5, the proposed method outperformed the previous approaches. The closest result to our model is EISR. In Table 4.4, the improvement with RMSE and MAE values is around 0.12% and 0.069% respectively. While in Table 4.5 the improvement with RMSE and MAE is around 2.25% and 1.53% for Ciao when $d = 5$.

According to the results above, it can be concluded that the proposed method succeeded to improve prediction accuracy

Table 4.3. Evaluation the performance of non-cold-start users for FilmTrust. boldface indicates the best results and (*) indicates second-top results.

	Metrics	PMF	SVD	SocRec	EISR	MSRA-SVD
d=5	MAE	0.714	0.709	0.674*	-----	0.6214
	RMSE	0.949	0.925	0.878*	-----	0.8351
d=10	MAE	0.735	0.709	0.668*	-----	0.6174
	RMSE	0.968	0.954	0.875*	-----	0.8325

Table 4.4. Evaluation the performance of non-cold-start users for Ciao. Boldface indicates the best results and (*) indicates second-top results.

	Metrics	PMF	SVD	SocRec	EISR	MSRA-SVD
d=5	MAE	0.920	0.784	0.899	0.7279*	0.7274
	RMSE	1.206	1.033	1.183	0.9601*	0.9589
d=10	MAE	1.078	0.788	0.815	0.7276*	0.7263
	RMSE	0.822	1.037	1.076	0.9596*	0.9572

Table 4.5. Evaluation the performance of non-cold-start users for Last.Fm. Boldface indicates the best results and (*) indicates second-top results.

	Metrics	PMF	SVD	SocRec	EISR	MSRA-SVD
d=5	MAE	0.427	0.414	0.422	0.4035*	0.3944
	RMSE	0.537	0.534	0.531	0.5263*	0.5182
d=10	MAE	0.426	0.414	0.422	0.4030*	0.3931
	RMSE	0.535	0.533	0.530	0.5258*	0.5177

4.3.3.2 Impact of different dimension values

Figures 4.6, 4.7, 4.8, 4.9, 4.10 and 4.11 present the impact of changing the number of dimensions (d) on the results of MAE and RMSE metrics for the MSRA-SVD, E-SVD (using explicit relations only) and SVD models. The value of (d) shows the dimension of the latent features of the user-item matrix. Additionally, when the value of (d) increases, more latent features will be included in the dimension reduction methods so that the accuracy is improved. However, when the value of (d) continues to rise, the accuracy will be negatively affected.

Figures 4.6 and 4.7 plot the MAE and RMSE values for the FilmTrust dataset. The MAE and RMSE values fluctuate as (d) is varied for all models. The substantial issue is the impact of social relations on the results. Whereas the highest values (lowest accuracy) are produced by SVD, the results of RMSE are between 0.69 and 0.71 for all dimensions values. Moreover, the best results are produced by the MSRA-SVD

model, which uses explicit and implicit social relations. The results are between 0.6174 when d is 10 and 0.6215 when d is 90 for MAE. For RMSE, the results are between 0.8325 and 0.8344 when d equals 10 and 90 respectively.

Another aspect can be seen in Figures 4.8 and 4.9, namely, the changing of the MAE and RMSE results in the same model. For example, when the dimension value (d) is 10, the RMSE is too high (low accuracy) compared with other values. Furthermore, the accuracy improves as the dimension value increases and it attains its best result when the number of dimensions is 50 in both Figures 4.8 and 4.9. However, in the same figures, the accuracy declines after 50 to approximately 0.7274 in MAE and 0.9584 in RMSE when the dimension value equals 90. Additionally, Figures 4.10 and 4.11 plot the MAE and RMSE values for the Last.Fm dataset. The MAE and RMSE improve as the number of dimensions increases. The best result is attained when the number of dimensions equals 70, where the MAE value is approximately 0.3916 and the RMSE value is about 0.5162 for Figures 4.10 and 4.11 respectively. In contrast, when the number of dimensions increases to more than 70, the accuracy decreases to 0.3926 and 0.5165 in Figures 4.10 and 4.11 respectively.

To sum up, when (d) value rises, more latent features are added and the accuracy improves. Nonetheless, if (d) value passes the threshold, the accuracy decreases because extra features mean noise data.

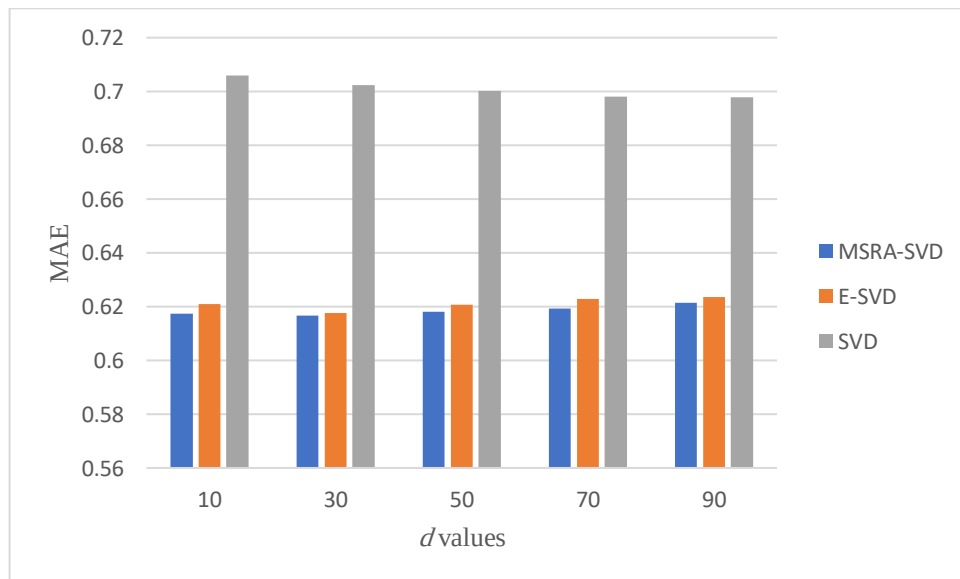


Figure 4.6. MAE results for various numbers of dimensions for FilmTrust dataset.

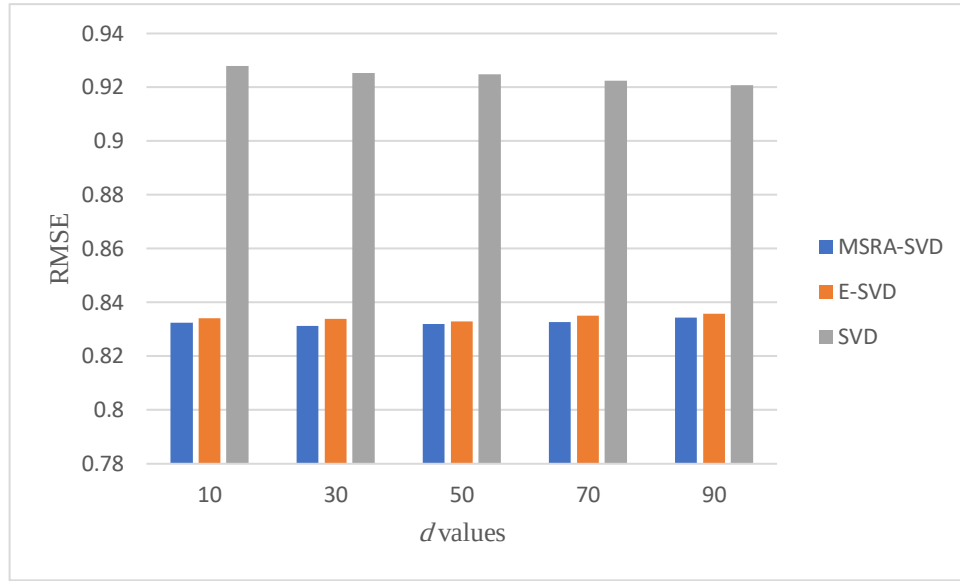


Figure 4.7. RMSE results for various numbers of dimensions for FilmTrust dataset.

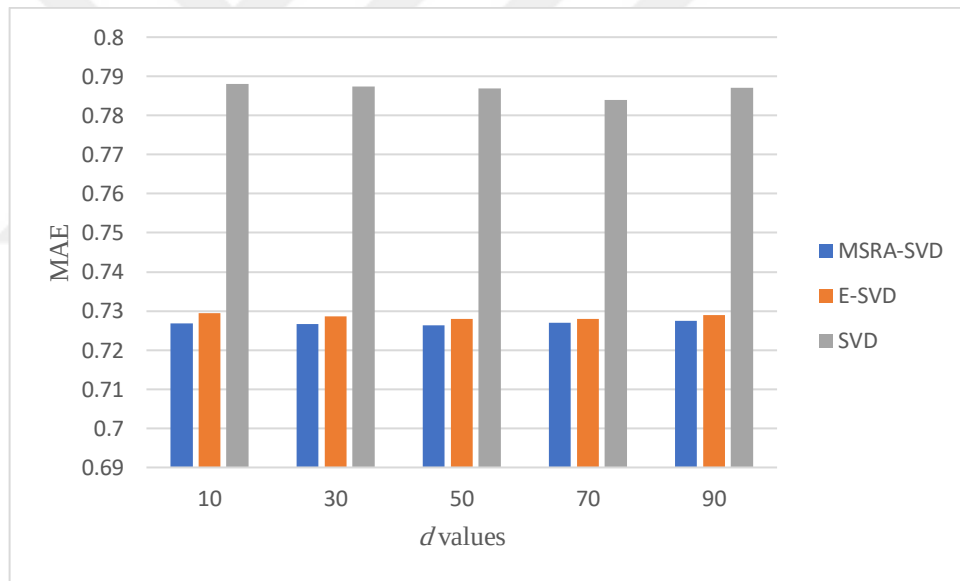


Figure 4.8. MAE results for various numbers of dimensions for Ciao dataset.

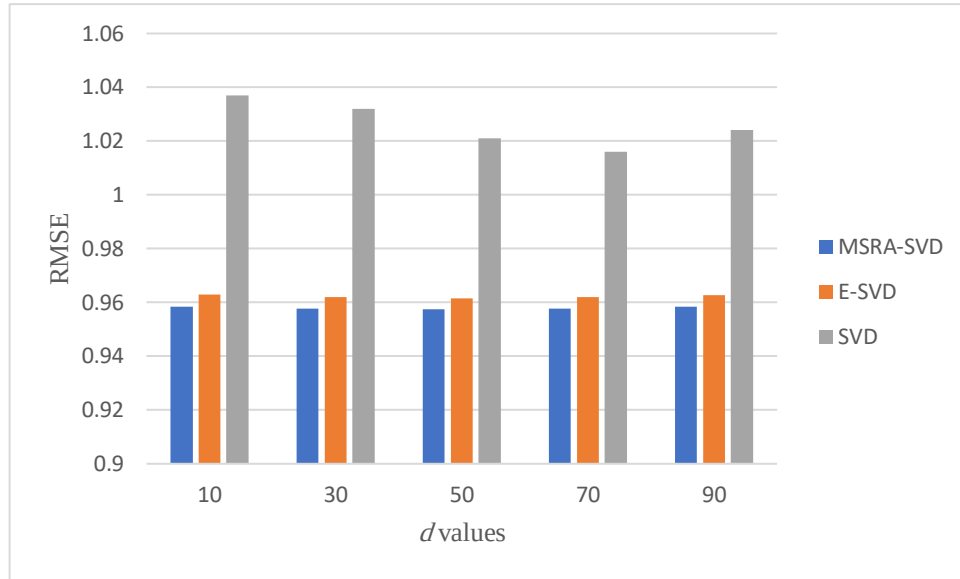


Figure 4.9. RMSE results for various numbers of dimensions for Ciao dataset.

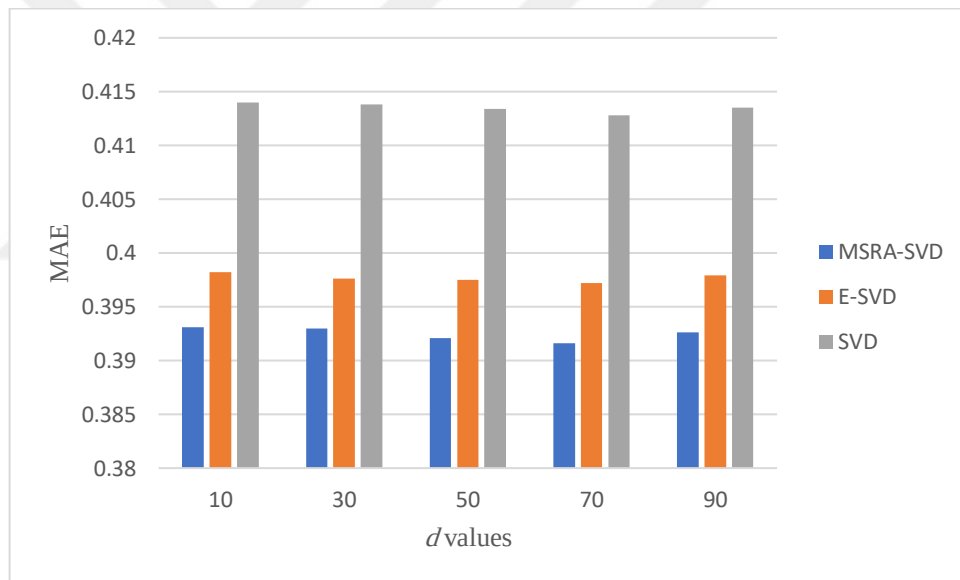


Figure 4.10. MAE results for various numbers of dimensions for Last.Fm dataset.

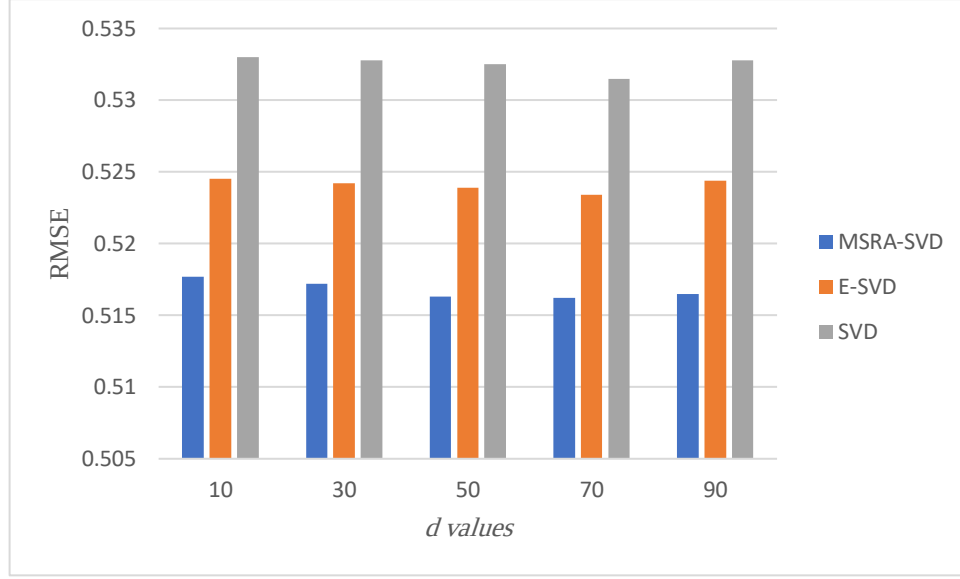


Figure 4.11. RMSE results for various numbers of dimensions for Last.Fm dataset.

4.3.3.3 Impact of changing lambda values

In this section, Figures 4.12-4.17 depict the results of MAE and RMSE for different values of lambda in FilmTrust, Ciao and Last.Fm, respectively.

Lambda values are used to avoid overfitting by $\lambda(b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2)$. We used different values of lambda to select the best value that achieved the best results in terms of MAE and RMSE. It can be seen in Figures 4.12-4.17 that the figures' styles are almost similar for MAE and RMSE results in each dataset. In a specific lambda value, the result is low, whereas in another lambda value, it is high for both metrics. For example, in Figures 4.12 and 4.13 (FilmTrust dataset), the best results (0.6174 in MAE and 0.8325 in RMSE) are achieved when lambda equals 0.001. On the other hand, the worst results are registered when lambda equals 0.4 for both metrics. However, the style of the figure varies in different datasets according to sparsity and users' behavior in a dataset. In other words, the style of Figures 4.12 and 4.13 is different from that of Figures 4.14 and 4.15. For instance, the best results of Figures 4.14 and 4.15 (Ciao dataset) are carried out when lambda equals 0.01. They are 0.7263 and 0.9572 for MAE and RMSE respectively.

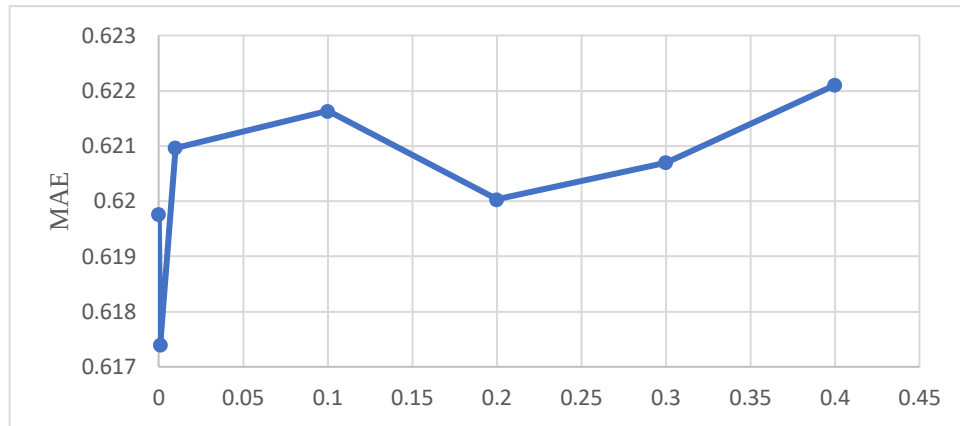


Figure 4.12. Impact of different lambda values on MAE results for FilmTrust.

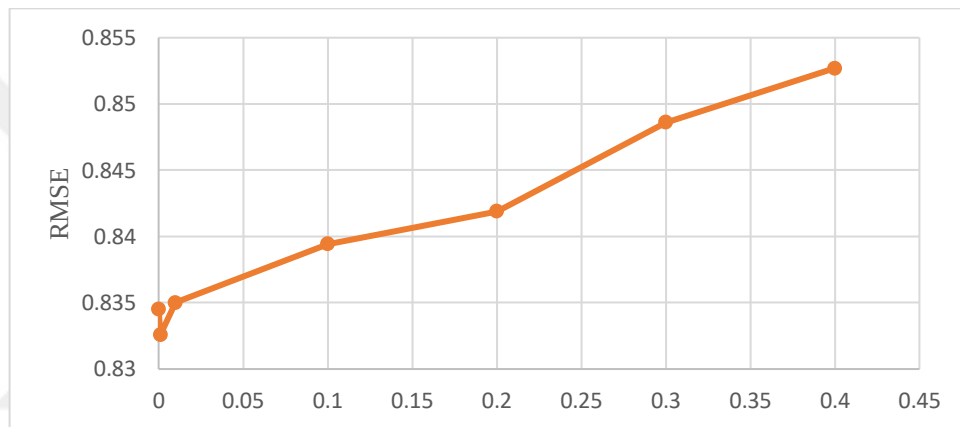


Figure 4.13. Impact of different lambda values on RMSE results for FilmTrust.

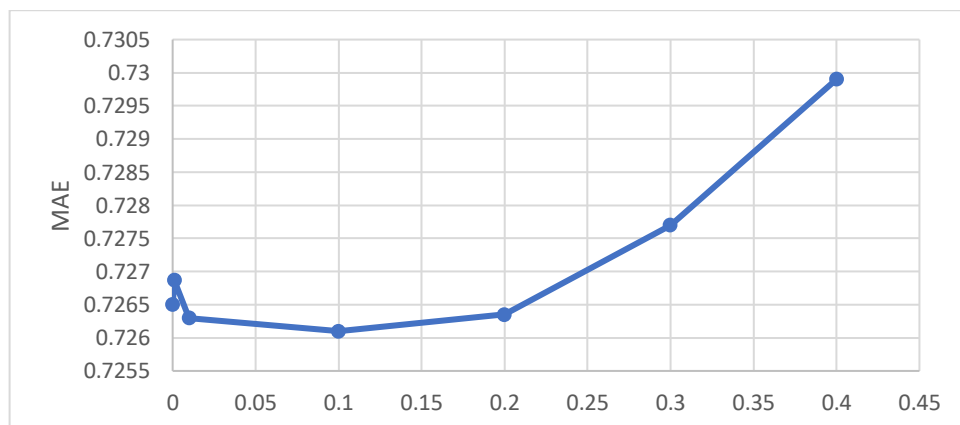


Figure 4.14. Impact of different lambda values on MAE results for Ciao.

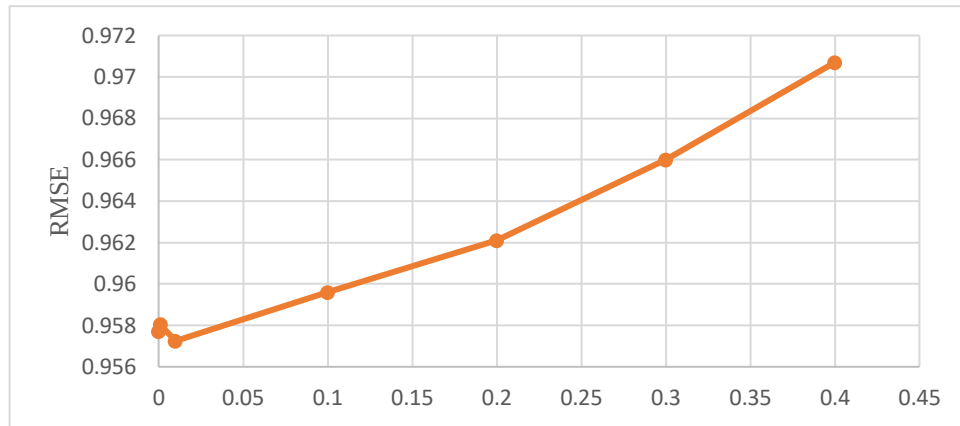


Figure 4.15. Impact of different lambda values on RMSE results for Ciao.

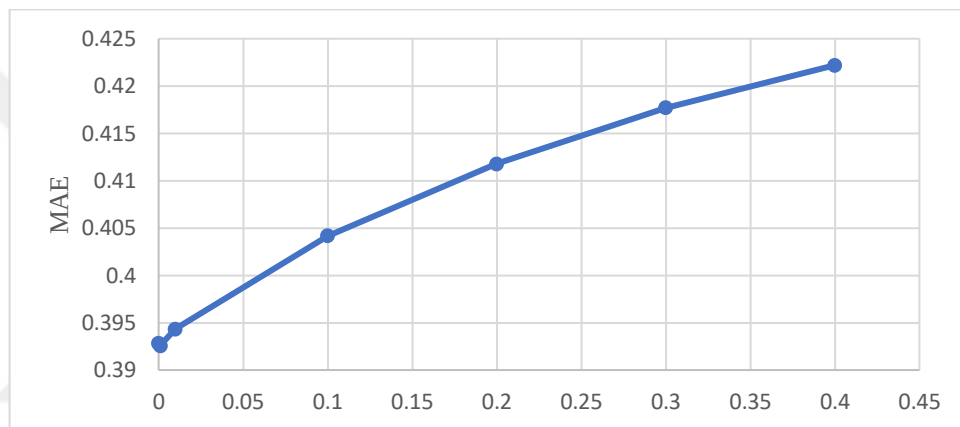


Figure 4.16. Impact of different lambda values on MAE results for Last.Fm.

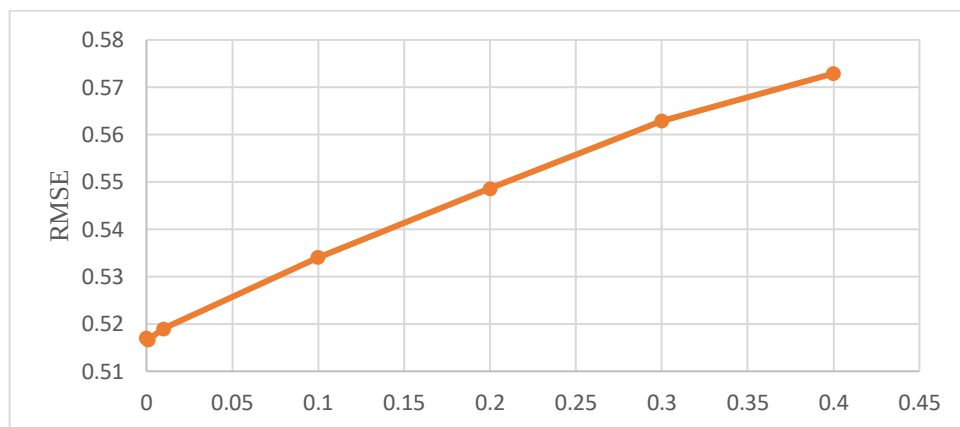


Figure 4.17. Impact of different lambda values on RMSE results for Last.Fm.

4.3.3.4 Impact of changing beta values

Figures 4.18-4.23 show the results of MAE and RMSE for various values of beta, for FilmTrust Ciao and Last.Fm datasets when dimension value (d) is 10.

Beta values are utilized to add similarity values of a particular user's friends to his/her feature vector. Subsequently, the regularization square error is reduced during the training phase. Beta values are used for the similarity of explicit and implicit relations. Figures 4.18-4.23 display the results of MAE and RMSE for different beta values for the datasets of this study. Like lambda figures, every dataset has a different style based on users' behavior in their dataset. Figures 4.18 and 4.19 depict the results of the FilmTrust dataset. In these figures, the results are 0.6243 in MAE and 0.8366 in RMSE when the beta is 0.01. Additionally, when the beta value increases, the results of MAE and RMSE decreases (the accuracy is improved). With the increasing in beta value, the accuracy improves. The best results (0.6174 and 0.8325 for MAE RMSE respectively) are accomplished when beta equals 0.1. However, when the values of beta continue to increase, the results of MAE and RMSE slightly rise to reach the peak when beta equals 0.5. Furthermore, the same scenario can be shown in Figures 4.20-4.23 with different values. For example, in Figures 4.20 and 4.21 (Ciao dataset), the highest results are achieved when beta equals 0.001, where the MAE result is 0.7263 and the RMSE result is 0.9572.

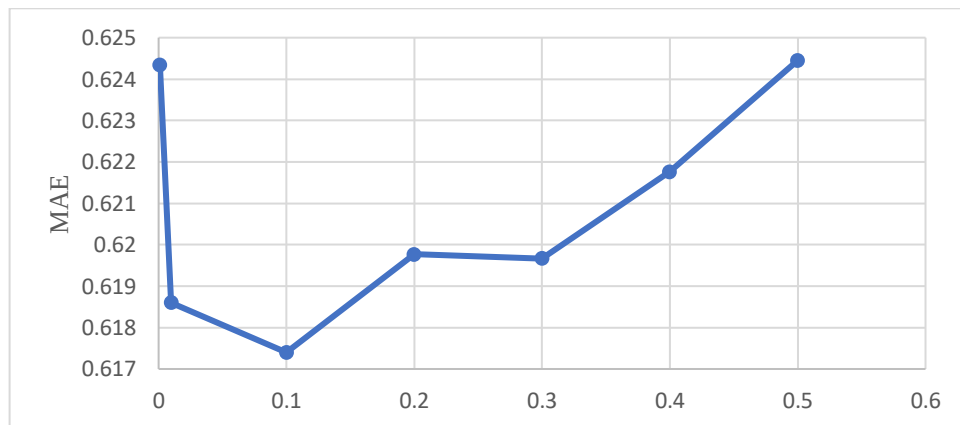


Figure 4.18. Impact of different beta values on MAE results for FilmTrust.

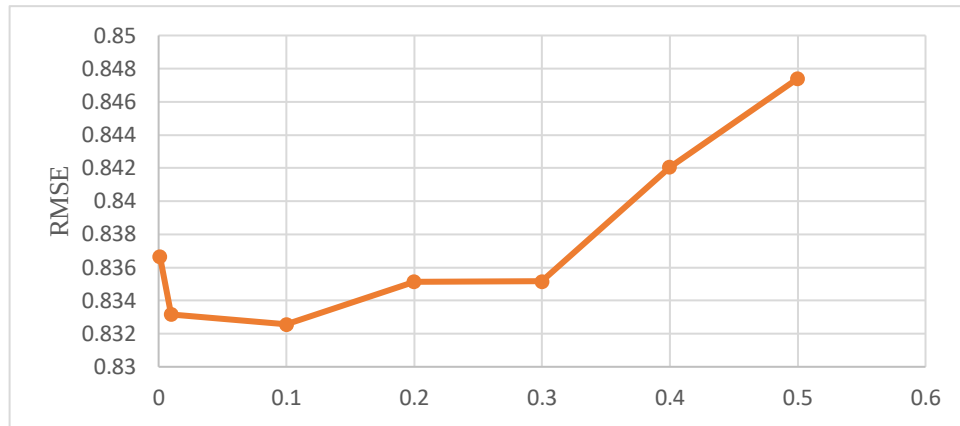


Figure 4.19. Impact of different beta values on RMSE results for FilmTrust.

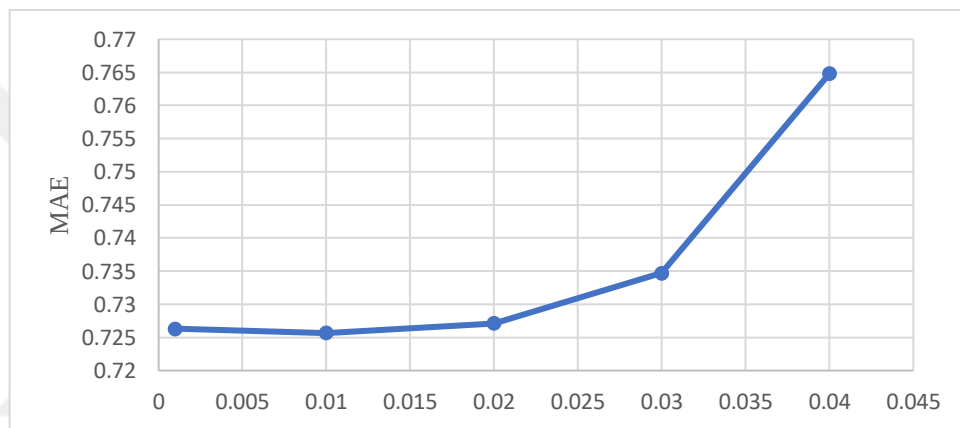


Figure 4.20. Impact of different beta values on MAE results for Ciao.

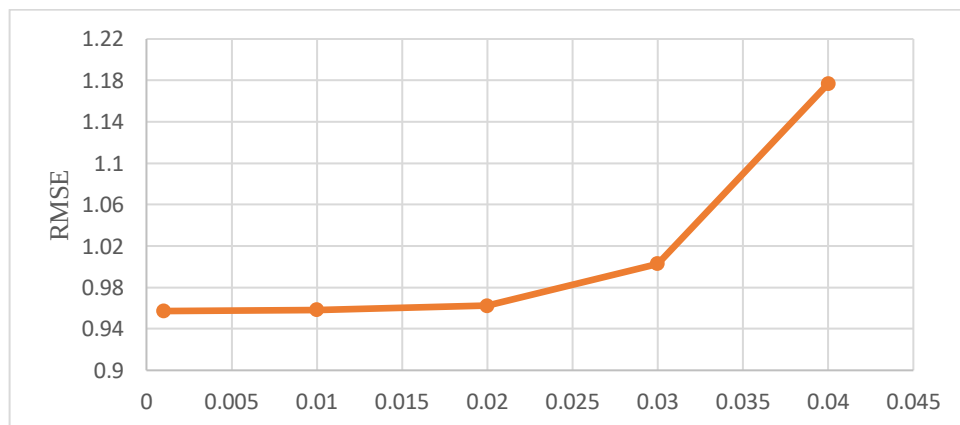


Figure 4.21. Impact of different beta values on RMSE results for Ciao.

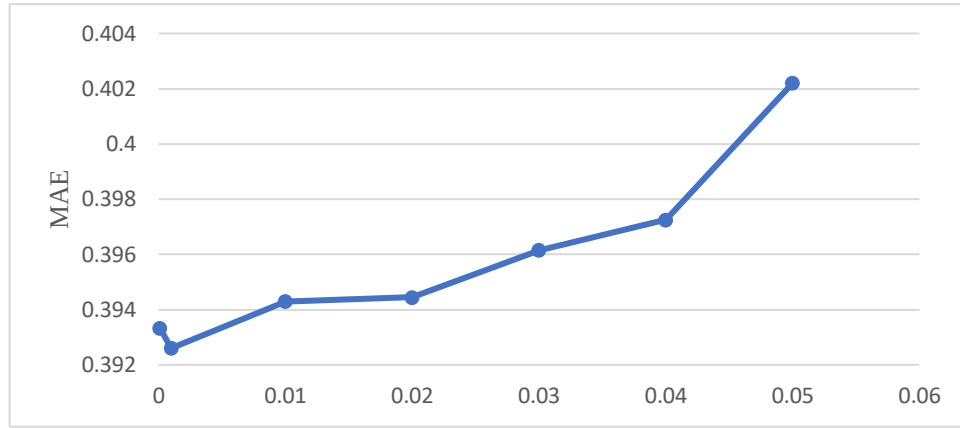


Figure 4.22. Impact of different beta values on MAE results for Last.Fm.

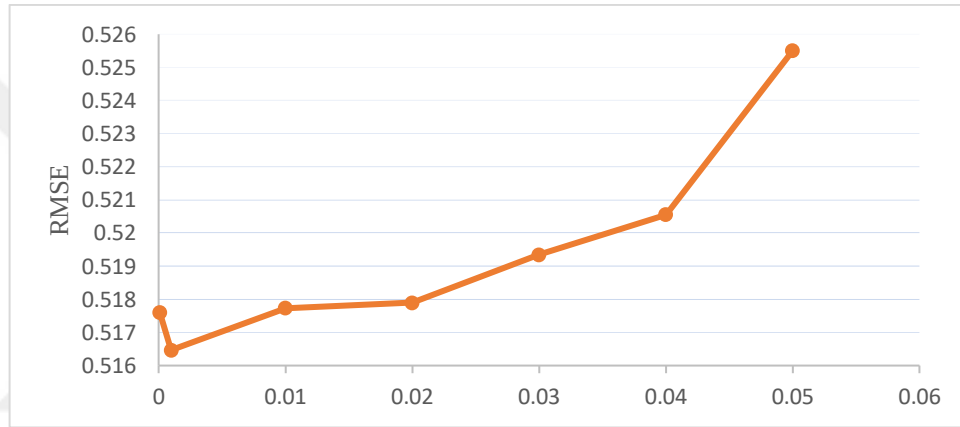


Figure 4.23. Impact of different beta values on RMSE results for Last.Fm.

4.3.3.5 Impact of changing gamma values

Gamma values or learning rate values are used in optimization techniques to tune the parameter values in a specific algorithm, where the parameters are changed at each single iteration in the training process to reach a minimum cost of a loss function. Figures 4.24-4.29 exhibit the results of MAE and RMSE for different gamma values when dimension value (d) is 10. As shown in Figures 4.24 and 4.25 (FilmTrust dataset), the results of MAE and RMSE are almost stable when the gamma values are between 0.01-0.04. However, when the value increases to 0.05, the results increase sharply to reach the worst results, which are 0.6873 and 0.9737 for MAE and RMSE respectively. In Figures 4.28 and 4.29 (Last.Fm dataset) the first and last value of gamma (0.01 and 0.07) present high results (low accuracy) in terms of MAE and

RMSE. However, the best results are registered when the gamma value equals 0.03, where the results are 0.3932 for MAE and 0.5177 for RMSE.

Hence, the results of MAE and RMSE metrics are affected by changing the values of lambda, beta and gamma parameters. According to the dataset, the values of these parameters and results of MAE and RMSE change. The data sparsity and users' behavior play a vital role in affecting the values of MAE and RMSE with different values of these parameters. To find the optimal values of these parameters, for example, we change the values of the lambda parameter and retain the values of the other parameters intact. We then compute the MAE and RMSE results for each value of the lambda parameter. Lastly, we determine the best value of lambda based on the best results of MAE and RMSE. The same scheme can be followed to determine the best values of beta and gamma.

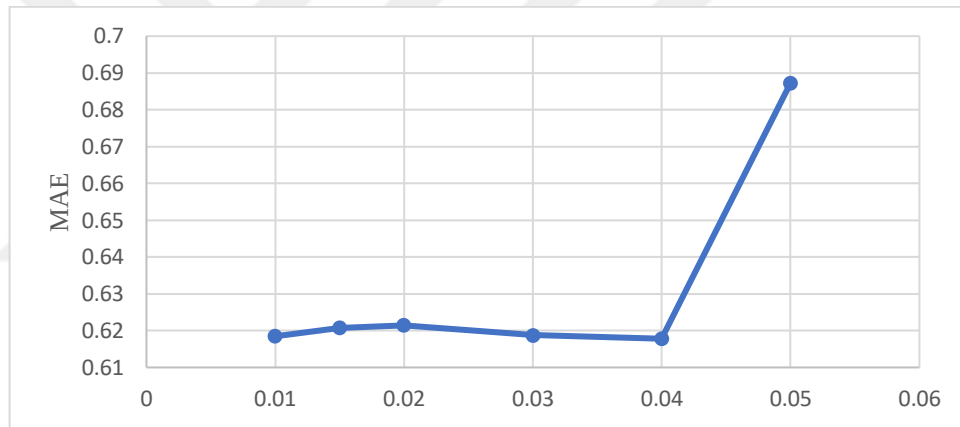


Figure 4.24. Impact of different gamma values on MAE results for FilmTrust.

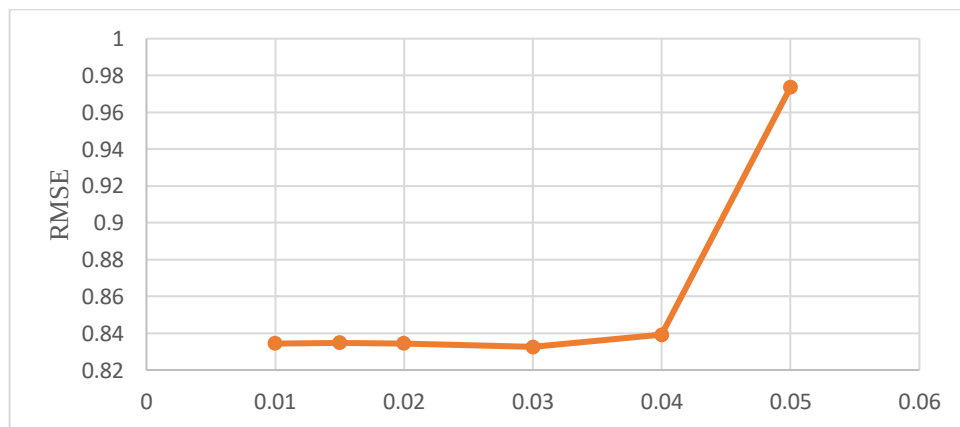


Figure 4.25. Impact of different gamma values on RMSE results for FilmTrust.

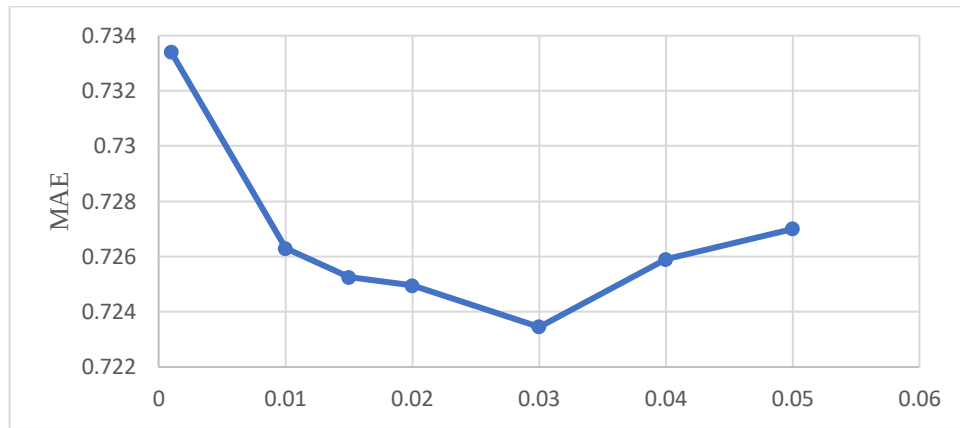


Figure 4.26. Impact of different gamma values on MAE results for Ciao.

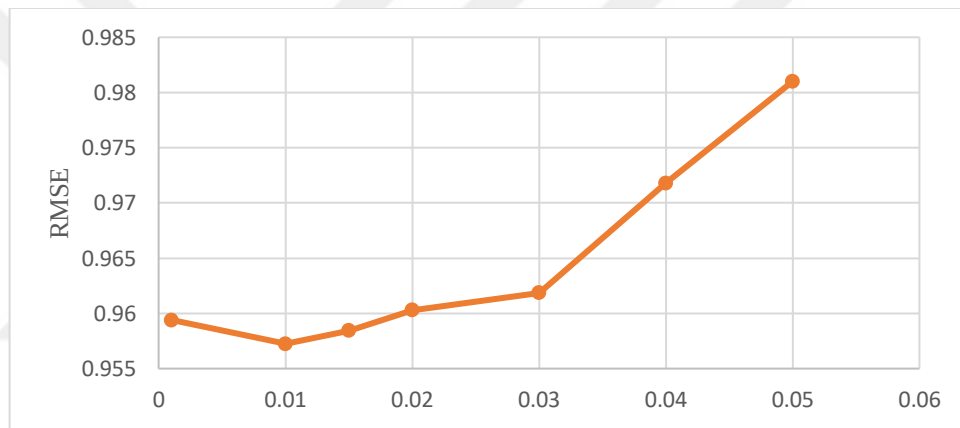


Figure 4.27. Impact of different gamma values on RMSE results for Ciao.

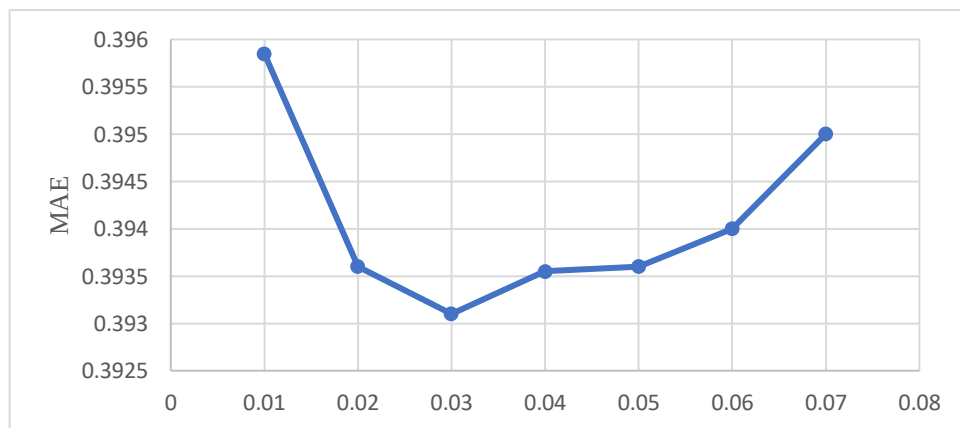


Figure 4.28. Impact of different gamma values on MAE results for Last.Fm.

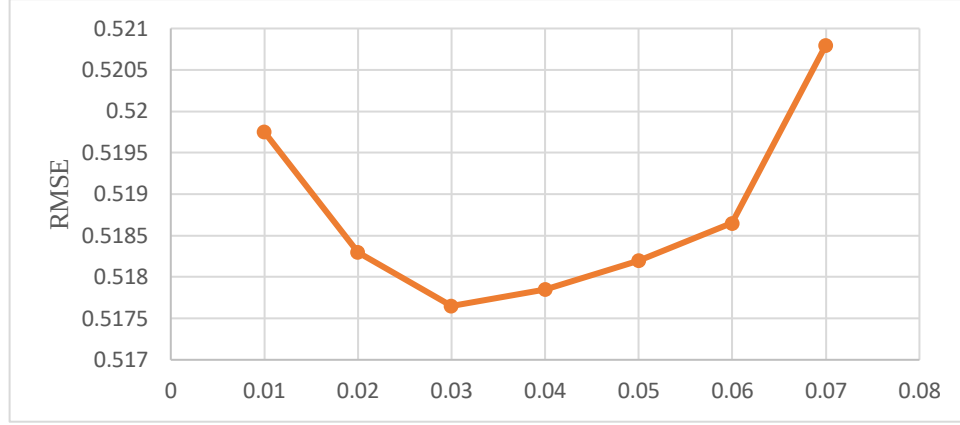


Figure 4.29. Impact of different gamma values on RMSE results for Last.Fm.

4.3.3.6 Decision support accuracy results

In this subsection, another criterion is exploited to assess this model. The evaluation of the proposed method is displayed in terms of precision, recall and F-measure. In these metrics, the high values indicate high accuracy. Tables 4.6, 4.7 and 4.8 depict the results for a different number of Top- n recommended items (10, 15 and 20). The results are displayed when dimension value (d) equals 10. To evaluate the performance of the proposed model, the results are reported for the MSRA-SVD model and the SVD method.

The results in the following tables reveal that the proposed method significantly outperforms the SVD method in all cases. For example, in Table 4.6, MSRA-SVD improved precision by 0.06%, 0.11% and 0.48% compared with the SVD method when $n = 20$ for FilmTrust, Ciao and Last.Fm, respectively. The same scenario can be seen in Tables 4.7 and 4.8, where the proposed model outperforms the SVD method in all n values.

Another aspect can be shown in the following tables. In Table 4.6, the results decrease when the value of n increases. The result is 0.2984 when $n = 10$ and 0.1514 when $n = 20$. The precision metric displays the relevant items from all items recommended by a model. Thus, when the value of n is small (10), the probability of predicting relevant items out of all recommended items is high (the error percentage is low). On the other hand, when n increases, the error percentage increases because the system should provide many items so the precision values decline.

Additionally, in table 4.7, the results of the recall metric increase when the value of n raises. The recall metric measures the percentage of the relevant items in the dataset. Here, the number of relevant items is always constant. But the variable here is the value of n , which is divide over the value of n over the number of all relevant items. Consequently, the recall value increases when the n value increases.

Table 4.6. Precision results.

	Method	Top 10	Top 15	Top 20
FilmTrust	MSRA-SVD	0.2984	0.2015	0.1514
	SVD	0.291	0.196	0.147
Ciao	MSRA-SVD	0.3612	0.2826	0.2290
	SVD	0.3545	0.2758	0.2237
Last.Fm	MSRA-SVD	0.6289	0.4469	0.3376
	SVD	0.6085	0.4276	0.3227

Table 4.7. Recall results.

	Method	Top 10	Top 15	Top 20
FilmTrust	MSRA-SVD	0.6122	0.61655	0.62
	SVD	0.602	0.608	0.6075
Ciao	MSRA-SVD	0.9297	0.9558	0.9674
	SVD	0.9018	0.9265	0.9370
Last.Fm	MSRA-SVD	0.5494	0.5722	0.5731
	SVD	0.5402	0.5646	0.5655

Table 4.8. F-measure results.

	Method	Top 10	Top 15	Top 20
FilmTrust	MSRA-SVD	0.3968	0.3022	0.2429
	SVD	0.393	0.2964	0.2367
Ciao	MSRA-SVD	0.5203	0.4363	0.3703
	SVD	0.5203	0.4251	0.3612
Last.Fm	MSRA-SVD	0.5865	0.5018	0.4248
	SVD	0.5723	0.4866	0.4107

4.4 Summary

Social relations provide vital information that can enhance prediction system accuracy. Rating information and social relations were integrated into the SVD method to overcome the data sparsity problem. This chapter addressed the second research question.

RQ2: How can social relations and explicit rating feedback information be incorporated into the SVD method?

To answer this question, the MSRA-SVD model was proposed to alleviate the data sparsity problem. The model was devised by integrating rating feedback and explicit/implicit social relations. Implicit relations are extracted using the MSRA method. The proposed model was applied to three datasets: FilmTrust, Ciao and Last.fm. Furthermore, two types of evaluation were used: statistic accuracy (MAE and RMSE) and decision support accuracy (precision, recall and f-measure). The results proved that the MSRA-SVD model outperformed other studies in all metrics. These results prove the fact that the proposed method tackled the data sparsity problem successfully and enhanced the prediction accuracy. However, the accuracy can further be enhanced by overcoming the limitations of this model. Therefore, the next chapter is dedicated to achieving this goal.

5. INTEGRATING SOCIAL RELATIONS AND FEEDBACK INFORMATION INTO SVD++ TO ENHANCE PREDICTION ACCURACY

As mentioned in chapter 4, implicit relations involve substantial information that can be utilized to boost prediction accuracy. Several studies have adopted implicit relations. For example, (Reafee et al., 2016) used the resource allocation index algorithm to extract implicit relations. In that study, the authors exploited the implicit friends at the first step after an explicit friend and ignored other steps which may lead to vital information. In the previous chapter, we used the multi-steps resource allocation (MSRA) algorithm to extract such information. The results revealed that the MSRA-SVD method improved the recommendation system in terms of accuracy. However, the drawback of the MSRA-SVD model is ignoring implicit feedback information and using explicit rating information only. Implicit feedback information indicates a user's browsing, clicking, or purchasing of an item regardless of his/her interest in this particular product.

In this chapter, to overcome the data sparsity problem, an improved model is proposed integrating four sources of information: explicit social relations, implicit social relations, rating information and implicit feedback information. This model is the MSRA-SVD++ method. The contribution of this chapter is using the MSRA method in RS and integrating MSRA and SVD++ methods.

The third research question is addressed in this chapter.

RQ3 How can social information and explicit/implicit rating feedback information be incorporated into the SVD++ method?

Four sources of information are used in this chapter. Explicit social relations are real connections between users. Implicit social relations are extracted using the MSRA algorithm. Rating information is the user's ratings that are also called a user-item matrix. Finally, implicit feedback information is the user's behavior when rating an item. This means there is implicit feedback regardless of the rating type (good or bad); otherwise, there is no feedback. The following sections explain the entire process in this model.

5.1 Data Sources

Four sources of information are incorporated in this model, namely, a user-item matrix, implicit feedback, explicit social relations and implicit social relations.

- i- Users' ratings: let assume M is the number of users and N is the number of items. A matrix can be created called a user-item matrix (R) with size $M*N$. r_{ij} is a rating value given to user i and item j .
- ii- Implicit feedback: a user-item matrix (R) with size $M*N$. $\overline{r}_{i,j}$ is one if user i rated item j or otherwise zero.
- iii- Explicit social relations: let assume $G1 = (V, E1)$, where $(u, f) \in E1$ denotes explicit relations, whether direct or indirect.
- iv- Implicit social relations: let assume $G2 = (V, E2)$, where $(u, f) \in E2$ denotes implicit relations (friend-of-friend).

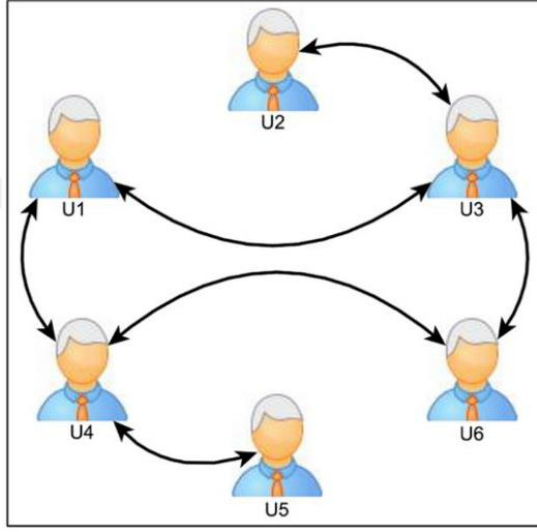
Figure 5.1 (A) shows an example of the user-item matrix, (B) shows implicit feedback (C) depicts explicit social relations and (D) displays implicit relations.

	I1	I2	I3	I4	I5	I6
U1		4	3		5	
U2	3	5		4		
U3				3		5
U4	1		4			3
U5	4	3			4	

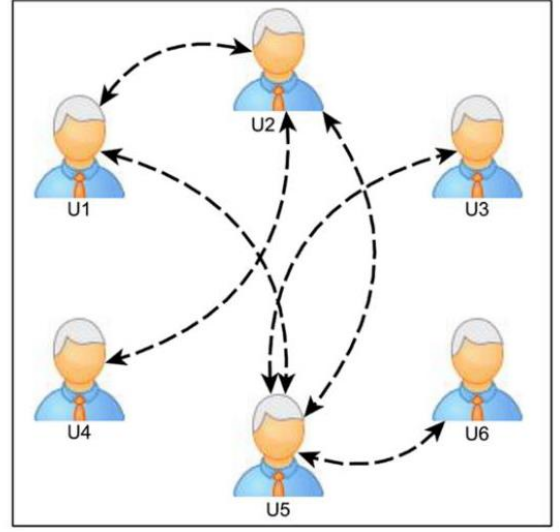
(A)

	I1	I2	I3	I4	I5	I6
U1	0	1	1	0	1	0
U2	1	1	0	1	0	0
U3	0	0	0	1	0	1
U4	1	0	1	0	0	1
U5	1	1	0	0	1	0

(B)



(C)



(D)

Figure 5.1. An example of data sources for the proposed model.

In Figure 5.1 (A), the ratings are available in the dataset, whereas in Figure 5.1 (B), a binary user-item is created; the source of this matrix is users' ratings (Figure 5.1 (A)). The value is one when there is a rating at the same position in Figure 5.1 (A). The value is zero for other positions. Figure 5.1 (C) depicts explicit relations that are directly available in the social dataset. Finally, implicit relations are extracted from explicit social relationships and this is displayed in Figure 5.1 (D).

5.2 Multi-step resource allocation (MSRA)

Social networks encompass a large amount of information, including explicit information such as direct relations, indirect relations and trust relations. Another kind of information is implicit relations. Regarding the implicit relations, the link prediction technique is exploited to predict the missing information in a social network. The Multi-Steps Resource Allocation (MSRA) algorithm was proposed by (Wu and Li,

2014). MSRA is adopted to discover implicit relations of users within the social network. A user not only can be affected by his friends' tastes but also by other users even if there is a distance between them in the network especially when those users are friends of his/her friends. Therefore, MSRA is utilized to apply the link prediction technique and predict missing relations, which may contain vital information. According to the results of chapter 4, MSRA achieved better results than RAI as it could predict implicit relations with an acceptable accuracy ratio (for more details see 4.1.2). MSRA involves a multi-step process, but in this work two-step resource transmission is employed to get at the implicit relations as shown below:

$$s_{xy}^{MSRA1} = \sum_{z \in |r(x) \cap r(y)|} \frac{1}{|r(z)|} \quad (5.1)$$

$$s_{xy}^{MSRA2} = s_{xy}^{MSRA1} + \sum_{z1 \in Z1} \sum_{z2 \in Z2} \frac{1}{k_{z1}} \frac{1}{k_{z2}} \quad (5.2)$$

Where Z and Z1 refer to $r(x) \cap r(y)$, and Z2 to $r(z1) \cap r(y)/Z1$.

5.3 The SVD++ method

SVD++ was produced by (Koren et al., 2008). It is an expansion of the SVD method. The implicit rating information is the data that is considered in SVD++. Recently, many studies have utilized this approach (Jiao et al., 2019; Shi et al., 2020) for its ability to enhance accuracy. For example, assume a 2D user-item matrix (M) wherein the rows p indicate the users and the columns q represent the items. The rating value where the users can rate the items can be expressed as $r_{p,q}$. The main idea of this method is reducing the amount of rating information (user-item matrix), where the new user matrix and the new item matrix are derived by detecting latent features from a user-item matrix. As mentioned in chapter 2, the SVD++ method is expanded of the SVD method. Unlike the SVD method, implicit feedback is taken into consideration in SVD++. The implicit feedback is represented by a binary user-item matrix, where every rating position in the user-item matrix corresponds value one in the binary user-item matrix. (Figure 5.1 (B) shows the binary user-item matrix). Then the prediction value is computed as follows:

$$r_{u,i} = \mu + b_u + b_i + q^T \left(p_u + \frac{1}{\sqrt{|N_u|}} \sum_{j \in N_u} y_j \right) \quad (5.3)$$

$$b_i = \frac{(\sum_{u \in R(i)} r_{u,i} - \mu)}{\lambda_1 + |R_i|} \quad (5.4)$$

$$b_u = \frac{(\sum_{u \in R(u)} r_{u,i} - \mu - b_i)}{\lambda_2 + |R_u|} \quad (5.5)$$

Where μ is the mean value of the entire ratings and b_u and b_i are the observed deviations from the μ of user u and item I respectively. The number of items rated by user u is denoted as $|N_u|$. $|R_u|$ is the number of users that rated a specific item, $|R_i|$ is the number of items that rated by a set of users and y_j is the left orthogonal matrix of the implicit matrix (binary user-item matrix). λ_1 and λ_2 are two regularization parameters.

All the parameters are regularized using the following constraint:

$$\lambda(b_i^2 + b_u^2 + \|p_u\|^2 + \|q_i\|^2 + \|y_j\|^2).$$

Regularization means that parameters' values are restricted to a specific threshold. Additionally, an optimization technique can be exploited with the SVD method to reduce the regularization error. The stochastic gradient descent (SGD) algorithm is a frequent algorithm used in dimension reduction methods. In this algorithm, the training involves several times of loop. The parameters (p_u , q_i , b_i and b_u) are updated each round to reduce the regularization errors as follows:

$$E = \sum_{(u,i) \in m} \left(r_{ui} - \mu - b_u - b_i - q_i^T \left(p_u + \frac{1}{\sqrt{|N_u|}} \sum_{j \in N_u} y_j \right) \right)^2 + \lambda(b_i^2 + b_u^2 + \|p_u\|^2 + \|q_i\|^2 + \|y_j\|^2) \quad (5.6)$$

$$b_u = b_u + \alpha (E - \lambda b_u) \quad (5.7)$$

$$b_i = b_i + \alpha (E - \lambda b_i) \quad (5.8)$$

$$p_u = p_u + \alpha (E q_i - \lambda p_u) \quad (5.9)$$

$$q_i = q_i + \alpha (E p_u - \lambda q_i) \quad (5.10)$$

$$y_j = y_j + \gamma \left(E \left(\frac{q_i}{\sqrt{|N_u|}} \right) - \lambda y_j \right) \quad (5.11)$$

5.4 The MSRA-SVD++ model

In this section, four sources of information are used to feed the matrix factorization technique. They are rating information that includes explicit and implicit feedback information, explicit social relations and implicit social relations. The following steps illustrate this model in detail:

Step 1: From the social data, a user-user matrix is created. The similarity between each pair of explicit friends is computed via the Pearson correlation coefficient (PCC) as follows:

$$sim(i, f) = \frac{\sum_{h=1}^n (r_{i,jh} - \bar{r}_i)(r_{f,jh} - \bar{r}_f)}{\sqrt{\sum_{h=1}^n (r_{i,jh} - \bar{r}_i)^2} \sqrt{\sum_{h=1}^n (r_{f,jh} - \bar{r}_f)^2}} \quad (5.12)$$

$r_{i,j}$ is the rating of item j by user I and $r_{f,j}$ represents the rating of item j by user f and f indicates a friend of I whether explicit or implicit. $\bar{r}_x$ refers to the average rating of a user. N is the number of common items between users I and f . In this method, the similarity value between I and f ranges from $[-1, 1]$. Therefore, the higher value means that the two users are more similar. The similarity values are placed in the created matrix.

Step 2: Acquiring the implicit relations via the MSRA algorithm using Equations (5.1) and (5.2). A user-user matrix for implicit social relations. The similarity between each pair of implicit friends is calculated via the PCC method (5.13).

$$sim(i, f^*) = \frac{\sum_{h=1}^n (r_{i,jh} - \bar{r}_i)(r_{f^*,jh} - \bar{r}_{f^*})}{\sqrt{\sum_{h=1}^n (r_{i,jh} - \bar{r}_i)^2} \sqrt{\sum_{h=1}^n (r_{f^*,jh} - \bar{r}_{f^*})^2}} \quad (5.13)$$

f^* denotes implicit friends. The similarity values are placed in the implicit friends' matrix.

Step 3: From the rating file, a user-item matrix is created. Next non-cold-start users are separated according to their ratings, where only users with more than five ratings are kept.

Step 4: From the rating data, a user-item binary matrix is created. It involves ones or zeros; ones for the rated positions and zeros for unrated positions.

Step 5: All these sources are employed in SVD++ to produce the new model. The computed similarity is utilized by the optimization steps to reduce the regularization error using the optimization technique. The stochastic Gradient Descent (SGD) optimization algorithm is used in this study. Moreover, as the similarity is calculated for users only, the regularization error is divided into two equations. The first equation is to reduce the regularization error for users' factors. This is achieved by adding the similarity values. The second equation is to reduce the regularization error for items' factors. Figure 5.2 shows the impact of explicit and implicit relations on users.

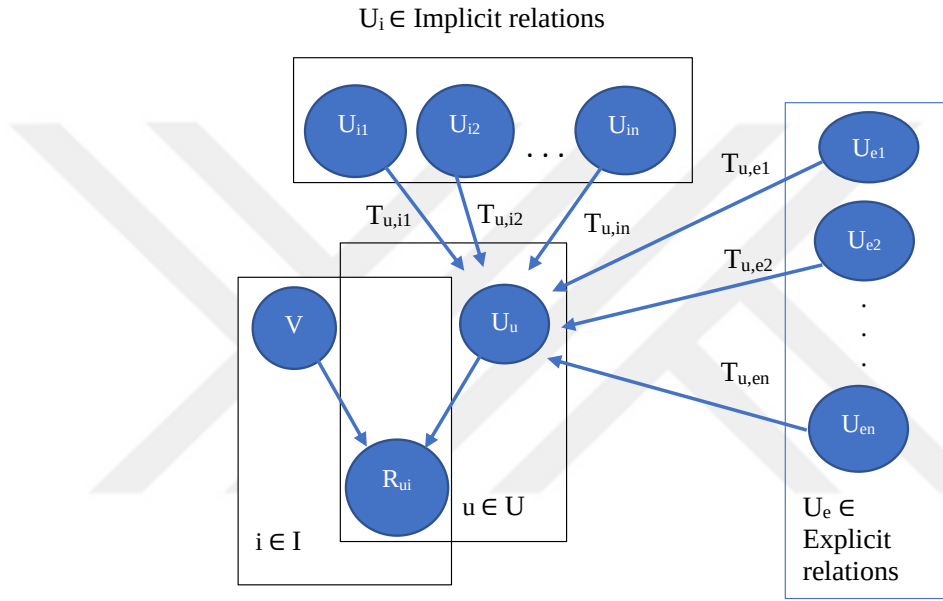


Figure 5.2. The impact of explicit and implicit relations in the SVD++ method.

Accordingly, the regularization error is modified by integrating the results of explicit and implicit similarity to produce Equations (5.14) and (5.15).

$$\begin{aligned} \frac{\partial E}{\partial U_u} = & \sum_{(u,i) \in K} \left(r_{u,i} - \mu - b_i - b_u - q^t \left(p_u + \frac{1}{\sqrt{|N_u|}} \sum_{j \in N_u} y_j \right) \right)^2 + \lambda (b_u^2 + b_i^2 + \\ & \|p\|^2 + \|q\|^2 + \|y_j\|^2) + \beta_e \sum_{i=1}^u \sum_{f \in F(i)} \text{sim}(i, f) \|p_i - p_f\|_{fro}^2 + \\ & \beta_i \sum_{i=1}^u \sum_{f^* \in F^*(i)} \text{sim}(i, f^*) \|p_i - p_{f^*}\|_{fro}^2 \end{aligned} \quad (5.14)$$

$$\begin{aligned} \frac{\partial E}{\partial V_i} = & \sum_{(u,i) \in m} \left(r_{ui} - \mu - b_u - b_i - q^t \left(p_u + \frac{1}{\sqrt{|N_u|}} \sum_{j \in N_u} y_j \right) \right)^2 + \lambda (b_i^2 + b_u^2 + \\ & \|p\|^2 + \|q\|^2 + \|y_j\|^2) \end{aligned} \quad (5.15)$$

where β_e and β_i are constant values that are used to control the impact degrees of explicit and implicit relations respectively, $F(i)$ and $F^*(i)$ denote the numbers of user i 's explicit and implicit friends respectively, and $\text{sim}(x,y)$ refers to the similarity value between users x and y . λ is the constant value that determines the degree of the constraint. These equations are utilized to find the best value for p_u , q_u , b_u , b_i and y_j vectors. The following equations show the updating values after each loop:

$$p_u = p_u + \gamma \left(\frac{\partial E}{\partial U_u} q_i - \lambda p_u \right) \quad (5.16)$$

$$q_i = q_i + \gamma \left(\frac{\partial E}{\partial V_i} \left(p_u + \frac{1}{\sqrt{|N_u|}} \sum_{j \in N_u} y_j \right) - \lambda q_i \right) \quad (5.17)$$

$$y_j = y_j + \gamma \left(\frac{\partial E}{\partial U_u} \left(\frac{q_i}{\sqrt{|N_u|}} \right) - \lambda y_j \right) \quad (5.18)$$

$$b_u = b_u + \gamma \left(\frac{\partial E}{\partial U_i} - \lambda b_u \right) \quad (5.19)$$

$$b_i = b_i + \gamma \left(\frac{\partial E}{\partial V_i} - \lambda b_i \right) \quad (5.20)$$

γ is the learning rate.

Step 6: After optimizing the factors above, now the SVD++ method is applied to compute the missing predictions using Equation (5.3).

Step 7: The performance evaluation is done by comparing the computed values in the training part with the actual values in the test part. The accuracy metrics (MAE, RMSE, precision, recall and f-measure) are used to judge the proposed model. Figure 4.5 provides an overview of the proposed model.

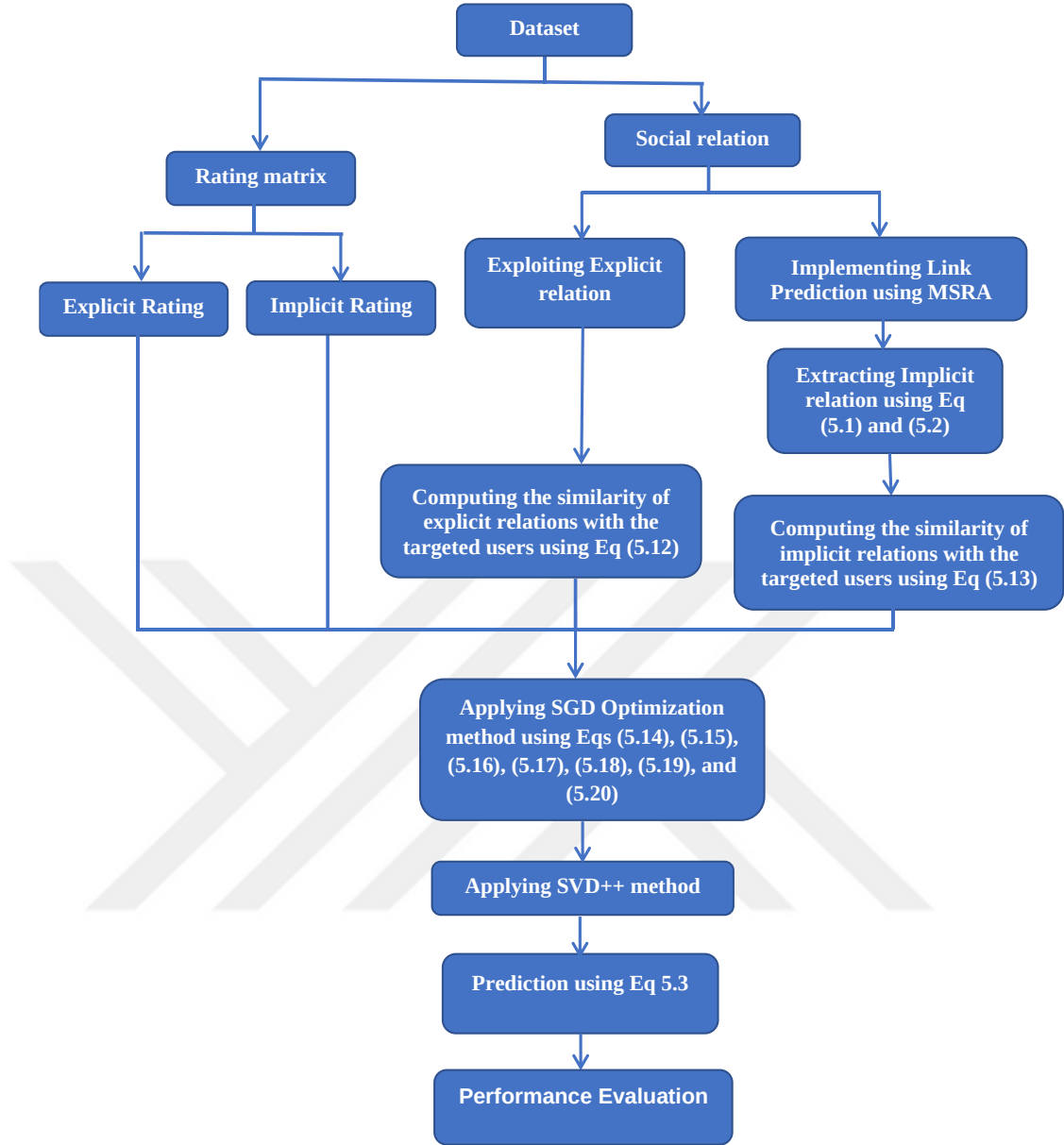


Figure 5.3. An Overview of MSRA-SVD++.

5.5 Experimental Preparation

In this section, the datasets used in this study are described and the evaluation metrics are presented.

5.5.1 Description of datasets

Three real-world datasets are tested in this model: FilmTrust, Ciao and Last.Fm. The description was given in chapter 3. Table 5.1 shows the statistics of these datasets.

Table 5.1. Statistics of datasets.

Dataset	Last.FM	Ciao	Film Trust
# users	1122	6767	1508
# items	18745	22229	2071
# rating	55,140	185759	35497
% Density	0.0026	0.0012	1.14
# explicit relation	11064	111780	1853
# implicit relation using MSRA	177987	610710	3138

5.5.2 Experimental setup

The implementation of this study depends on some parameters for realizing the optimal results. These parameters are listed in Table 5.2.

Table 5.2. Parameter setup.

Dataset	Last FM	Ciao	Film Trust
β_e	0.002	0.001	0.005
β_i	0.002	0.001	0.005
α	0.01	0.006	0.01
λ_e and λ_i	0.001	0.0001	0.001
λ_1 and λ_2	10, 25	4, 8	7, 10
d	5, 10	5, 10	5, 10
Iterations	5	5	5

To evaluate the proposed methods, state-of-the-art approaches are benchmarked against our study including RSTE (Ma et al., 2009), PMF (Mnih and Salakhutdinov, 2008), SVD++ (Koren, 2008), EISR (Reafee et al., 2016) and TrustMF (Bo et al., 2013).

The most similar studies in the literature of this study are TrustMF, EISR and our previous model (MSRA-SVD). Thus, for equitable comparison, the proposed models were tested under the same conditions of the aforementioned studies such as the number of iterations, explicit social relations, the dimensional reduction value and the number of users. Table 5.2 shows the parameters setup.

5.5.3 Results and analysis

The 5-fold Cross-validation was adopted in this experiment. The datasets were randomly divided into five folds: four-folds are used for training (80%) and one for testing (20%). To guarantee testing all the data, each execution is performed five times and the average was computed to yield the outcome of each implementation. Moreover, the experiment was implemented several times and the mean was calculated to display the eventual result. The following subsections demonstrate the results of statistic accuracy metrics and the impact of changing dimension, beta and learning rate values. Finally, decision support accuracy is utilized as well in the last subsection to evaluate the performance of our proposed method.

5.5.3.1 Statistic accuracy metrics

Tables 5.3, 5.4 and 5.5 depict the accuracy of the results for non-cold-start users in Ciao, Last.fm and FilmTrust datasets respectively. The accuracy results were computed through the MAE and RMSE metrics. For both metrics, high values indicated low accuracy. If, on the other hand, the values are close to zero, the results are highly accurate. Two results were displayed in both tables when the number of dimensions (d) is 5 and 10. The proposed models were compared with state-of-the-art studies. These include PMF and SVD++ that used rated information only and the other studies that utilized social relations as well, such as EISR, SocReg, TrustMF and RSTE. The best results are marked using Boldface and the symbol (*) indicates the second-best results. All these studies are shown in Tables 5.3-5.5.

Table 5.3 shows the influence of social relations on improving the prediction process. Comparing the results of models that do not use social sources with ones that do use them shows the vital role of social relations in boosting accuracy.

According to Table 5.3, the proposed model outperformed all the previous approaches. The closest result is those of our previous model (MSRA-SVD) in terms of MAE and RMSE for Ciao and Last.fm datasets. Comparing the second top method and the proposed method, the latter improves the results by approximately 0.63% and 1.11% for MAE and RMSE respectively when d is 5. Moreover, the improvement is about 0.58% for MAE and 1% for RMSE when d equals 10.

The same aspect can be seen in Table 5.4, where the proposed again yielded the best results, and the second top approach is MSRA-SVD. The proposed method enhances the results by around 13.6% in terms of MAE and 11.57% in terms of RMSE when d is 5. Furthermore, the improvement is 13.4% and 11.6% for MAE and RMSE respectively when d is 10.

Finally, Table 5.5 depicts the results of MAE and RMSE for the FilmTrust dataset. Once again, the proposed model outperformed the other approaches in MAE and RMSE metrics for d values 5 and 10. The second top result is achieved jointly by MSRA-SVD and TrustMF. In terms of MAE, MSRA-SVD is the second top method, whereas in terms of RMSE, TrustMF is the second top method. Additionally, the improvement of the proposed method compared with the second top method is around 2.73% for MAE and 1% for RMSE when d equals 5. The proposed method also ameliorates the results by approximately 2.29% for MAE and 2.39% for RMSE when d is 10.

To sum up, the proposed method enhanced the results in terms of accuracy and overcame the data sparsity problem by exploiting four sources of information: explicit social relations, implicit social relations, explicit rating values and implicit rating feedback. All the results confirm that the proposed method succeeded to enhance the recommendation system performance.

Table 5.3. Evaluation the performance of non-cold-start users for Ciao. Boldface marks best results and (*) indicates the second-best results.

	Metrics	PMF	SVD++	RSTE	EISR	Trust MF	MSRA-SVD	MSRA-SVD++
d=5	MAE	0.920	0.75	0.767	0.728	0.742	0.7274*	0.7228
	RMSE	1.206	1.010	1.020	0.960	0.983	0.9589*	0.9482
d=10	MAE	1.078	0.7488	0.763	0.728	0.753	0.7263*	0.7221
	RMSE	0.822	1.001	1.013	0.959	1.014	0.9572*	0.9476

Table 5.4. Evaluation the performance of non-cold-start users for Last.fm. Boldface marks best results and (*) indicates the second-best results.

	Metrics	PMF	SVD++	RSTE	EISR	Trust MF	MSRA-SVD	MSRA-SVD++
d=5	MAE	0.427	0.413	----	0.403	----	0.3944*	0.3405
	RMSE	0.537	0.5326	----	0.526	----	0.5182*	0.4582
d=10	MAE	0.426	0.4123	----	0.403	----	0.3931*	0.3403
	RMSE	0.535	0.5321	----	0.525	----	0.5177*	0.4575

Table 5.5. Evaluation the performance of non-cold-start users for FilmTrust. Boldface implies best results and (*) indicates the second-best results.

	Metrics	PMF	SVD++	RSTE	EISR	Trust MF	MSRA-SVD	MSRA-SVD++
d = 5	MAE	0.714	0.6215	0.628	-----	0.631	0.6214*	0.6044
	RMSE	0.949	0.8364	0.810	-----	0.810*	0.8351	0.802
d=10	MAE	0.735	0.6221	0.640	-----	0.631	0.6174*	0.6032
	RMSE	0.968	0.8352	0.835	-----	0.819*	0.8325	0.7994

5.5.3.2. Impact of different dimensions values

Figures 5.4, 5.6 and 5.8 display the results of MAE for different dimension values (5, 10, 30, 50, 70 and 90) in Ciao, Last.fm and FilmTrust datasets respectively. Moreover, Figures 5.5, 5.7, and 5.9 depict the results of RMSE for the same datasets. The

dimension indicates several latent features that are extracted from a user-item matrix to display it in a few features. When the number of dimensions increases, more features are added to display a matrix. However, adding many features destroys the matrix by adding some noise. Not all the data is pure. Some users rate items randomly, so it is very hard to follow their behavior or subsequently find their preferences.

Each figure shows the results of four different methods, MSRA-SVD++, MSRA-SVD, E-SVD++ and SVD++ methods. The first method is our model proposed in this chapter, whereas MSRA-SVD is our previous model (chapter 4). The E-SVD++ model integrates explicit social relations and the SVD++ method without implicit social relations. Finally, the SVD++ method is a pure method that uses a user-item matrix as a unique source. These figures show the results of MAE and RMSE metrics. When the result is close to zero (small values), it means high accuracy. High values refer to less accuracy.

As shown in all figures that the MSRA-SVD++ yields best results compared to the other methods in all dimension values for MAE and RMSE metrics. For example, in Figures 5.4 and 5.5, when dimension value (d) equals 5, the proposed method achieved 0.7228. However, the closest study to our method (E-SVD++) achieved 0.7244 in terms of MAE. The same superiority can be seen in all dimension values. The highest values (less accuracy) are displayed by the SVD++ method. The impact of social relations is highlighted in these results. These relations play a vital role to boost prediction accuracy.

Another aspect that can be noted in Figures 5.4 and 5.5 is that the MAE and RMSE results improve when the dimension value (d) increases. They decrease from 0.7228 when $d=5$ to reach 0.7198 when $d=50$. The best results in Figures 5.4 and 5.5 are attained when d equals 50. Afterward, the results increase (accuracy decreases) when d increases to reach 0.723 when d equals 70 and 0.7235 when d is 90. The best results vary from one dataset to another according to data sparsity and user behavior. In the Ciao dataset (Figures 5.4 and 5.5), the best results are achieved when d equals 50. In the Last.Fm dataset (Figures 5.6 and 5.7), the best results are obtained when d equals 70. They are 0.3394 for MAE and 0.4563 for RMSE. Whereas in the FilmTrust dataset (Figures 5.8 and 5.9), the best results are attained when d equals 10.

In this last dataset (Figures 5.8 and 5.9), although the MSRA-SVD model makes use of social relations, its results are close to those of the SVD++ method. Moreover, the SVD++ results are closer to all models in these figures, unlike the difference between them in Ciao and Last.Fm datasets (Figures 5.4 - 5.7). The reason for this scenario is that the FilmTrust dataset has a high-density ratio of 1.14%, unlike Ciao and Last.Fm, which are 0.0012 and 0.0026 respectively. Therefore, the impact of social relations reduces for a high-density dataset. Consequently, the results of using social relations will be close to a model that uses only rating information.

Hence, the proposed method achieved the best results in terms of MAE and RMSE in all dimension values. Moreover, for different dimension values, different results are obtained due to the latent features used to display the matrix. When we increase the features to a specific number, a matrix can be displayed more accurately. Finally, social relations have a pivotal role to enhance prediction accuracy for a dataset that has high sparsity. They, however, have less influence in dataset with high density.

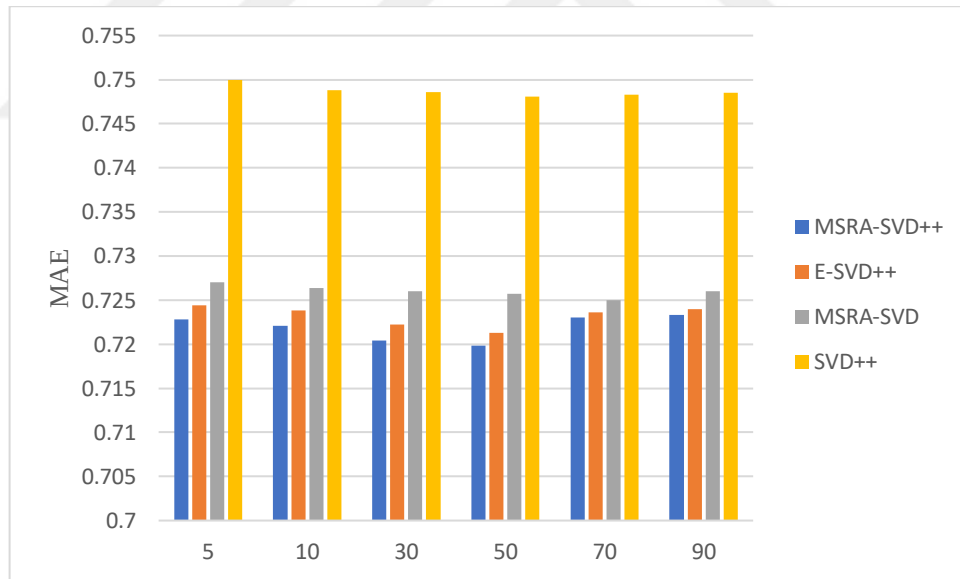


Figure 5.4. MAE results for various numbers of dimensions in Ciao.

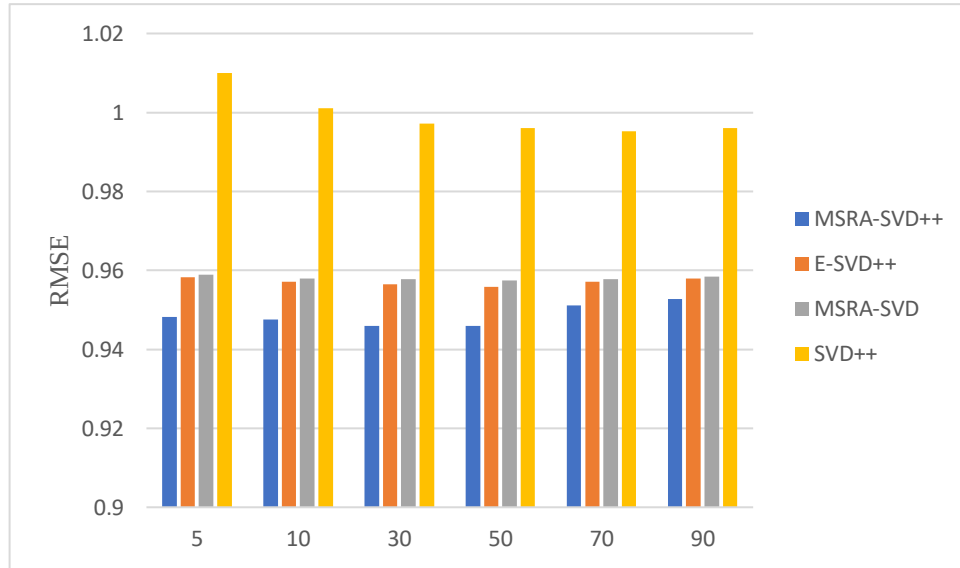


Figure 5.5. RMSE results for various numbers of dimensions in Ciao.

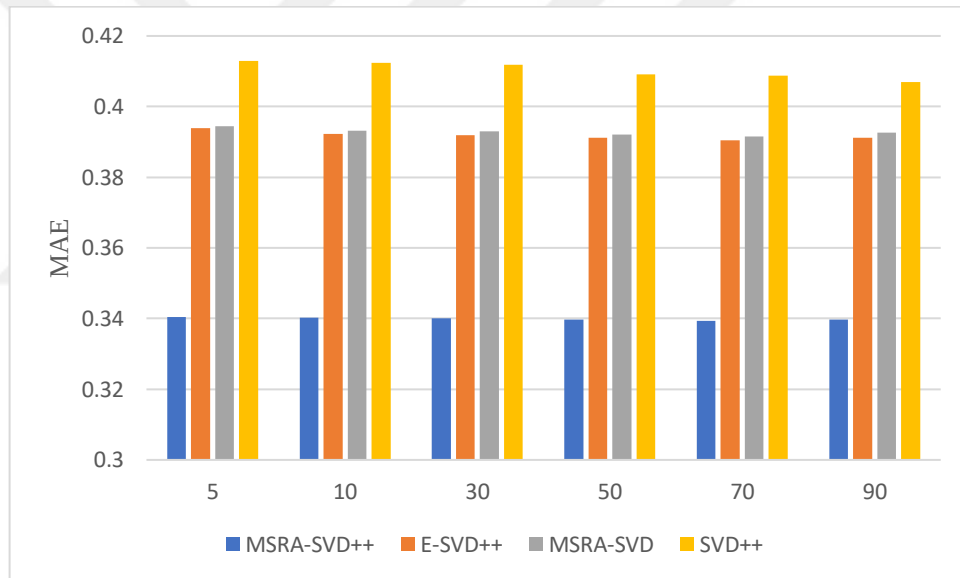


Figure 5.6. MAE results for various numbers of dimensions in Last.Fm.

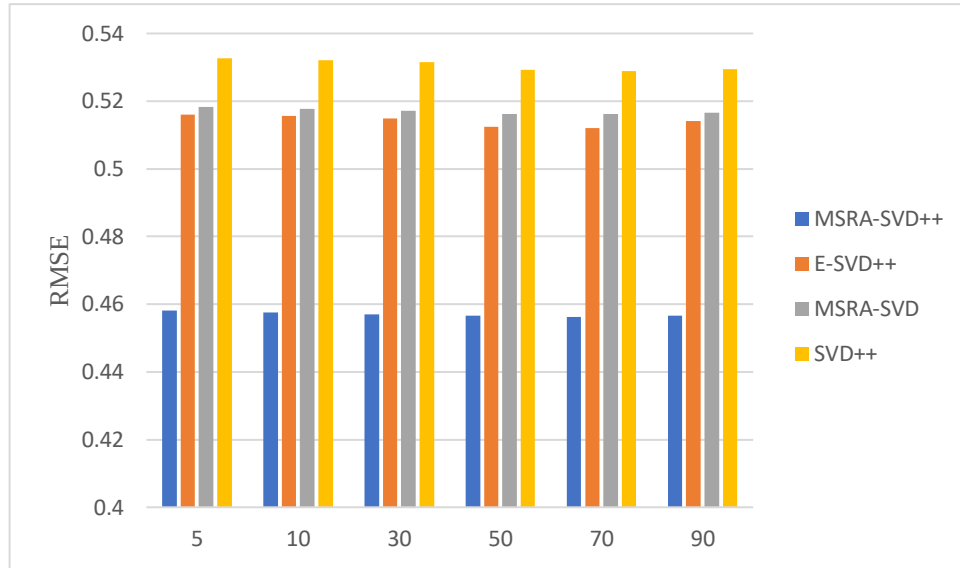


Figure 5.7. RMSE results for various numbers of dimensions in Last.Fm dataset.

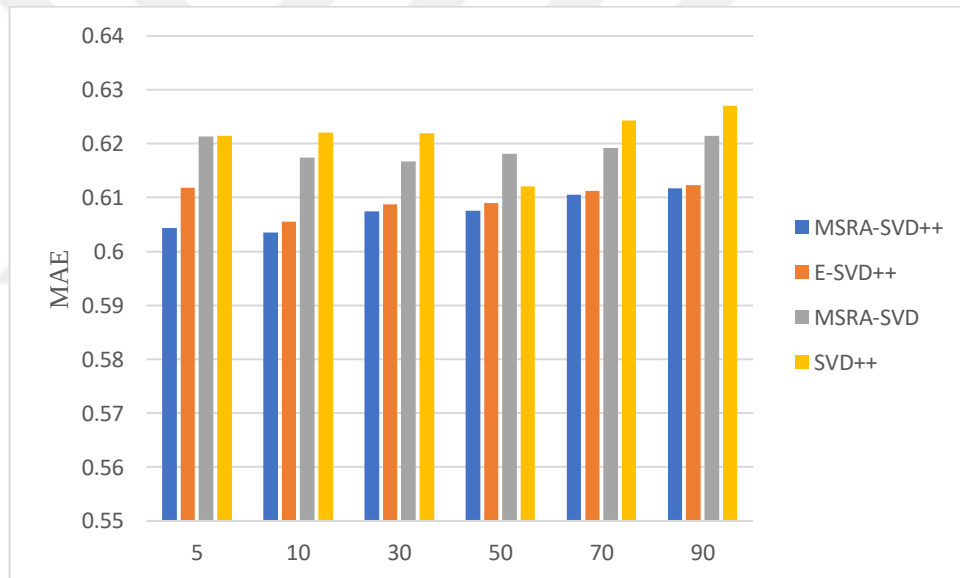


Figure 5.8. MAE results for various numbers of dimensions in FilmTrust.

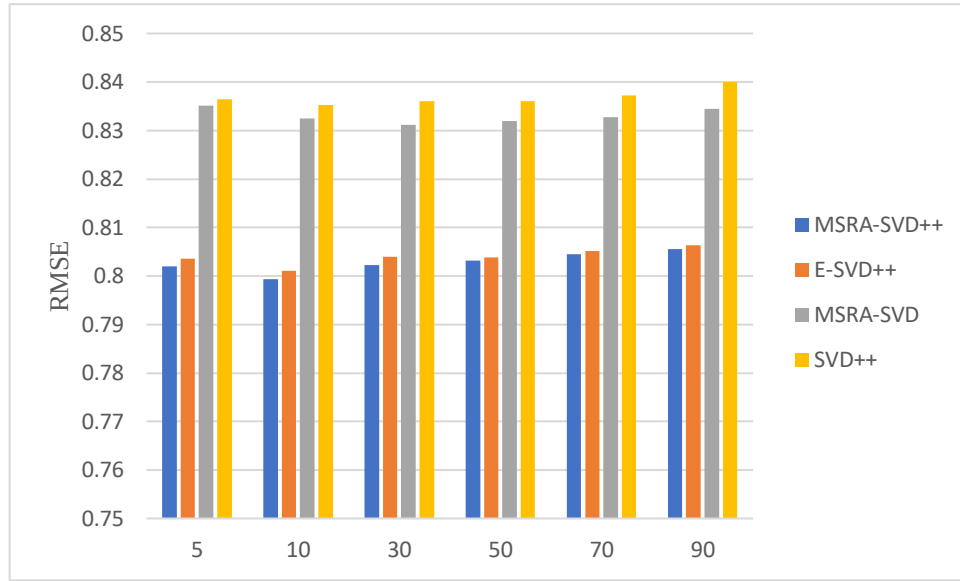


Figure 5.9. RMSE results for various numbers of dimensions in FilmTrust.

5.5.3.3 Impact of changing beta values

Figures 5.10, 5.12 and 5.14 display the results of the MAE metric for some beta values. Figures 5.11, 5.13 and 5.15, on the other hand, depict the results of the RMSE metric for the same beta values used in MAE metric in Ciao, Last.Fm and FilmTrust datasets respectively. It is known that low values of MAE and RMSE indicate high accuracy and vice versa. Beta values are utilized along with similarity values among users. More specifically, beta values control the degree of employing users' similarity to reduce the regularization error during the training phase. High beta values refer to using a high similarity ratio. In this chapter, a set of beta values are used. They are (0.0001, 0.001/0.002, 0.01, 0.02, 0.03, 0.04 and 0.05) for Ciao and Last.Fm and (0.0001, 0.005, 0.1, 0.2, 0.3, 0.4, and 0.5) for FilmTrust. As shown in the following figures, the MAE and RMSE results are varied. But in general, high beta values cause low accuracy (high values of MAE and RMSE). Small beta values also lead to low accuracy. The best beta values that produce the highest accuracy are different from one dataset to another according to the sparsity ratio and the level of similarity among users. For example, the best beta value in general is 0.001 for Figures 5.10 and 5.11, where the results are 0.7221 and 0.9476 for MAE and RMSE respectively. In contrast, the best beta value for Figures 5.12 and 5.13 is 0.002 (better than the previous best value), the results are 0.3403 and 0.4575 for MAE and RMSE respectively. Likewise, the beta value is

different in Figures 5.14 and 5.15. The best beta value in these figures is 0.005. The results for MAE and RMSE are 0.6035 and 0.7995 respectively.

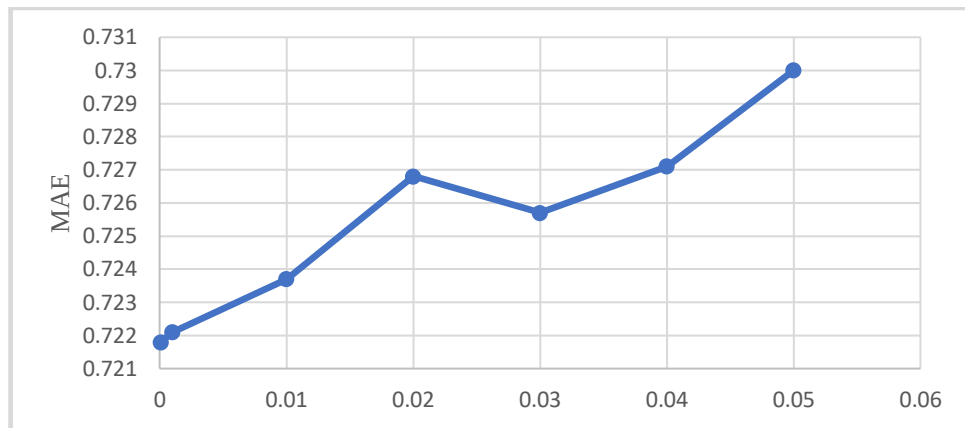


Figure 5.10. Impact of different beta values on MAE results for Ciao.

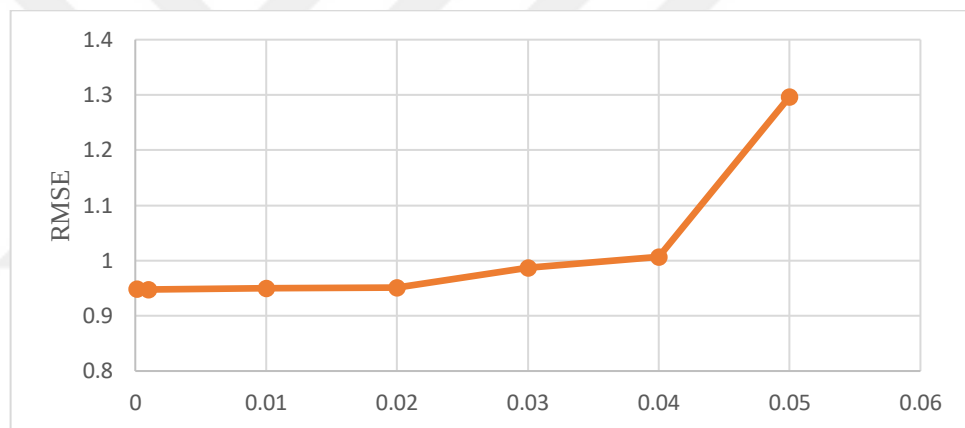


Figure 5.11. Impact of different beta values on RMSE results for Ciao.

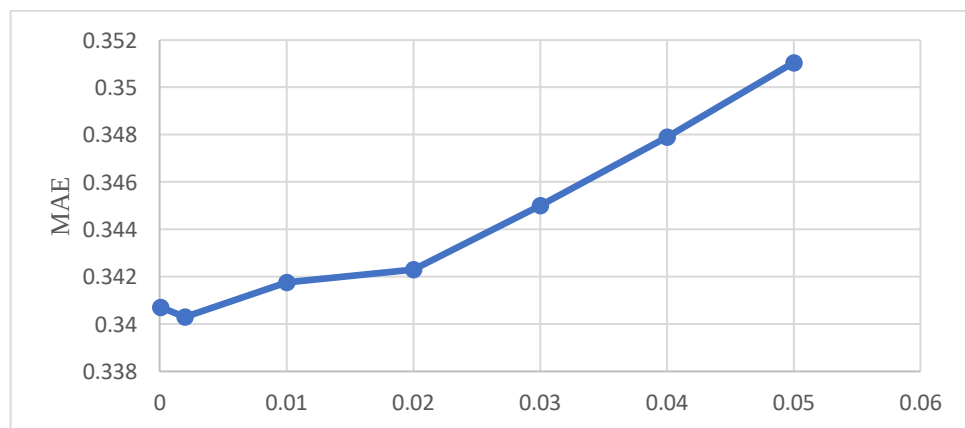


Figure 5.12. Impact of different beta values on MAE results for Last.Fm.

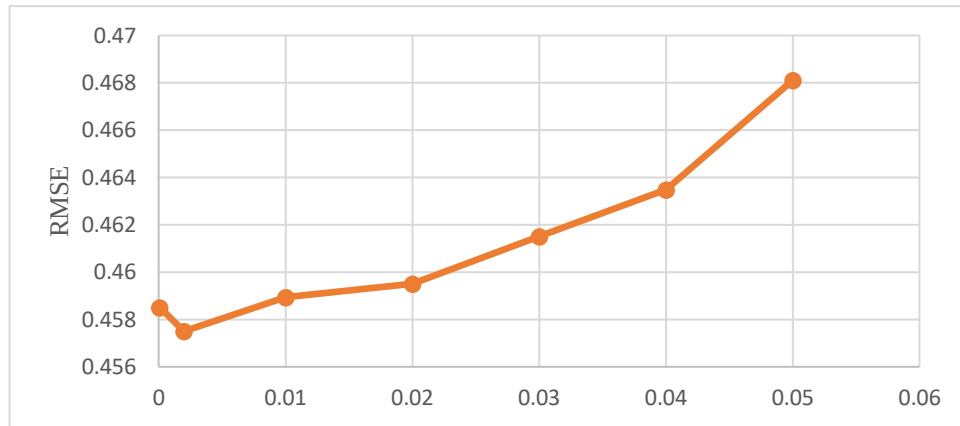


Figure 5.13. Impact of different beta values on RMSE results for Last.Fm.

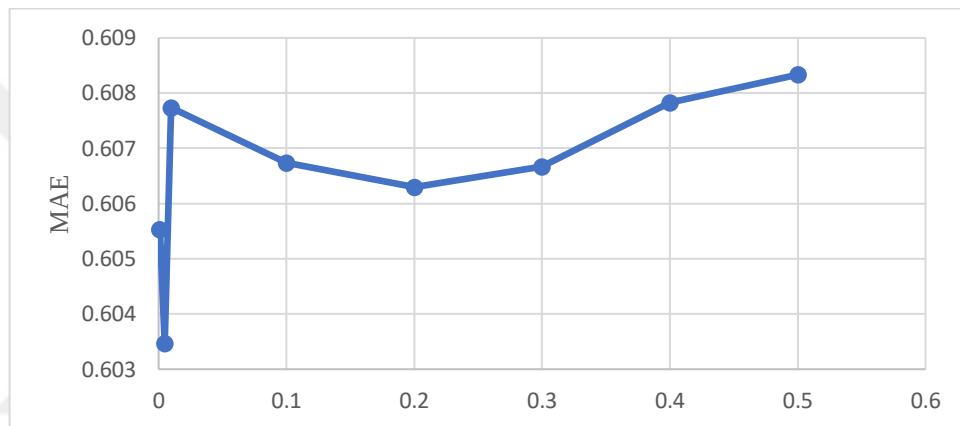


Figure 5.14. Impact of different beta values on MAE results for FilmTrust.

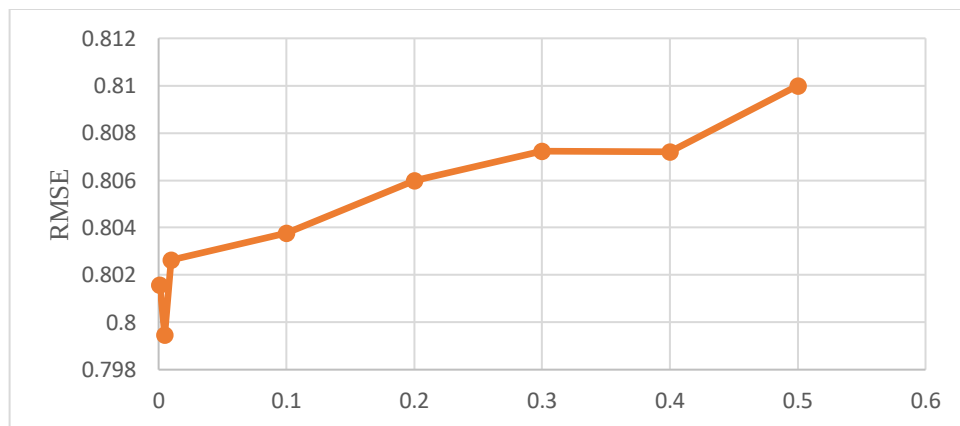


Figure 5.15. Impact of different beta values on RMSE results for FilmTrust.

5.5.3.4 Impact of changing gamma values

Figures 5.16-5.21 describe the results of MAE and RMSE for various gamma (learning rate) values. The odd number figures are dedicated to the results of RMSE, while the even numbers figures are devoted to the results of MAE. The x-axis indicates the learning rate values, which are determined by a dataset according to the data sparsity and the rating values to reach the best results. In Figures 5.16 and 5.17, the x-axis is between 0.001 and 0.3. This range is determined by trial and error. In Figures 5.18 and 5.19, however, the x-axis values are between 0.1 and 0.7. Finally, in Figures 5.20 and 5.21, the range is between 0.1 and 0.5.

As seen below, the structure is almost similar for each pair of the figures (5.16 and 5.17), (5.18 and 5.19) and (5.20 and 5.21) because each pair indicates the results of MAE and RMSE for one dataset. For instance, the results of Figures 5.16 and 5.17 start with bad values when gamma equals 0.001. They then improve slightly until they reach their best results (0.7221 for MAE and 0.9476 for RMSE) when gamma equals 0.006. Afterward, the results decline again when the gamma value increases to reach 0.7282 in Figure 5.16 and around 0.956 in Figure 5.17. Likewise, in Figures 5.18 and 5.19, the results are worse (0.3488 and 0.4732 respectively) when the gamma value is 0.07. Furthermore, the same scenario of Figures 5.16 and 5.17 can be seen in Figure 5.18. The result is bad at the first value of the x-axis (0.01), but then it starts to improve to reach its best 0.3403 at 0.02. After that, the MAE results rise significantly (accuracy decreases) when the gamma value increases until reach 0.3488 at 0.07. In contrast, the first value of the x-axis produces the best result in Figure 5.19. The RMSE results then increase (accuracy declines) when the gamma value rises. In Figures 5.20 and 5.21, the best results start when gamma is 0.01. The results are 0.6039 and 0.7995 for MAE and RMSE respectively. Then, the MAE and RMSE values raise in conjunction with an increase in the gamma value to reach the worst results at 0.05.

In summary, gamma (or learning rate) values play a significant role to enhance prediction accuracy during the training stage. They can reduce the regularization error to reach reasonable results. The gamma values vary in different datasets according to their sparsity ratio and rating values. Finally, gamma values are determined by trial and error.

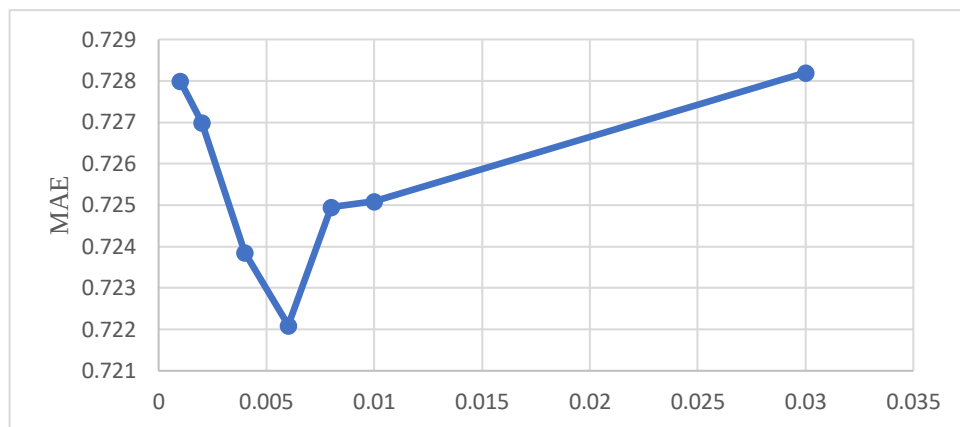


Figure 5.16. Impact of different gamma values on MAE results for Ciao.

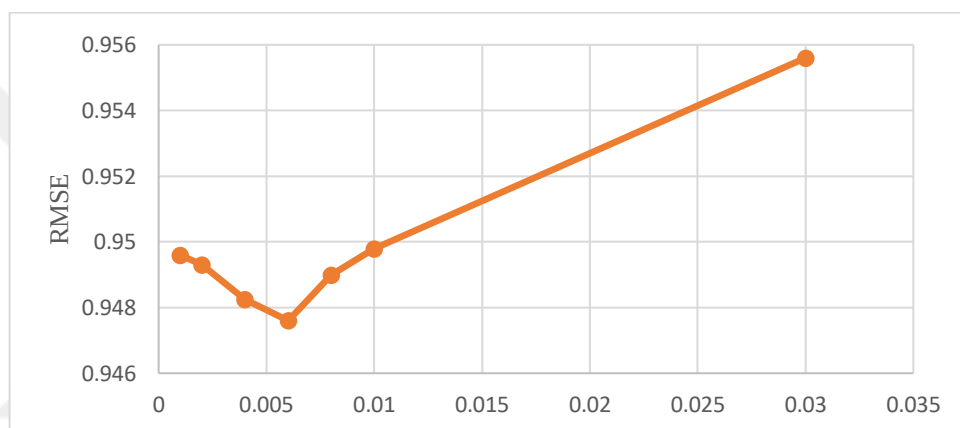


Figure 5.17. Impact of different gamma values on RMSE results for Ciao.

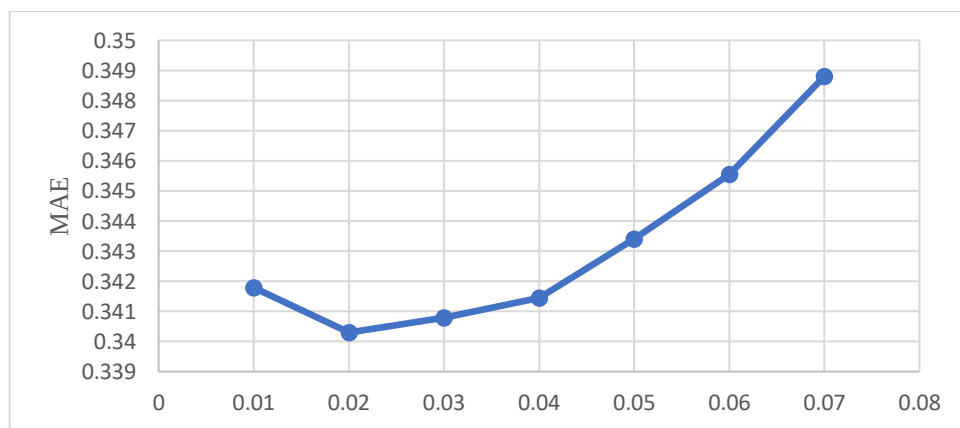


Figure 5.18. Impact of different gamma values on MAE results for Last.Fm.

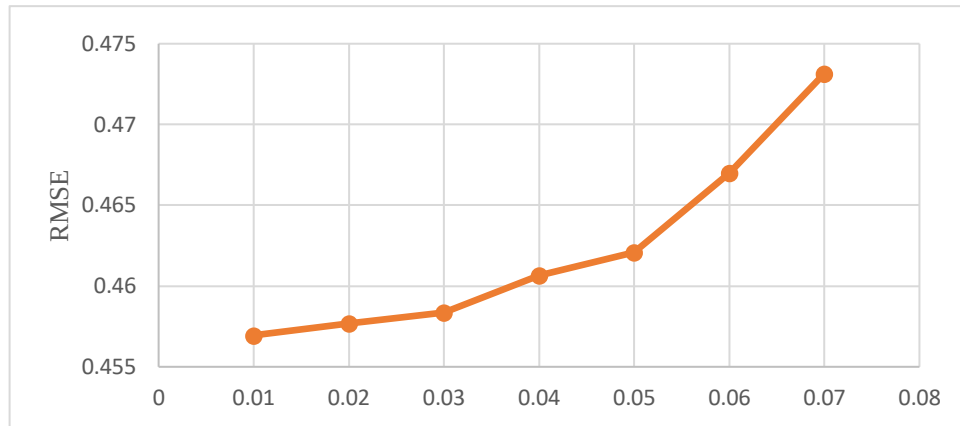


Figure 5.19. Impact of different gamma values on MAE results for Last.Fm.

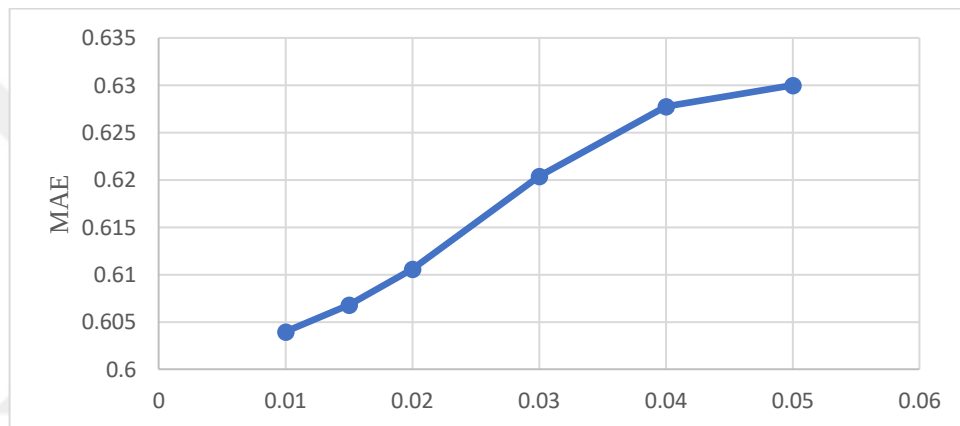


Figure 5.20. Impact of different gamma values on RMSE results for FilmTrust.

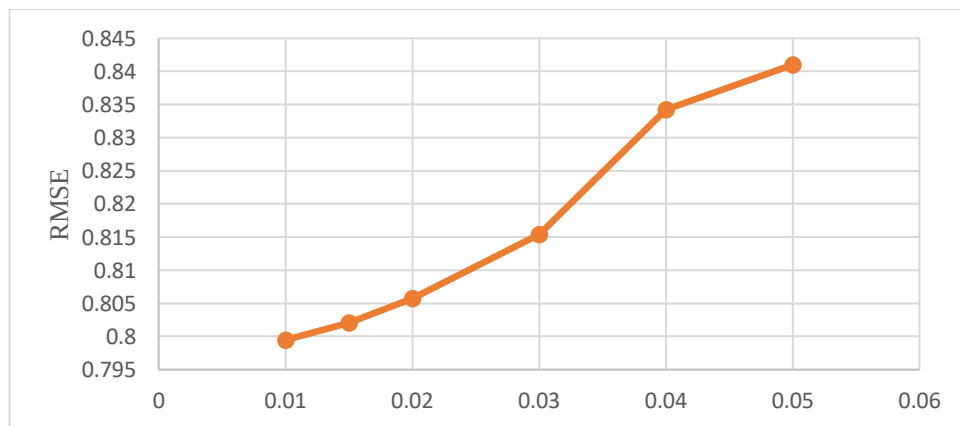


Figure 5.21. Impact of different gamma values on MAE results for FilmTrust.

5.5.3.5 Decision support accuracy results

Tables 5.6, 5.7 and 5.8 display the results of decision support accuracy, which involves three metrics: precision, recall and f-measure. The results are presented in multi top- n recommended items (10, 15 and 20). Moreover, the results of the proposed method are benchmarked with our previous model (MSRA-SVD) and the SVD method. The three models were tested on the Ciao, Last.fm and FilmTrust datasets. As shown in Tables 5.6, 5.7 and 5.8, the proposed model outperformed the SVD and MSRA-SVD methods in all values of n in terms of precision, recall and f-measure metrics respectively. For example, the results improved by approximately 0.40% when n is 15 for the Ciao dataset. Furthermore, the improvement is about 0.66% when n equals 15 and 0.65% when n is 20 for FilmTrust dataset. Additionally, Table 5.7 achieved the best results in terms of recall for Last.Fm dataset. The improvement is 0.55%, 1.11% and 1.14% when n is 10, 15 and 20 respectively. The same scenario can be seen with Ciao and Film Trust, where all results confirm that the superiority of the proposed model over the other models.

Tables 5.6 and 5.7 can be regarded as contrary. In Table 5.6, the precision value decreases when n raises. To justify, if we have 100 various items (relevant and irrelevant), the likelihood of choosing 10 relevant items is higher than choosing 20 such items. In contrast, the results of recall and the value of n both increase in Table 5.7. In the recall equation, the dominator, which indicates the relevant items of the entire dataset, is always constant regardless of n value. Only the numerator, which implies the top- n relevant items, changes according to n value. Thus, the recall value increases as long as n increases.

To sum up, the proposed method improved the performance of the recommendation system and outperformed the MSRA-SVD model in terms of precision, recall and f-measure for all top- n values.

Table 5.6. Precision results.

	Method	Top 10	Top 15	Top 20
Ciao	SVD	0.3545	0.2758	0.2237
	MSRA-SVD	0.3612	0.2826	0.2290
	MSRA-SVD++	0.4316	0.2968	0.2291
Last.Fm	SVD	0.6085	0.4276	0.3227
	MSRA-SVD	0.6289	0.4469	0.3376
	MSRA-SVD++	0.6982	0.5051	0.3765
FilmTrust	SVD	0.291	0.196	0.147
	MSRA-SVD	0.2984	0.2015	0.1514
	MSRA-SVD++	0.3004	0.2037	0.1525

Table 5.7. Recall results.

	Method	Top 10	Top 15	Top 20
Ciao	SVD	0.9018	0.9265	0.9370
	MSRA-SVD	0.9297	0.9558	0.9674
	MSRA-SVD++	0.9415	0.9634	0.9691
Last.Fm	SVD	0.5402	0.5646	0.5655
	MSRA-SVD	0.5494	0.5722	0.5731
	MSRA-SVD++	0.5615	0.5917	0.5930
FilmTrust	SVD	0.602	0.608	0.6075
	MSRA-SVD	0.6122	0.61655	0.62
	MSRA-SVD++	0.617	0.6188	0.6225

Table 5.8. F-measure results.

	Method	Top 10	Top 15	Top 20
Ciao	SVD	0.5089	0.4251	0.3612
	MSRA-SVD	0.5203	0.4363	0.3703
	MSRA-SVD++	0.5919	0.4538	0.3705
Last.Fm	SVD	0.5723	0.4866	0.4107
	MSRA-SVD	0.5865	0.5018	0.4248
	MSRA-SVD++	0.6224	0.5453	0.4606
FilmTrust	SVD	0.393	0.2964	0.2367
	MSRA-SVD	0.3968	0.3022	0.2429
	MSRA-SVD++	0.404	0.3065	0.2471

5.6 Summary

Using explicit information only to improve a recommendation system performance is not enough. Therefore, implicit information can be exploited to extract valuable sources to mitigate the limitations of recommendation systems such as the data sparsity problem. This chapter was dedicated to highlighting the third research question.

RQ3 How can social information and explicit/implicit rating feedback information be incorporated into the SVD++ method?

The MSRA-SVD++ method was adopted in this chapter to achieve the third objective and answer the third question of this research. The proposed method surmounted the drawbacks of the previous studies. EISR is one of those studies. It exploited only the first step of indirect friends and ignored the other steps. Furthermore, EISR did not use implicit feedback information. Four sources of information were exploited in this chapter. They are implicit social relations, explicit social relations, explicit rating information and implicit rating information. All these sources were integrated into the SVD++ method. Moreover, an optimization technique is used to reduce the square regularization error. It is Stochastic Gradient Descent (SGD). Additionally, the proposed method was applied to three real-world datasets: Ciao, Last.Fm and

FilmTrust. Five accuracy metrics were utilized to test the proposed model. They are MAE, RMSE, precision, recall and f-measure. The results revealed that implicit feedback information and implicit social relations have a major role to boost the prediction accuracy. The prediction performance is improved by exploiting such sources. Eventually, the MSRA-SVD++ model outperforms the MSRA-SVD that exploits three sources instead of four.



6. SVD++ AND CLUSTERING APPROACHES TO ALLEVIATE THE COLD-START PROBLEM FOR RECOMMENDATION SYSTEMS

With the growing utilization of E-commerce, huge amounts of information are available online, which leads to information overload. Therefore, finding preference items among massive choices is a really hard job. Recommendation systems (RSs) are a successful solution to overcome this problem. The previous chapters (4 and 5) alleviated the sparsity problem. In this chapter, the cold-start problem is highlighted.

One of the most common challenges is the cold-start problem which includes cold-start users and cold-start items. A cold-start item is a new item that has been introduced to a system. It is hard to predict the customer preference for this item, or the item will be at the end of the prediction list due to the lack of feedback given on this item (still new). Likewise, the cold-start user problem occurs when a new user registers in the system or when a user has few ratings (less than a threshold) (Son, 2016). Recently, the cold-start user problem has attracted many researchers to alleviate this limitation. We think that clustering users into groups based on their tastes and relations helps to recognize cold-start users' friends and subsequently recommend items that meet such users' tastes. An RS can exploit this technique by recommending items that were rated by the users of a particular cluster to which the targeted user belongs. Accordingly, this chapter is dedicated to handling the cold-start- user problem by exploiting clustering and dimension reduction techniques. Partitioning Around Medoids (PAM) algorithm is adopted to implement the clustering method and SVD++ method is employed for the prediction process. Moreover, the fourth research question is addressed in this chapter.

RQ4 How can user clustering based on social relations and confidence values can be created using the PAM algorithm?

The rest of this chapter illustrates and answer of the mentioned research question.

6.1 An Overview of the Proposed Model

As mentioned above, two techniques are utilized in this chapter to tackle the cold-start problem: PAM algorithm and SVD++ method. Moreover, social relations involve

explicit and implicit social relations. Also, rating information, which includes explicit rating and the implicit feedback information, is used to cluster users. Furthermore, to increase the information sources that are used in the clustering process, implicit trust relations and confidence values are exploited. The implicit trust relations are extracted based on rating values. Subsequently, three sources of information are used in the clustering process and two different sources are utilized in the prediction stage; so, five sources of information are exploited in this model. Due to Last.Fm dataset does not have cold start users, the experiments are applied on Ciao and FilmTrust datasets.

Although social relations are provided worthy information can boost prediction accuracy and utilized in many studies such as (Ma et al., 2009; Ma et al., 2011; Jamali et al., 2010; Zehang et al., 2013; Guo et al., 2016; and Reafee et al., 2016), it is not available in all datasets. Moreover, in most datasets even when social relations are available; it is much sparser than rating information. Therefore, to overcome the lack of information of datasets, combining social relations with confidence values, which can be extracted from rating information, is implemented to produce weight values. The weight values are employed to compute the distances among users and then clustered them into several groups/communities. For each community, a set of cold start users are determined, then the SVD++ method is applied to compute the prediction items for those users.

6.2 Methods and Data Sources

The proposed model involves applying several methods such as extraction of social relations, confidence values, applying PAM algorithm, and SVD++ method. Social relations consist of direct relations or trust relations and indirect relations or friend relations. For both types, there are explicit and implicit social relations. In this model, explicit and implicit relations for direct social relations are utilized. The following subsections explain the extraction of explicit social relations, implicit social relations, confidence values, PAM algorithm, and SVD++ method.

6.2.1 Extraction of social relations

social relations are usually used in RS to alleviate the sparsity and cold-start problems and improve the accuracy of similarities between users (Chen et al., 2018). In this

study, the similarity values are utilized to cluster users into a set of communities on the basis of social relations between them. In other words, if two users (trustors) have a similar set of trustees, there is a high likelihood that both users have similar tastes, and they will be consequently in the same community. Additionally, social relations involve two types: explicit and implicit. Therefore, in this study, both types are adopted to group users on the basis of their relations, and the following subsections demonstrate the extraction process of these types.

6.2.1.1 Extraction explicit social relations

Explicit relations are obtained directly from a dataset. Usually, these data include two columns: trustor and trustee. Based on these columns, a binary user-user matrix is created: trustor users are the rows, and trustee users are the columns. The one values are placed when there are interactions between trustor and trustee users; otherwise place zero.

However, explicit relations are few and counting on these relations to compute the common features among users will lead to the exclusion of the majority of them from the process. Subsequently, the clustering process gives inaccurate results, since most users remain out of clusters as outliers. Therefore, implicit relations are extracted and added to explicit ones to increase the relations among users.

6.2.1.2 Extraction implicit social relations

Implicit social relations are a substantial source used to enrich social networks with more information. These relations can be concluded based on explicit relations. Several properties are dedicated to developing implicit trust relations such as asymmetry, transitivity, dynamicity, and context-dependence, for more details, review section (2.7). In this study, the transitivity technique is followed to extract implicit trust relations. For example, assume that user a trusts user b , and user b trusts user c , then undoubtedly there is a high likelihood of establishing a new relationship (implicit relation) between a and c .

Implicit social relations are added to the binary user-user matrix that was created when we collected explicit social relations. Like explicit relations, if there is an implicit

relationship between a particular pair of users the one value is placed, or zero otherwise. After implicit and explicit trust relations are attained, the similarity value between each pair of the explicit and/or implicit user is computed. The Jaccard binary measure is employed to compute the similarity by using Equation (6.1).

$$jac_{u,v} = \frac{x}{x+y+z} \quad (6.1)$$

where x is the number of common trusted users between users u and v . Whereas y indicates the number of trusted users for user u only, and z refers to the number of trusted users for user v only.

6.2.2 Confidence values

Although much information can be extracted from the social network by applying the transitive technique, some datasets like FilmTrust have poor data in terms of social relations, namely, 1358 explicit relations and 73057 implicit ones. Clustering users on the basis of explicit and implicit social relations leads to ignoring many users as outliers. Thus, more sources are required to include more users in the clustering technique.

Subsequently, rating information is employed to compute confidence values or trust statements between each pair of users. Then, both the similarity values and confidence values are incorporated using the following Equation:

$$w_{u,v} = \begin{cases} \frac{sim_{u,v} + T_{u,v}}{2}, & sim_{u,v} \neq 0, \text{ and } T_{u,v} \neq 0 \\ sim_{u,v}, & sim_{u,v} \neq 0, \text{ and } T_{u,v} = 0 \\ T_{u,v}, & T_{u,v} \neq 0, \text{ and } sim_{u,v} = 0 \\ 0, & otherwise \end{cases} \quad (6.2)$$

where $sim_{u,v}$ denotes the similarity between users u and v , which is computed using Equation (6.1). $T_{u,v}$ is the confidence value between users u and v and it is computed by Equation (6.3).

$$T_{u,v} = A_{u,v}/A_u \quad (6.3)$$

$A_{u,v}$ is the number of common items that were rated by users u and v , and A_u denotes the number of items rated by user u .

To understand the process of finding implicit trust relations from explicit relations, the following example displays this issue. Figure 6.1 shows an example of explicit trust relations.

	U1	U2	U3	U4	U5	U6	U7	U8	U9
U1			1		1	1			
U2	1						1	1	1
U3					1		1		1
U4	1								1
U5	1						1	1	1
U6				1					1
U7									1
U8									
U9	1			1		1			

Figure 6.1. An example of explicit relations.

Now, the transitive technique can be applied to the above figure to extract implicit trust relations. The implicit relations are concluded based on a rule if x trusts y and y trusts z , so x trusts z . Figure 6.2 shows the extraction process of implicit trust relations for 9 users. In Figure 6.2, the black values refer to the explicit relations and red values indicate implicit trust relations. For example, u_1 trusts u_6 , u_6 trusts u_4 , it can be concluded u_1 trusts u_4 . Moreover, u_1 trusts u_5 , u_5 trusts u_7 , thus a new relationship can be established which, u_1 trusts u_7 .

	U1	U2	U3	U4	U5	U6	U7	U8	U9
U1			1	1	1	1	1	1	1
U2	1		1	1	1	1	1	1	1
U3				1	1	1	1	1	1
U4	1		1		1	1			1
U5	1		1	1			1	1	1
U6	1			1					1
U7	1			1		1			1
U8									
U9	1			1		1			

Figure 6.2. The extraction process of implicit relations.

To compute the similarity between u1 and u2, Equation (6.1) is implemented as follows:

$$jac_{u1,u2} = \frac{7}{7+0+1} = 0.875$$

The similarity between u4 and u5 as follows:

$$jac_{u4,u5} = \frac{3}{3+2+3} = 0.375$$

The similarity for all users is computed, a new user-user matrix is created after computing similarity values.

The confidence values are used to reduce the sparsity ratio in a dataset based on the users' ratings. Figure 6.3 depicts a sample of a user-item matrix.

	T1	T2	T3	T4	T5	T6	T7	T8
U1	4		3	4		2		5
U2		3	4	2	5			
U3		4		4		2	2	4
U4	3		5		2	4		
U5	2	5		2		4		4

Figure 6.3. A sample of a user-item matrix.

The confidence values are computed as follows:

$$T_{u1,u2} = \frac{2}{5} = 0.4$$

$$T_{u2,u1} = \frac{2}{4} = 0.5$$

$$T_{u3,u4} = \frac{1}{5} = 0.2$$

$$T_{u4,u3} = \frac{1}{4} = 0.25$$

All the confidence values are computed using the same way, in the end, a user-user matrix will be created for the confidence values.

Now, confidence values and explicit trust relations with implicit trust relations are ready to use. Weight values are calculated using Equation (6.2). In this equation, social relations and confidence values are integrated for each pair of users. If a pair of users has a similarity value of social relations and confidence value, the first option of the Equation (6.2) implements. While if it has a similarity value only, the weight value assigns the similarity value. Whereas if the pair has trust value only, the weight assigns the trust value. Finally, if there is neither a trust value nor a similarity value, the weight will be zero.

After obtaining the weight values, PAM algorithm can be implemented now to cluster the users into several groups. The following section explains PAM algorithm.

6.2.3 Partitioning Around Medoids (PAM) Algorithm

The clustering technique in a recommendation system is a process that distributes the users of a given dataset into many groups according to the distances between them. In other words, users similar in terms of ratings, relations or any other features are gathered together in one community. Subsequently, items that interest a particular cluster are recommended to a user who belongs to that cluster.

PAM is a conventional partitioning clustering algorithm based on the k-medoids technique. A medoid or center is an object of a particular cluster where the distance between each object and the selected medoid is the minimum. The k-medoid is more robust than the k-means as it is less sensitive to outliers and noise.

Algorithm 1: PAM

- 1- Select k medoids randomly, where $k < n$ and n is the number of nodes in the whole dataset. Each medoid indicates the center of a cluster.
- 2- Apply the distance measure (the Euclidean distance, the Manhattan distance, or any other distance method) to find the distance (cost₁) between each point and the selected medoids.
- 3- Assign each node to the closest medoid.
- 4- Compute the total cost of all nodes.
- 5- Select a new medoid from the non-medoid objects randomly.
- 6- Compute the new cost (cost₂) using the similarity measure.

7- Difference = $\text{cost}_2 - \text{cost}_1$

8- If the difference < 0 , swap the new medoid with the old one.

9- Repeat steps 2-8 until the medoids do not change.

6.2.4 SVD++ method

SVD++ was produced by (Koren *et al.*, 2008). It is an expansion of the SVD method. The implicit rating information is the data that is considered in SVD++. Recently, many studies have employed this approach (Jiao *et al.*, 2019; Shi *et al.*, 2020) due to its ability to enhance accuracy. The main idea of this method is reducing the amount of rating information (user-item matrix), where the new user matrix and the new item matrix are derived by detecting the latent features from the user-item matrix. Then the prediction value is computed as follows:

$$r_{u,i} = \mu + b_u + b_i + q^T \left(p_u + \frac{1}{\sqrt{|N_u|}} \sum_{j \in N_u} y_j \right) \quad (6.4)$$

$$b_i = \frac{(\sum_{u \in R(i)} r_{u,i} - \mu)}{\lambda_1 + |R_i|} \quad (6.5)$$

$$b_u = \frac{(\sum_{u \in R(u)} r_{u,i} - \mu - b_i)}{\lambda_2 + |R_u|} \quad (6.6)$$

where μ is the mean value of all the entire ratings; and b_u and b_i are the observed deviations from the μ of user u and item i respectively. The number of items rated by user u is denoted as $|N_u|$, $|R_u|$ is the number of users who rated a specific item, $|R_i|$ is the number of items that were rated by a set of users, and y_j is the left orthogonal matrix of the implicit matrix. λ_1 and λ_2 are two regularization parameters.

6.3 Proposed Model

In this study, the datasets are divided into two parts: training and testing according to a threshold that determines cold-start users for the testing phase and non-cold-start users for the training phase. Two main sources are employed in the proposed model: social relations and users' ratings. First, the social relations and ratings of non-cold-start users are exploited to compute the similarity and confidence values respectively. Then, the PAM clustering method is adopted to group users into several communities. Second, again users' ratings are utilized, which include explicit and implicit feedback,

to compute the missing predictions for cold-start users in each cluster. Figure 1 shows the details of the proposed method.

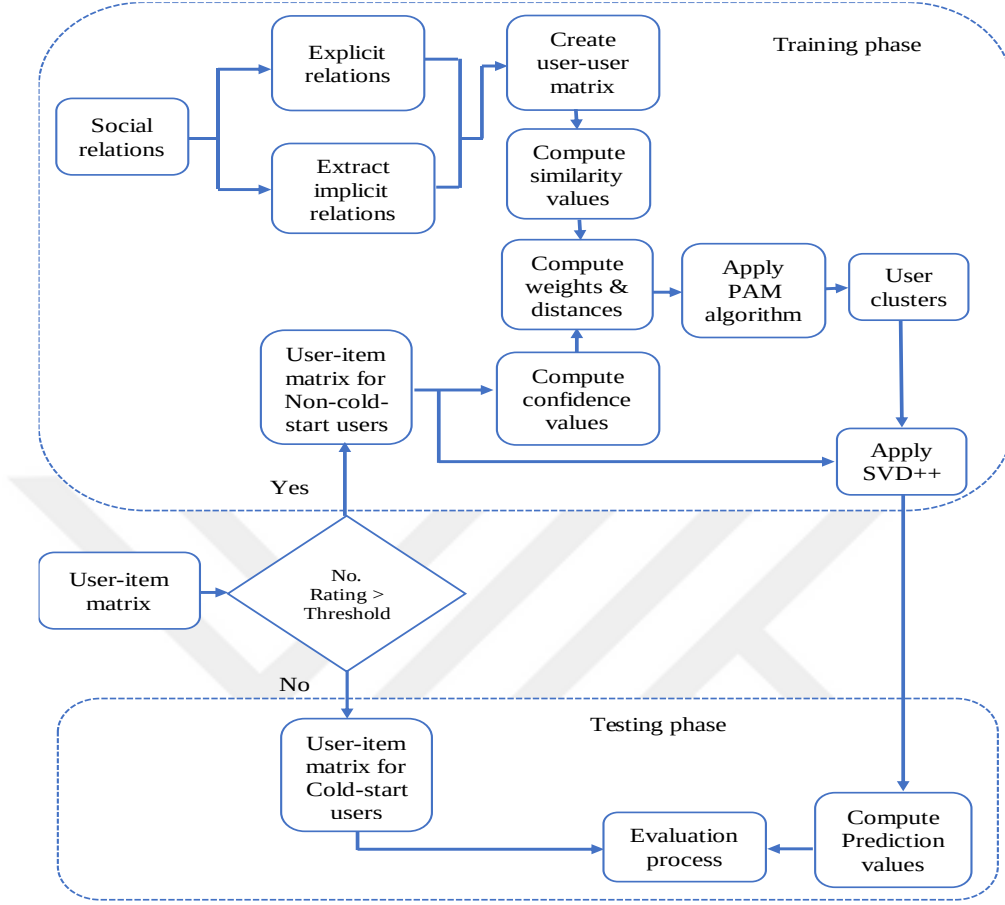


Figure 6.4. An overview of the proposed model.

The following steps explain the proposed method.

Step 1: The dataset is divided based on the rating value of each user into two phases. If the rating value exceeds a threshold, the user is considered a non-cold-start user, and his rating is used in the training phase; otherwise, he is assigned as a cold-start user in the testing phase.

Step 2: In the training phase, a binary user-user matrix is created to represent explicit and implicit social relations. If there is a relation between two users, one value is placed; otherwise, it is zero.

Step 3: The similarity of explicit and implicit social relations is computed using Equation (6.1).

Step 4: To decrease clusters' outliers, rating information is employed to compute confidence values among each pair of users by using Equation (6.3). A new user-user matrix is created for the confidence values.

Step 5: The confidence values and similarity values are incorporated to compute weight values by using Equation (6.2).

Step 6: After acquiring the weights, the distances are calculated by subtracting the weight values from one. Then, the PAM clustering algorithm is adopted to cluster the users according to the computed distances. The value of k (the number of clusters) is determined by trial and error. The MAE and RMSE metrics are computed for different k values, then k value is selected depending on the best result among these metrics. Moreover, two conditions are dedicated to determining the medoids in the PAM algorithm. First, the condition uses the greatest number of relations of the object to be a medoid. While the second condition finds the minimum distance between the medoid and the objects on a particular cluster. The output of this step is users' clusters.

Step 7: The stochastic gradient descent method is utilized to optimize the result of the SVD++ method by using the following Equations (6.8) – (6.12), which are applied to tune the values of p_u , q_i , y_j , b_u , and b_i respectively.

$$E = \sum_{(u,i) \in m} \left(r_{ui} - \mu - b_u - b_i - q^t \left(p_u + \frac{1}{\sqrt{|N_u|}} \sum_{j \in N_u} y_j \right) \right)^2 + \lambda (b_i^2 + b_u^2 + \|p\|^2 + \|q\|^2 + \|y_j\|^2) \quad (6.7)$$

$$p_u = p_u + \gamma(Eq_i - \lambda p_u) \quad (6.8)$$

$$q_i = q_i + \gamma \left(E \left(p_u + \frac{1}{\sqrt{|N_u|}} \sum_{j \in N_u} y_j \right) - \lambda q_i \right) \quad (6.9)$$

$$y_j = y_j + \gamma \left(E \left(\frac{q_i}{\sqrt{|N_u|}} \right) - \lambda y_j \right) \quad (6.10)$$

$$b_u = b_u + \gamma(E - \lambda b_u) \quad (6.11)$$

$$b_i = b_i + \gamma(E - \lambda b_i) \quad (6.12)$$

E is the regularization error value, $\lambda > 0$ is utilized to set the degree of the constraint used to avoid over-fitting, γ is the learning rate, and $\|x\|$ denotes the Frobenius norm.

Step 8: In the testing phase, The SVD++ method is implemented in each cluster to compute the predictions for cold-start users through Equation (6.4). After applying SVD++ on each cluster, an average is calculated to attain the final outcome.

Step 9: The evaluation process is done by comparing the computed prediction values with the actual values in the dataset.

In the training phase, the prediction value is computed by utilizing the values of p_u , q_i , y_j , b_u , and b_i factors, then it is compared with the actual rating value to compute the difference (regularization error) between them. Additionally, stochastic gradient descent is a well-known optimization method, which is applied for several loops to reduce the regularization error and update the values of the aforementioned factors. Consequently, the prediction value will be too close to the actual rating. After tuning the values of these factors, they can be utilized in the testing phase to predict items for cold-start users.

According to the stochastic gradient descent method, there is a rule that should be followed to update the factors. These rules depend on some parameters, such as learning rate value (γ), the regularization error (E), constant value to avoid overfitting (λ), and the old values of the factors. In Equation (6.7), the regularization error (E) is computed in each loop, then it is used in Equations 6.8 – 6.12 to update and tune the values of p_u , q_i , y_j , b_u , and b_i respectively. Equations (6.8) and (6.9) are used to update p_u and q_i latent factor vectors respectively. Equation (6.10) is used to update the implicit feedback set, which is represented by y_j . Finally, Equations (6.11) and (6.12) are adopted to update the observed deviations values of user u and item i respectively.

6.4 Experimental Evaluation

6.4.1 Description of the datasets

The proposed model is applied to real-world datasets that are widely used in recommendation systems, namely, Ciao and FilmTrust. Ciao is a general item dataset that is comprised of 7375 users and 99746 items, and the rating scale is 1-5 (Tang et al., 2012). Due to the memory consumption, each item with less than 2 ratings is removed, and thus, there are 22229 remaining items. FilmTrust is for movies, and it involves 1508 users and 2071 items. The rating includes 8 levels from 0.5-4 with a step size of 0.5 (Guo et al., 2013). Table 6.1 shows the details of the datasets.

Table 6.1. Statistics of the FilmTrust and Ciao.

Dataset	FilmTrust	Ciao
# users	1508	6767
# items	2071	22,229
# ratings	35,494	185,759
%Density	1.14	0.0012
# explicit relations	1853	111780
# implicit relations	73,057	54,056,070

6.4.2 Experimental setup

The implementation of this study depends on setting some parameters to achieve the best results. These parameters are shown in Table 6.2.

Table 6.2. Parameter setup.

Dataset	Ciao	FilmTrust
γ	0.06	0.07
λ	0.01	0.3
λ_1 and λ_2	6,8	20, 15
d	5, 10	5, 10
Iterations	5	5

The proposed method is evaluated by comparing its results with those of several state-of-the-art approaches including RSTE (Ma et al., 2009), SocialMF (Jamali and Ester, 2010), SocialRec (Ma et al., 2011), and TrustSVD (Guo et al., 2015); furthermore, the proposed method is compared with two recent studies including Trust ANLF (Parvin et al., 2019) and RTARS (Ahmadian et al., 2019) in terms of MAE and RMSE. The benchmarks were achieved with the aforementioned studies under the same conditions, which include the number of users, latent dimension values, and the number of iterations. The test was conducted for cold-start users. In this study, users were considered to be cold-start ones if they have less than 5 item ratings for Ciao and less than or equal 5 items for FilmTrust.

6.4.3 Results and discussion

As mentioned above, two datasets were utilized to evaluate the proposed method. For both datasets, the user-item matrices were divided into two parts: training and testing. The testing part included cold-start users and the remaining users were in the training part.

6.4.3.1 Determining the best cluster

Tables 6.3 and 6.4 show the results of the MAE and RMSE for different numbers of clusters. For example, 7 clusters mean that the users were distributed into 7 groups according to the distance between a user and the center of these clusters. As shown in Table 6.3, for the Ciao dataset, the results of the MAE and RMSE vary depend on the number of clusters, the number of normal users (noncold-start users), and the number of cold-start users in each cluster. The best result of the MAE is 0.7126 and that of the RMSE is 0.9221 when the number of clusters is 15. The same scenario can be seen in Table 6.4, but the best results are produced when the number of clusters is 7 for both metrics.

Table 6.3. Accuracy results of various numbers of clusters for Ciao.

No. of Clusters	MAE	RMSE
5	0.7268	0.9333
7	0.7200	0.9259
9	0.7186	0.9346
11	0.7213	0.9312
13	0.7278	0.9258
15	0.7126	0.9221
17	0.7475	0.9614

Table 6.4. Accuracy results of various numbers clusters for FilmTrust.

No. of Clusters	MAE	RMSE
5	0.6021	0.7885
7	0.5960	0.7706
9	0.6155	0.8181
11	0.6325	0.8064
13	0.6024	0.7996
15	0.6167	0.7811
17	0.6312	0.8078

6.4.3.2 Benchmarking with our previous models

Tables 6.5 and 6.6 show the results of the proposed method and our previous models (MSRA-SVD and MSRA-SVD++) in terms of MAE and RMSE. The results are reported in two-dimension values (5 and 10). Moreover, Table 6.5 shows the results of Ciao dataset, and Table 6.6 shows the results of FilmTrust dataset. The proposed model achieved a significant improvement that can be seen in both tables. The proposed method in Table 6.5 outperformed the previous models by approximately 1.73% for MAE and 2.87% for RMSE when $d=5$. Furthermore, the improvement of the proposed method is around 1.88% for MAE and 2.88% for RMSE when $d=10$ in the same table. Table 6.6 depicts the superiority of the proposed method, the improvement is about 4.1% and 10.7% for MAE and RMSE, respectively when $d=10$.

Consequently, the clustering technique plays a significant rule to enhance the prediction system for cold-start users. Clustering users based on their relations and confidence can gather similar/close users together (even cold-start users) in one cluster, then predict items from users who have a rich history to the cold start-users in the same cluster. This way as we have seen in the results was affected to enhance the accuracy and it is better than predicting items to the cold-start users from the entire dataset regardless of users' relations as achieved in the previous models.

Table 6.5. Performance comparison for cold-start users for Ciao. Boldface implies best results and (*) indicates the second-best results.

	Metrics	MSRA-SVD	MSRA-SVD++	Proposed model
D=5	MAE	0.7662	0.7270*	0.7144
	RMSE	0.9880	0.9499*	0.9226
D=10	MAE	0.7658	0.7263*	0.7126
	RMSE	0.9873	0.9494*	0.9221

Table 6.6. Performance comparison for cold-start users for FilmTrust. Boldface implies best results and (*) indicates the second-best results.

	Metrics	MSRA-SVD	MSRA-SVD++	Proposed model
D=5	MAE	0.6826	0.6526*	0.5958
	RMSE	0.8967	0.8627*	0.7703
D=10	MAE	0.6821	0.6518*	0.596
	RMSE	0.8965	0.8621*	0.7706

6.4.3.3 Benchmarking with state-of-the-art methods

Using social information only does not achieve good results, especially for datasets that have few social relations such as FilmTrust. Accordingly, this study overcame this limitation by utilizing confidence values along with social relations to reduce the sparsity to a reasonable level and decrease the outliers (users) during the clustering process by assigning them to clusters based on their social relations. Additionally, to conduct an efficient evaluation, the comparison used trust-based methods that also exploited the social relations and some of them used rating information as well. Tables

6.7 and 6.8, report the results of the MAE and RMSE for two latent factor values of (d) 5 and 10. The evaluation results are given for the Ciao and FilmTrust datasets. The best results are displayed using Boldface and (*) indicates the second-best results.

Additionally, as exhibited in Table 6.7 our method achieved the best results. The MAE is improved by approximately 0.47% and the RMSE is improved by approximately 0.64%, when (d) is 10 on the Ciao dataset compare with the second-best results (TrustANLF). Likewise, in Table 6.8, the proposed method outperforms the other studies in all cases. For instance, the closest results to our model are given by RTARS, the proposed method obviously superior. The results are approximately 0.50% and 0.44% in terms of the MAE and RMSE respectively when d is 10 in FilmTrust. A similar improvement can be seen when (d) equals 5 on the same dataset. The improvement is 53% for MAE and 48% for RMSE. Hence, for users having few ratings, extra sources of information are needed to predict their preferences. Therefore, social relations with confidence values provide a perfect choice for recommendation systems to serve this kind of user.

Table 6.7. Performance comparison for cold-start users for Ciao. Boldface implies best results and (*) indicates the second-best results.

	Metrics	RSTE	SocialRec	Trust SVD	Trust ANLF	Proposed model
D=5	MAE	0.957	0.789	0.729*	-----	0.7144
	RMSE	1.113	0.998	0.953*	-----	0.9226
D=10	MAE	0.803	0.730	0.721	0.716*	0.7126
	RMSE	1.014	1.031	0.962	0.928*	0.9221

Table 6.8. Performance comparison for cold-start users for FilmTrust. Boldface implies best results and (*) indicates the second-best results.

	Metrics	RSTE	SocialRec	Trust SVD	Trust ANLF	RTARS	Proposed model
D=5	MAE	0.680	0.670	0.661	-----	0.599*	0.5958
	RMSE	0.884	0.857	0.853	-----	0.774*	0.7703
D=10	MAE	0.674	0.668	0.663	0.607	0.599*	0.596
	RMSE	0.900	0.897	0.853	0.784	0.774*	0.7706

6.4.3.4 Impact of implicit social relations

Figure 6.5 shows the impact of utilizing implicit relationships. This figure displays 10% of the relations using the Gephi software. As shown in this figure, the number of users increases when including implicit relations. For example, Figure 6.5 (a) indicates the explicit relations only for the Ciao dataset, which has 111780 relations. Figure 6.5 (b), on the other hand, depicts the explicit-implicit relations for the same dataset, which has 54,056,070 relations. As seen in both figures, there are significantly more relations among users in Figure 6.5 (b). This fact demonstrates that the information extracted via explicit-implicit relations. This information can include more users, especially cold-start users since the opportunity to serve them by exploiting explicit-implicit relations is higher than when using explicit relations only. Furthermore, if the clusters are established using explicit relations only, most users will be considered outliers and are not assigned to any cluster. Therefore, many of cold-start users are not serviced. Subsequently, implicit relations provide an excellent option to enrich the relations and can contribute to clustering the users more efficiently. Similarly, Figures 6.5 (c) and (d) show the explicit and explicit-implicit relations respectively for the FilmTrust dataset. In these figures, the relations are also increased by adding the implicit relations. Whereas the number of explicit relations is 1853, the number of explicit-implicit relations is 73,057.

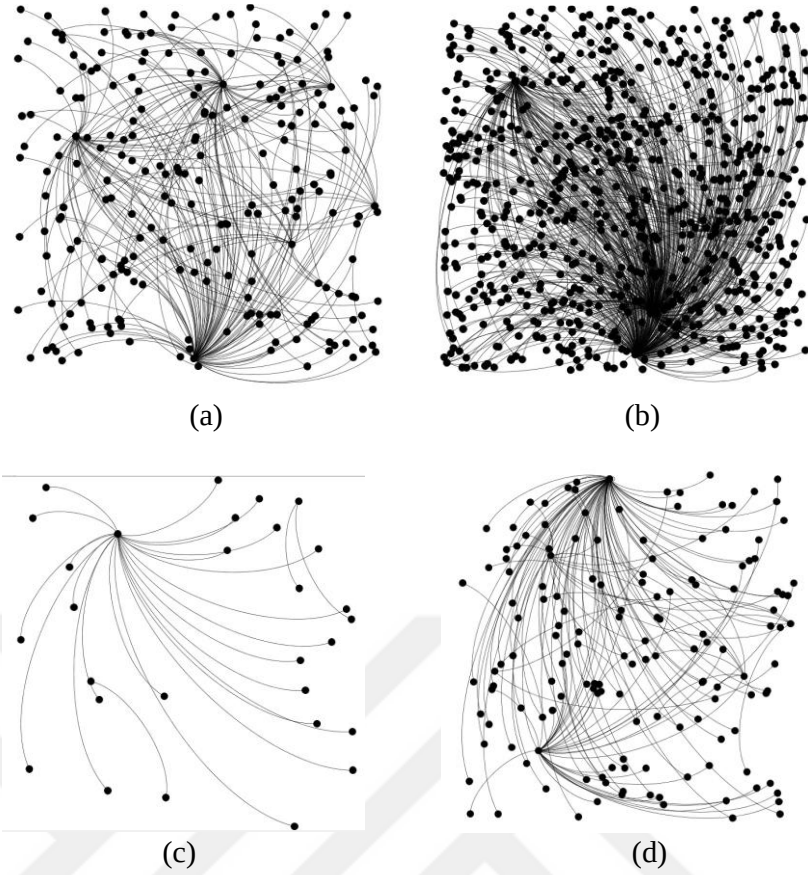


Figure 6.5. The impact of adding implicit relations. Figures (a) and (c) are explicit relations only for Ciao and FilmTrust, respectively. Figures (b) and (d) are explicit-implicit relations for the same dataset.

6.4.3.5 Impact of different lambda values

Figure 6.6 shows the results of the MAE and RMSE for different λ values ranging from 0.01 to 0.9. Figures 6.6 and 6.7 exhibit the results of the Ciao dataset for MAE and RMSE, respectively. Figures 6.8 and 6.9 depict the results of the FilmTrust dataset, for MAE and RMSE, respectively. It is worth mentioning that the MAE and RMSE are the accuracy metrics. The optimal value of these metrics is zero, and high values represent less accurate results. As shown in all figures, the results vary as long as λ increases. For example, Figure 6.6 shows that the best result is when λ equals 0.01, which is 0.7126, and then the results increase minimally from 0.1 to 0.9. Almost the same synopsis can be observed in Figure 6.7, where there is a sharp improvement from 0.01 to 0.1 until achieving the best result, which is 0.9102 when λ is 0.1. Nevertheless, the results moderately rise from 0.2 to 0.9, where the worst result is 0.9246 in terms of the RMSE. Moreover, in Figure 6.8, the MAE significantly improves when λ increases. Additionally, the improvement continues until λ equals 0.2, where the MAE

achieves the best result, which is 0.5959. However, when λ continues to increase, the accuracy slightly decreases, where the MAE is 0.597 when λ equals 0.9. The same scenario can be detected in Figure 6.9, but the improvement of the RMSE persists until the best result is reached, which is 0.7703 when λ is 0.3, however; the results worse when λ value increases.

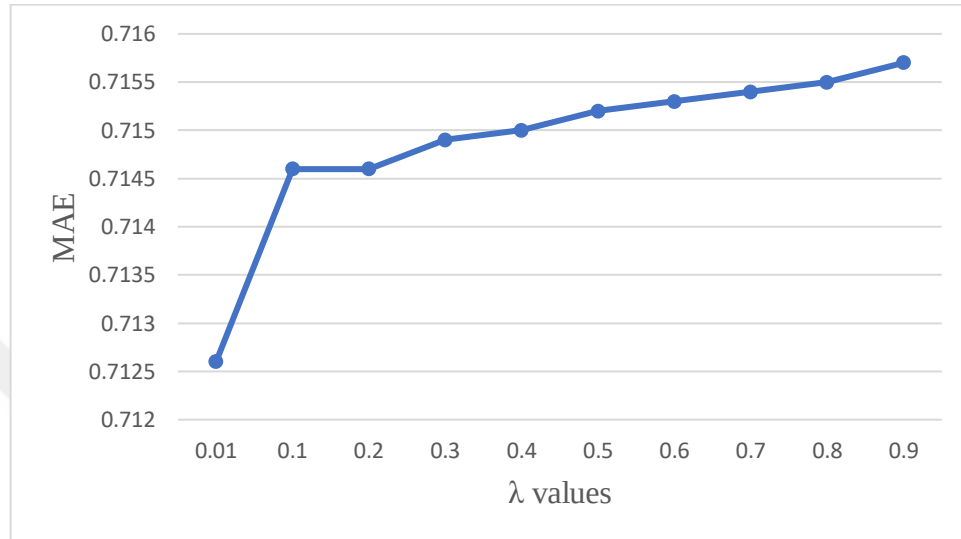


Figure 6.6. The MAE results for different lambda values for Ciao.

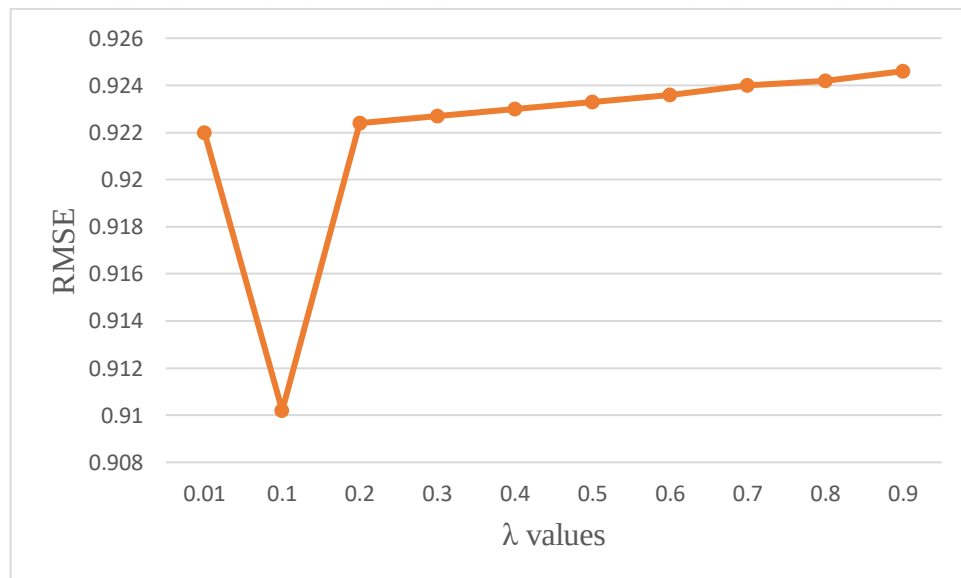


Figure 6.7. The RMSE results for different lambda values for Ciao.

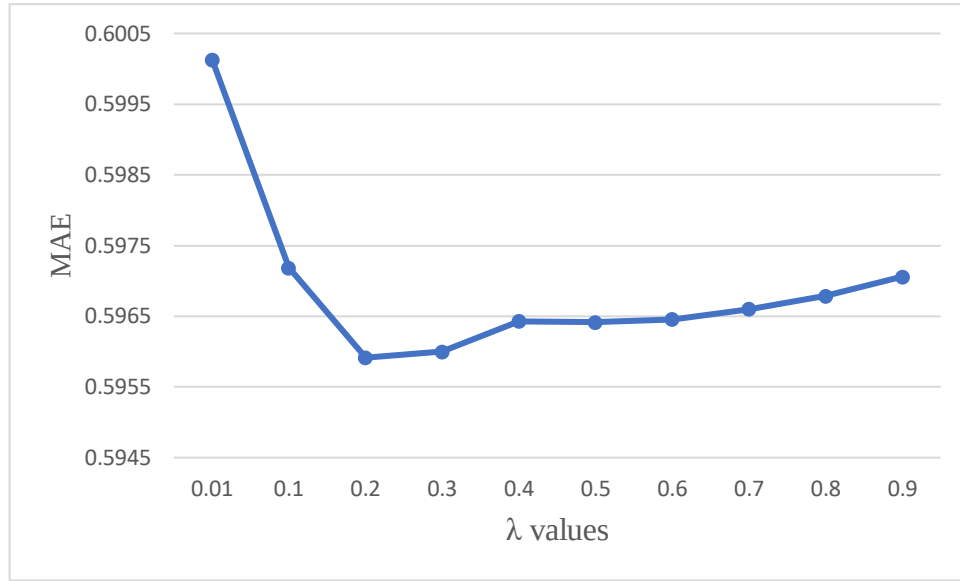


Figure 6.8. The MAE results for different lambda values for FilmTrust.



Figure 6.9. The RMSE results for different lambda values for FilmTrust.

6.4.3.6 Impact of using confidence values

Tables 6.9 and 6.10 display the results of MAE and RMSE for different cluster numbers. Two views are displayed in these tables with and without confidence values. In other words, the first view reports the results by combining social relations with confidence values. The second view depicts the results via social relations only. As seen in both tables, the clustering with confidence values outperformed that without confidence values. Two criteria are utilized to check the impact of using confidence

values. They are the accuracy metrics (MAE and RMSE) and the number of users in each cluster. Regarding the number of users, it is obvious that the clustering with confidence values includes more users than that without confidence values. For example, in Table 6.10, there is a big gap between clustering with confidence and clustering without confidence. The maximum number of users in the clustering without confidence values is 55 when the number of the clusters is 7. On the other hand, in the clustering with confidence values, the number of users is 302 with the same number of clusters. Thus, 247 users remain out of the clustering process as outliers. Table 6.10 displays the results of the FilmTrust dataset. In this dataset, there are few social relations among users, namely, 1853 explicit relations and 73,057 implicit ones. So, in the clustering process, relying on social relations (without confidence values) leaves most users out of the clustering as outliers. In contrast, Table 6.9 shows the results of the Ciao dataset. A small difference in the number of users can be seen between clustering with confidence and clustering without confidence values. As the Ciao dataset has a huge number of social relations among users, namely, 11780 for explicit relations and 54,056,070 for implicit once, confidence adds a small number of relations. Consequently, the difference between confidence and without confidence is small. For example, the difference is 13 users when the number of clusters is 15. Regarding the second criterion, the accuracy of the results is affected positively by adding more users. Thus, the results with confidence outperformed the results that do not use confidence values.

In summary, the relations among users can be represented using social relations or confidence values. Integrating both types lead to increasing relations between users. Hence more users can be involved in the clustering process. Thus, the clustering is implemented more accurately and more users will be served.

Table 6.9. The clustering results with and without confidence values, for Ciao dataset.

With confidence values				Without confidence values		
No. of Clusters	MAE	RMSE	No. cold-start users	MAE	RMSE	No. cold-start users
15	0.7126	0.9221	749	0.7166	0.9486	736

Table 6.10. The clustering results with and without confidence values, for FilmTrust dataset.

With confidence values				Without confidence values		
No. of Clusters	MAE	RMSE	No. cold-start users	MAE	RMSE	No. cold-start users
7	0.5960	0.7706	302	0.6317	0.7856	55

6.4.3.7 Decision support system results

Tables 6.11, 6.12, and 6.13 show the results of precision, recall, and f-measure respectively. The precision metric is used to check the ratio of recommended items by a system to the number of top-n value. The recall metric is utilized to compute the ratio of relevant items that are recommended by a system to the total number of relevant items in the testing part. The f-measure is used to compute a tradeoff between precision and recall metrics. The highest values refer to the best results, where the optimal result is one.

The results are reported for MSRA-SVD++ and the proposed model of this chapter. Moreover, the outcomes are displayed in multi top-n recommended items (2, 3, and 4). We selected three different values of top-n because 5 ratings or less are considered to represent cold-start users; the maximum number of ratings is 4 in the Ciao dataset and is 5 in the FilmTrust dataset. Subsequently, it is unreasonable to predict 10 items or more for such users (cannot measure the performance). The results revealed that the proposed method better than MSRA-SVD++ method in terms of precision, recall, and f-measure. For example, in Table 6.11, the improvement is 0.013, 0.0031, and 0.0051 for top-2, top-3, and top-4 respectively in Ciao dataset. Similarly, the proposed model outperformed MSRA-SVD++ model by about 0.2364 when $n=2$, 0.1772 when $n=3$, and 0.1461 when $n=4$, for FilmTrust dataset. In Table 6.12, the superiority of the proposed model over MSRA-SVD++ can be seen in all top-n values. For instance, the improvement is 0.0077, 0.0081, and 0.0518, when $n=2$, 3, and 4 respectively. The f-measure is outperformed the MSRA-SVD++ method as long as precision and recall metrics outperformed. Hence, once again clustering users into several groups based on their confidence and relations between them as done in this model and predicting items to cold-start users better than assuming all users as one cluster.

Table 6.11. Results of precision.

Dataset	Method	Top-2	Top-3	Top-4
Ciao	MSRA-SD++	0.4345	0.27	0.1921
	Proposed model	0.448	0.2731	0.1972
FilmTrust	MSRA-SD++	0.2462	0.1587	0.1069
	Proposed model	0.4826	0.3359	0.253

Table 6.12. Results of recall.

Dataset	Method	Top-2	Top-3	Top-4
Ciao	MSRA-SD++	0.0887	0.1008	0.101
	Proposed model	0.1064	0.1089	0.1528
FilmTrust	MSRA-SD++	0.11	0.1271	0.128
	Proposed model	0.3121	0.3324	0.3337

Table 6.13. Results of F-measure.

Dataset	Method	Top-2	Top-3	Top-4
Ciao	MSRA-SD++	0.1617	0.1468	0.1324
	Proposed model	0.1996	0.1516	0.1365
FilmTrust	MSRA-SD++	0.1605	0.1412	0.1165
	Proposed model	0.3748	0.3341	0.2878

6.5 Summary

Cold-start users are a common challenge in recommendation systems and at the same time, cold-start users represent a ratio in a particular system cannot ignore it. Therefore, this chapter aims to alleviate the effects of the cold-start problem and addresses the fourth research question.

RQ4 How can user clustering based on social relations and confidence values can be created using the PAM algorithm?

The limitations of the previous studies such as Trust ANLF (Parvin et al., 2019) and RTARS (Ahmadian et al., 2019) that they assumed all users in one group, and they predict items to targeted users from this group. However, we think that users belong to different tastes where each group has similar properties and different from another group. Additionally, the clustering technique can differentiate between different users based on their social relations. The contributions of this model are:

- I. The weight of each node is computed by incorporating social relations and confidence values.
- II. The selection of the medoid node in each cluster is determined based on the node density in terms of relationships and distance from the center.
- III. The PAM and SVD++ methods are integrated into one model.

The cold-start users are considered in this study who have less than five and greater than zero ratings. The backbone of this model is employing a clustering technique using the PAM algorithm, which clusters users into several groups based on the computed distances between users. The distances were computed using social relations (explicit and implicit) and confidence values. The clustering process was applied two times: with and without confidence values. After acquiring users' clusters, the SVD++ method was adopted in each cluster to predict items for cold-start users. Moreover, The SGD optimization method was applied to reduce the regularization error. Five accuracy metrics were utilized to test the proposed model, which are MAE, RMSE, precision, recall, and f-measure. The results proved that the proposed method outperformed other studies. Furthermore, the results reported that using confidence values can involve more cold-start users and increase prediction accuracy as well.

7. A NOVEL OVERLAPPING METHOD TO ALLEVIATE THE COLD-START PROBLEM IN RECOMMENDATION SYSTEMS

Cold-start problem is one of the main problems in recommendation systems. It can be categorized into cold-start- users and cold start- items. Cold-start-users occur when a user rated items rarely or when a new user registers in a system. Although the previous chapter alleviated the cold-start-users problem, still lacks sufficient of accurate prediction. We think users may have several tastes, clustering users into groups means distributing users into groups based on their tastes. The limitation of the clustering, a user belongs to a single cluster regardless of his/her multiple tastes. In reality, a user likes to watch action movies and comedy movies. Therefore, such kind of user should be served by recommending items from different groups. Accordingly, this chapter is expanded the previous chapter by overcoming clustering limitations and handling the cold-start-user problem. A novel overlapping method is proposed by developing the Partitioning Around Medoids (PAM) clustering algorithm to implement an overlapping technique, then SVD++ method is employed for the prediction process. Moreover, the fifth research question is addressed in this chapter.

RQ5 How can detecting users' communities who belong to several clusters be established using a novel proposed overlapping method?

The rest of this chapter illustrates and answer of the mentioned research question.

7.1 An Overview of the Proposed Model

As mentioned above, a novel overlapping technique is utilized to tackle the cold-start-user problem. In the beginning, explicit and implicit social relations with confidence values are adopted to cluster users into groups using a traditional PAM clustering algorithm, and then a novel technique is applied to implement the joining between clusters. By applying this technique users who have various tastes could belong to different clusters and subsequently, their ratings are considered in many groups. Moreover, after creating the clusters and applying our novel overlapping technique, SVD++ method is employed for each cluster to predict items for the cold-start-users of this cluster. The proposed method is evaluated using Ciao and FilmTrust datasets.

Like the previous chapter, explicit and implicit social relations are exploited. Moreover, confidence values are used not only to boost prediction accuracy but to increase the number of served cold-start-users, especially in FilmTrust dataset. Combining social relations with confidence values, which can be extracted from rating information, is implemented to produce weight values. The weight values are employed to compute the distances among users and then clustered them into several groups/communities. Afterward, the average distance for each community is computed, and then the distance between each user in all other communities and the center of a particular cluster is computed. The distance is compared with the average distance of this cluster; if the comparison value is less or equal to the average distance of the cluster, a new user will be included in this cluster. The same process is implemented for all clusters. Finally, for each cluster, a set of cold start users is determined, later SVD++ method is applied to compute the prediction items for those users.

7.2 Methods and Data Sources

The proposed model involves applying several methods such as extraction of social relations, confidence values, applying a novel overlapping clustering algorithm, and SVD++ method. Each method requires specific sources. Therefore, social relations and confidence values are devoted to overlapping clustering technique. Rating information is used for SVD++ method.

Social relations consist of direct relations/trust relations and indirect relations/friend relations. For both types, there are explicit and implicit social relations. In this model, explicit and implicit relations for direct social relations are utilized. In addition, rating information involves explicit rating values and implicit feedback, which is extracted depending on explicit ratings. The following subsections explain the extraction of explicit social relations, implicit social relations, confidence values, PAM algorithm, and SVD++ method.

7.2.1 Extract explicit social relations

Explicit relations are obtained directly from a dataset. Usually this data includes two columns: trustor and trustee. Based on these columns, a binary user-user matrix will

be created, trustors users are the rows and trustee users are the columns. The ones values are placed when there are interactions between trustor and trustee users; otherwise place zero.

However, explicit relations numbers are few and most users will remain out of clusters as outliers. Therefore, implicit relations are extracted and will be added to explicit relations to increase the relations among users.

7.2.2 Extract implicit social relations

Implicit social relations are a substantial source used to enrich social networks with more information. These relations can be concluded on the basis of explicit relations via different techniques. The transitivity technique is followed to extract implicit trust relations. For example, assume that user a trusts user b , and user b trusts user c ; thus, there is a high likelihood of establishing a new relationship (implicit relation) between a and c .

After implicit and explicit trust relations are attained, a binary user-user matrix is created. Then, the similarity value between each pair of the explicit and/or implicit user is computed. The Jaccard binary measure is employed to compute the similarity using Equation (7.1).

$$jac_{u,v} = \frac{x}{x+y+z} \quad (7.1)$$

where x is the number of common trusted users between users u and v , and y and z are the number of trusted users of users u and v , respectively.

7.2.3 Confidence values

Confidence values are used to reduce numbers of outliers during the clustering process by exploiting users' ratings. Confidence values are essential information for datasets that have a low social relations ratio. FilmTrust dataset has poor social relations ratio. It has 1358 explicit relations and 73057 implicit relations. Thus, confidence values cover this limitation by exploiting rating values to create new user-user trust matrix.

The confidence value between users u and v and it is computed by the following Equation.

$$T_{u,v} = A_{u,v}/A_u \quad (7.2)$$

$A_{u,v}$ is the number of common items that were rated by users u and v , and A_u denotes the number of items rated by user u .

Then, both the similarity values and confidence values are incorporated using the following Equation:

$$w_{u,v} = \begin{cases} \frac{sim_{u,v} + T_{u,v}}{2}, & sim_{u,v} \neq 0, \text{ and } T_{u,v} \neq 0 \\ sim_{u,v}, & sim_{u,v} \neq 0, \text{ and } T_{u,v} = 0 \\ T_{u,v}, & T_{u,v} \neq 0, \text{ and } sim_{u,v} = 0 \\ 0, & otherwise \end{cases} \quad (7.3)$$

Where, $sim_{u,v}$ denotes the similarity between users u and v , which is computed using Equation (7.1).

Weight values are calculated using Equation (7.3). In this equation, social relations and confidence values are integrated for each pair of users. If a pair of users has a similarity value of social relations and confidence value, the first option of the Equation (7.3) implements. While if the pair has a similarity value only, the weight value assigns the similarity value. Whereas if the pair has trust value only, the weight assigns the trust value. Finally, if there is neither a trust value nor a similarity value, the weight will be zero.

After obtaining the weight values, A traditional PAM algorithm can be implemented now to cluster the users into several groups. The following section explains PAM algorithm.

7.2.4 Partitioning Around Medoids (PAM) algorithm

The clustering technique in a recommendation system is a process that distributes the users of a given dataset into many groups according to the distances between them. In other words, each set of similar users in terms of ratings, relations or any other features

are gathered together in one community. Subsequently, items that interest a particular cluster are recommended to a cold-start user who belongs to the that cluster.

PAM is a conventional partitioning clustering algorithm based on the k-medoids technique. A medoid or center is an object of a particular cluster where the distance between each object and the selected medoid is the minimum. The k-medoid is more robust than the k-means as it is less sensitive to outliers and noise.

Algorithm 1: PAM

Inputs: user-user matrix of distances values

Outputs: users clusters

- 1- Select k medoids randomly, where $k < n$ and n is the number of nodes in the whole dataset. Each medoid indicates the center of a cluster.
- 2- Apply the distance measure (the Euclidean distance, the Manhattan distance, or any other distance method) to find the distance (cost) between each point and the selected medoids.
- 3- Assign each node to the closest medoid.
- 4- Compute the total cost of all nodes.
- 5- Select a new medoid from the non-medoid objects randomly.
- 6- Compute the new cost ($cost_2$) using the similarity measure.
- 7- $Difference = cost_2 - cost_1$
- 8- If $difference < 0$, swap the new medoid with the old one.
- 9- Repeat steps 2-8 until the medoids do not change.

7.2.5 Novel overlapping method

After clustering users into groups, now a new overlapping method is implemented. For all clusters, an average distance is computed by dividing the summation of the cluster's distances by the numbers of all distances of this cluster. Then, for users outside the active cluster, the distance between the center of this cluster and all other users is computed, if the distance less than or equal the average of the active cluster, a new user will be added to this cluster. The following algorithm depicts the overlapping method of this study.

Algorithm 2: Novel overlapping clustering

The inputs: users' clusters, users' distances.

The output: users belong to multiple clusters.

1. For each cluster, compute the average of distances
2. For all users, compute the distance between them the center of the active cluster that utilized in step 1
3. Compare the output of step 2 with the output of step 1, if the result less or equal the distance of step 1, a new user is added to this cluster
4. Repeat steps 1, 2, and 3 for all clusters.

7.2.6 SVD++ method

SVD++ was produced by (Koren et al., 2008). It is an expanded of the SVD method, and the implicit rating information is the data that is considered in SVD++. Recently, many studies have utilized this approach (Jiao et al., 2019; Shi et al., 2020) due to its ability to enhance accuracy. The main idea of this method is reducing the amount of rating information (user-item matrix), where the new user matrix and the new item matrix are derived by detecting the latent features from the user-item matrix. Then the prediction value is computed as follows:

$$r_{u,i} = \mu + b_u + b_i + q^T \left(p_u + \frac{1}{\sqrt{|N_u|}} \sum_{j \in N_u} y_j \right) \quad (7.4)$$

$$b_i = \frac{(\sum_{u \in R(i)} r_{u,i} - \mu)}{\lambda_1 + |R_i|} \quad (7.5)$$

$$b_u = \frac{(\sum_{u \in R(u)} r_{u,i} - \mu - b_i)}{\lambda_2 + |R_u|} \quad (7.6)$$

Where μ is the mean value of all the entire ratings; and; b_u and b_i are the observed deviations from the μ of user u and item i , respectively. The number of items rated by user u is denoted as $|N_u|$, $|R_u|$ is the number of users that rated a specific item, $|R_i|$ is the number of items that were rated by a set of users, and y_j is the left orthogonal matrix of the implicit matrix. λ_1 and λ_2 are two regularization parameters.

7.3 Proposed Model

In this study, two main sources are employed to produce the proposed model. First, social relations and users' ratings are used for the clustering technique to group the users according to their relations. Second, rating information that indicates explicit and implicit feedback is utilized. The following steps are devoted to explaining the proposed method.

Step 1: Creating a binary user-user matrix from explicit social relations data.

Step 2: Extraction of implicit relations using transitive technique and adding the values of the implicit relations that are binary values to the previous binary user-user matrix.

Step 3: The similarity of explicit and implicit social relations is computed using Equation (7.1).

Step 4: To cluster the dataset more accurately, rating information is employed to compute confidence values between each pair of users using Equation (7.2). A new user-user matrix is created for the confidence values.

Step 5: The confidence values and similarity values are incorporated to compute weight values using Equation (7.3).

Step 6: After acquiring the weights, the distances are calculated by subtracting the weight values from one. Then, the PAM clustering algorithm is adopted to cluster the users according to the computed distances. Moreover, two conditions are considered to determine the medoids in the PAM algorithm. The first uses the greatest number of relations of the object to be a medoid. While the secondly finds the minimum distance between the medoid and the objects on a particular cluster.

Step 7: The average distance for each cluster is computed by dividing the summation of all distances in a cluster by the number of distances in the same cluster. Afterward, the distance between all users outside this cluster and the center inside this cluster is computed. If the distance less or equals the cluster's average distance, a new user is included in this cluster. The outcome of this step is overlapping, where many users belong to different clusters at the same time.

Step 8: The SVD++ method is applied in each cluster to compute the predictions for cold-start users by Equation (7.4). After applying SVD++ on each cluster, an average is computed to yield the last outcome. To enhance the prediction results, the stochastic gradient descent is applied by many studies to reduce the regularization error and to update the values of p_u , q_i , y_j , b_u , and b_i .

$$E = \sum_{(u,i) \in m} \left(r_{ui} - \mu - b_u - b_i - q^t \left(p_u + \frac{1}{\sqrt{|N_u|}} \sum_{j \in N_u} y_j \right) \right)^2 + \lambda (b_i^2 + b_u^2 + \|p\|^2 + \|q\|^2 + \|y_j\|^2) \quad (7.7)$$

$$p_u = p_u + \gamma(Eq_i - \lambda p_u) \quad (7.8)$$

$$q_i = q_i + \gamma \left(E \left(p_u + \frac{1}{\sqrt{|N_u|}} \sum_{j \in N_u} y_j \right) - \lambda q_i \right) \quad (7.9)$$

$$y_j = y_j + \gamma \left(E \left(\frac{q_i}{\sqrt{|N_u|}} \right) - \lambda y_j \right) \quad (7.10)$$

$$b_u = b_u + \gamma(E - \lambda b_u) \quad (7.11)$$

$$b_i = b_i + \gamma(E - \lambda b_i) \quad (7.12)$$

E is the regularization error value, $\lambda > 0$ is utilized to set the degree of the constraint used to avoid over-fitting, γ is the learning rate, and $\|x\|$ denotes the Frobenius norm. Figure 7.1 shows the details of the proposed method.

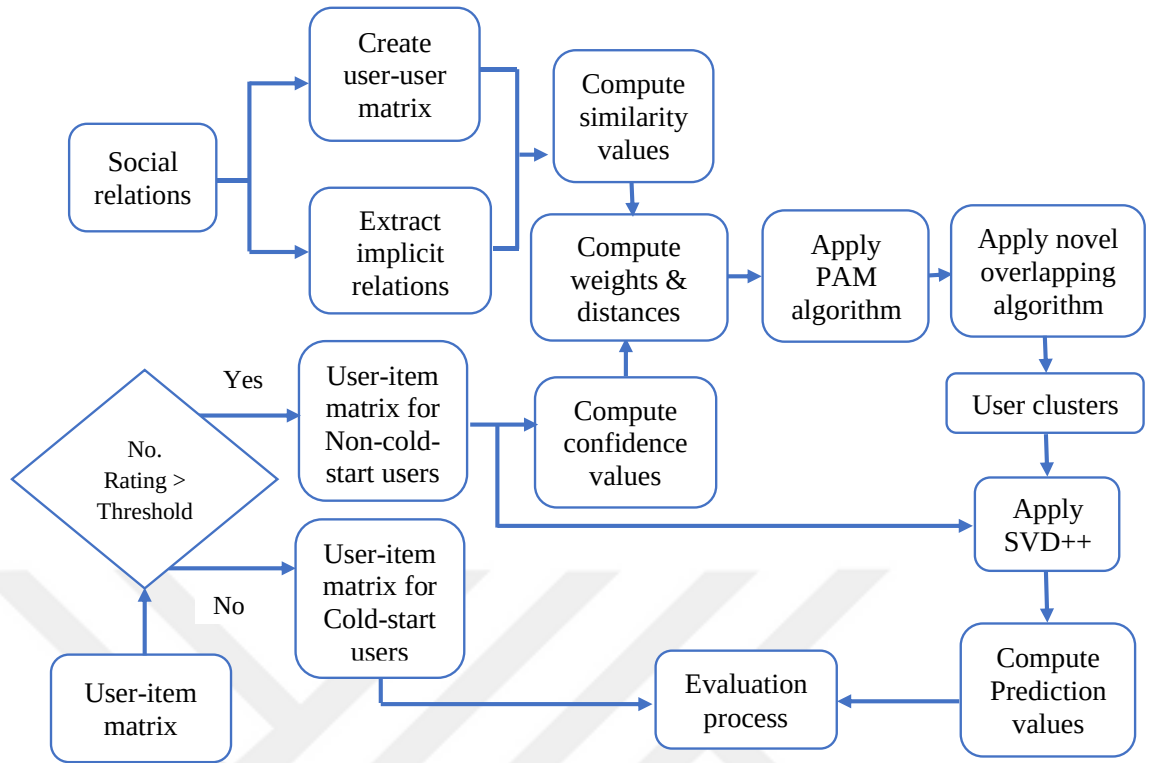


Figure 7.1. An overview of the proposed model.

7.4 Experimental Evaluation

In this section, the datasets used in this model are explained. The previous studies utilized in the comparison with our model are also demonstrated. Moreover, the results of the proposed model are discussed.

7.4.1 Description of the datasets

As in the previous chapter, Ciao and FilmTrust datasets are used to evaluate the proposed method. Ciao is a general item dataset that is comprised of 7375 users and 99746 items, and the rating scale is 1-5 (Tang et al., 2012). We reduced the number of the items to 22229 after removing items that have less than 2 ratings. Users who have less than five ratings are considered cold-start users. They are 749 in number. FilmTrust is specialized in movies and it involves 1508 users and 2071 items. The rating scale is 0.5-4 with a step size of 0.5 (Guo et al., 2013). Users who have less than six ratings are considered cold-start users. They are 338 in number. Table 7.1 shows the details of the datasets.

Table 7.1. Statistics of the FilmTrust and Ciao.

Dataset	Ciao	FilmTrust
# users	6767	1508
# items	22,229	2071
# ratings	185,759	35,494
# cold-start users	749	338
%Density	0.0012	1.14
# explicit relations	111780	1853
# implicit relations	54,056,070	73,057

7.4.2 Experimental setup

The implementation of this study relies on setting some parameters to achieve the best results. These parameters are shown in Table 7.2.

Table 7.2. Parameter setup.

Dataset	Ciao	FilmTrust
γ	0.01	0.03
λ	0.01	0.8
λ_1 and λ_2	6, 8	20, 15
d	5, 10	5, 10
Iterations	5	5

The proposed method is evaluated by comparing its results with those of several state-of-the-art approaches including RSTE (Ma et al., 2009), SocialMF (Jamali and Ester, 2010), SocialRec (Ma et al., 2011) and Trust SVD (Guo et al., 2015). Furthermore, the proposed method is compared with two recent studies in terms of MAE and RMSE. These are Trust ANLF (Parvin et al., 2019) and RTARS (Ahmadian et al., 2019). The benchmarks were achieved with the aforementioned studies under the same conditions. These conditions include the number of users, latent dimension values and the number of iterations. The testing was conducted for cold-start users. In this study, users were

considered to be cold-start users if they have less than 5 ratings for Ciao, and less or equal 5 ratings for FilmTrust.

7.4.3 Results and discussion

This model concentrates on alleviating the problem of cold-start-users. The datasets are divided into training and testing parts. All cold-start users were placed in the testing part and the rest in the training part. The following subsections explain the results of the proposed method in different measures.

7.4.3.1 Determining the best cluster

Tables 7.3 and 7.4 show the results of the MAE and RMSE for different numbers of clusters. Three samples of clusters are applied, 7,10 and 19 clusters are displayed in Ciao dataset, whereas 5,7 and 9 clusters are displayed in FilmTrust. In Table 7.3, the best results in terms of MAE and RMSE are reported when the number of clusters is 10. The results are 0.7099 for MAE and 0.9123 for RMSE. In Table 7.4, however, the best results are achieved when the number of clusters is 7. They are 0.5956 and 0.7686 for MAE and RMSE respectively. It can be concluded that neither too few nor too many clusters lead to the best results. Having few clusters may lead to considering some users as outliers and consequently, the accuracy is negatively affected. Likewise, having many clusters means adding more users, and using the tastes of those users mislead cold-start users and spoils the prediction accuracy. Accordingly, a reasonable number of clusters that include a proper ratio of users yield the best results.

Table 7.3. Accuracy results of various numbers of overlapping clusters for Ciao.

No. of Clusters	MAE	RMSE
7	0.7313	0.9325
10	0.7099	0.9123
19	0.7265	0.9216

Table 7.4. Accuracy results of various numbers of overlapping clusters for Film Trust.

No. of Clusters	MAE	RMSE
5	0.6397	0.8302
7	0.5956	0.7686
9	0.6554	0.8149

7.4.3.2 Benchmarking with our previous models

Tables 7.5 and 7.6 show the results of the proposed method and our previous models (MSRA-SVD, MSRA-SVD++ and PAM-SVD++) in terms of MAE and RMSE. The results are reported in two-dimension values (5 and 10). Moreover, Table 7.5 shows the results of the Ciao dataset and Table 7.6 shows the results of the FilmTrust dataset. As seen in both tables, the clustering and overlapping techniques have a significant impact to enhance the results. The improvement of the PAM-SVD++ model on MSRA-SVD++ in Table 7.5 is about 2.87% for RMSE and 1.73% for MAE when $d=5$. Similarly, there is an improvement when $d=10$ at both tables. Additionally, in Table 7.5, the overlapping technique is superior to all studies in terms of MAE and RMSE. The improvement of the proposed method is around 0.62% for MAE and 1.09% for RMSE when d equals 5. Likewise, the improvement is around 0.36% for MAE and 1.06% for RMSE when d value equals 10. In parallel, the results in Table 7.6 reveal that the proposed model again achieved the best results. In terms of RMSE, it outperformed all studies. The improvement is approximately 0.36% and 0.26% when d equals 5 and 10 respectively. In contrast, the proposed method is too close to the PAM-SVD++ model in terms of MAE. There is a tiny improvement to the proposed model of about 0.08% and 0.06% when d equals 5 and 10 respectively.

Table 7.5. Performance comparison with previous models for cold-start users in Ciao. Boldface marks best results and (*) indicates the second-best results.

	Metrics	MSRA-SVD	MSRA-SVD++	PAM-SVD++	Overlapping model
D=5	MAE	0.7662	0.7270	0.7144*	0.7100
	RMSE	0.9880	0.9499	0.9226*	0.9125
D=10	MAE	0.7658	0.7263	0.7126*	0.7099
	RMSE	0.9873	0.9494	0.9221*	0.9123

Table 7.6. Performance comparison with previous models for cold-start users in FilmTrust. Boldface marks best results and (*) indicates the second-best results.

	Metrics	MSRA-SVD	MSRA-SVD++	PAM-SVD++	Overlapping model
D=5	MAE	0.6826	0.6526	0.5958*	0.5953
	RMSE	0.8967	0.8627	0.7703*	0.7675
D=10	MAE	0.6821	0.6518	0.596*	0.5956
	RMSE	0.8965	0.8621	0.7706*	0.7686

7.4.3.3 Benchmarking with state-of-the-art methods

Tables 7.7 and 7.8 show the results of the proposed model (overlapping model) and the state-of-the-art that used social relations. The benchmarking is done using two different dimension values (5 and 10). Moreover, two metrics (MAE and RMSE) are utilized in this comparison. The overlapping technique is a successful way used in different applications. It is obvious that the overlapping model achieved the best results compared with the studies in Tables 7.7 and 7.8. The Boldface marks the best results, whereas the symbol (*) refers to the second-best results. In Table 7.7, our model exceeds the second-best results by 2.6% for MAE and 4.25% for RMSE when $d=5$. Furthermore, the improvement is about 0.85% for MAE and 1.69% for RMSE when $d=10$ at the same table. By the same token, the proposed model reported another outperformance in Table 7.8. It attains the best results in both metrics. The proposed model improved the results by around 0.61% and 0.84% for MAE and RMSE respectively when d equals 5. Likewise, the improvement that is accomplished by the

proposed method is 0.56% for MAE and 0.70% for RMSE when $d=10$. Consequently, the proposed overlapping model has a significant impact to enhance the results. Each part of users can assist the cold-start users to find their items of interest.

Table 7.7. Performance comparison for cold-start users in Ciao. Boldface implies best results and (*) indicates the second-best results.

	Metrics	RSTE	SocialRec	Trust SVD	Trust ANLF	Proposed model
D=5	MAE	0.957	0.789	0.729*	-----	0.7100
	RMSE	1.113	0.998	0.953*	-----	0.9125
D=10	MAE	0.803	0.730	0.721	0.716*	0.7099
	RMSE	1.014	1.031	0.962	0.928*	0.9123

Table 7.8. Performance comparison for cold-start users in FilmTrust. Boldface implies best results and (*) indicates the second-best results.

	Metrics	RSTE	SocialRec	Trust SVD	Trust ANLF	RTARS	Proposed model
D=5	MAE	0.680	0.670	0.661	-----	0.599*	0.5953
	RMSE	0.884	0.857	0.853	-----	0.774*	0.7675
D=10	MAE	0.674	0.668	0.663	0.607	0.599*	0.5956
	RMSE	0.900	0.897	0.853	0.784	0.774*	0.7686

7.4.3.4 Impact of different lambda values

Figures 7.2 and 7.3 show the results of the MAE and RMSE for different λ values ranging from 0.001 to 0.9. The results of both figures are reported when dimension value (d) equals 10. λ value is used to reduce the regularization error during the training phase. As seen in Figures 7.2 and 7.3, the results of MAE and RMSE are affected when λ value changes. In Figure 7.2, the MAE result starts with 0.7099 when λ equals 0.001, and then there is a tiny improvement when the λ value becomes 0.01. After 0.01 until 0.2 we can see the λ value return to 0.71. The best result (0.7093) is reported when the λ value equals 0.3. After 0.3 until 0.9, however, there is dramatically deteriorated in MAE results. The worst result is reported when the λ value is 0.9. Likewise, in Figure 7.3 we can note a similar design. The curve rises sharply to reach the worst result when the λ value is 0.9. The best result (0.9123) is obtained when the λ value is 0.01.

Figures 7.4 and 7.5 depict the results of MAE and RMSE for the FilmTrust dataset respectively. Again, the results are displayed for different λ values from 0.01 to 1. Unlike the scenario displayed in previous figures, the results improved when the λ value increases. The results at point 0.01 are 0.6132 for MAE and 0.7753 for RMSE. they decrease significantly to reach the best results (0.5956 at point 0.8 in Figure 7.4 and 0.7681 at point 0.6 in Figure 7.5). After point 0.8, however, the results decline to reach 0.5962 in Figure 7.4 and 0.7706 at point 1.

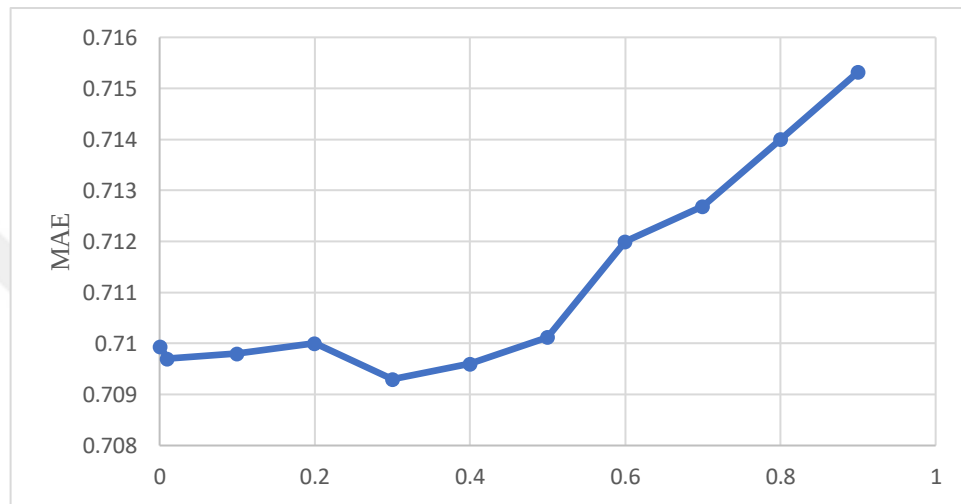


Figure 7.2. The MAE results for different lambda values in Ciao.

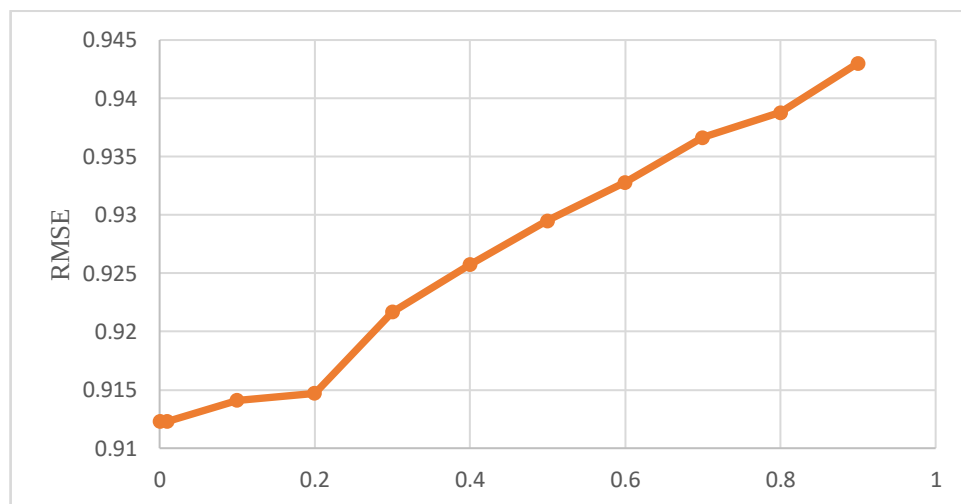


Figure 7.3. The RMSE results for different lambda values in Ciao.

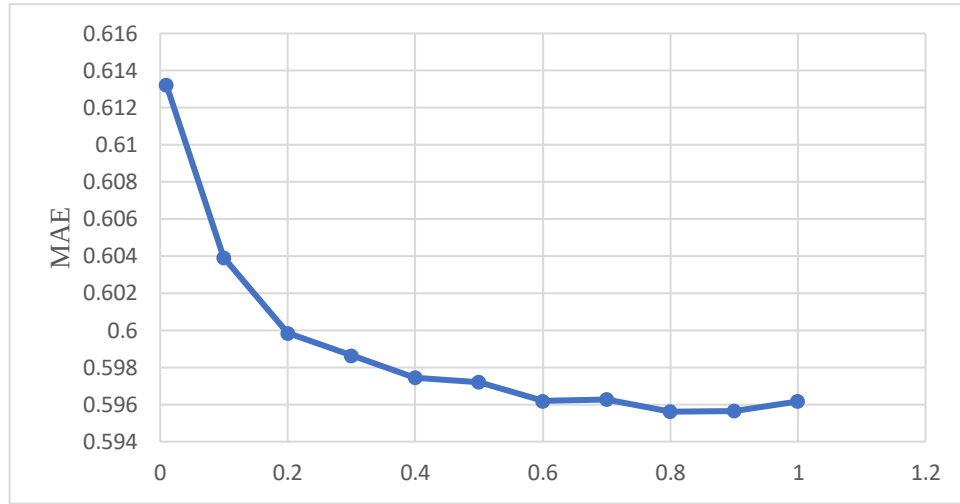


Figure 7.4. The MAE results for different lambda values in FilmTrust.

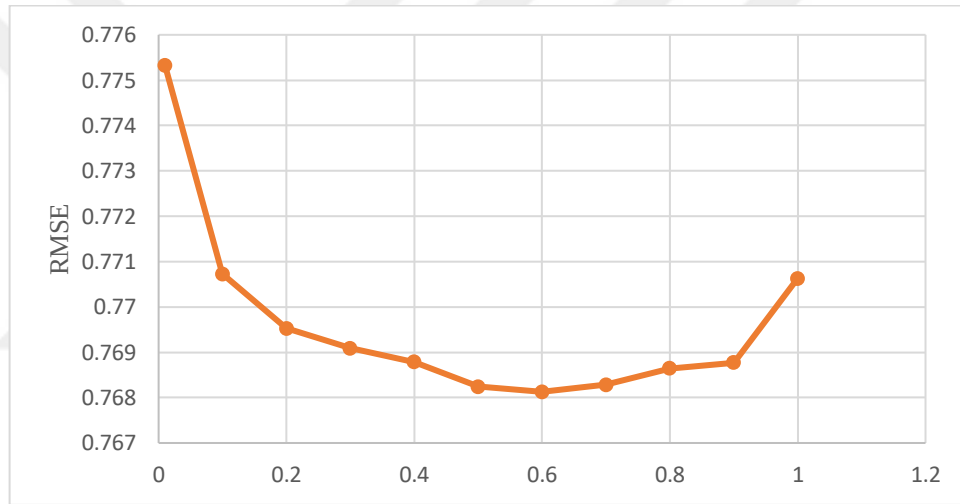


Figure 7.5. The RMSE results for different lambda values in FilmTrust.

7.4.3.5 Decision support system results

Tables 7.9, 7.10 and 7.11 show the results of precision, recall and f-measure respectively. The highest values refer to the best results, where the optimal result is one.

The results are reported for MSRA-SVD++, PAM-SVD++, and the proposed model in this chapter. Moreover, the outcomes are displayed in multi top-n recommended items (2, 3, and 4). Although the proposed method outperformed the PAM-SVD++ method in terms of MAE and RMSE, the latter achieved the best results in terms of precision, recall and f-measure. To justify these results, precision and recall metrics

measure the prediction accuracy of relevant items only (the rating value is greater than or equal to 3). There are predicted via a particular system and it is relevant in the testing part (true positive). So more relevant items are predicted in the PAM-SVD++ method than the proposed overlapping method. For example, the PAM-SVD++ method predicts 166 relevant items (for top-4 for all cold-start users). The proposed method, on the other hand, predicts 108 relevant items (for the same top-n items). This difference negatively affected the performance of the proposed method. However, the reason for the superiority of the proposed method in terms of MAE and RMSE metrics is that these metrics measure the predicted items by a system regardless of being relevant or irrelevant (all rating values). Accordingly, the proposed method predicts relevant and irrelevant items more precisely than the PAM-SVD++ method. In summary, the performance of the proposed method for all rating values is better than that of the PAM-SVD++ method. In contrast, the PAM-SVD++ method outperformed the proposed method in recognizing relevant items.

Table 7.9. Results of precision.

Dataset	Method	Top-2	Top-3	Top-4
Ciao	MSRA-SD++	0.4345	0.27	0.1921
	PAM-SVD++	0.4480	0.2731*	0.1972*
	Proposed model	0.4447*	0.2972	0.2048
FilmTrust	MSRA-SD++	0.2462	0.1587	0.1069
	PAM-SVD++	0.4826	0.3359	0.253
	Proposed model	0.3880*	0.2794*	0.2135*

Table 7.10. Results of recall.

Dataset	Method	Top-2	Top-3	Top-4
Ciao	MSRA-SD++	0.0987	0.1008	0.101
	PAM-SVD++	0.1064	0.1089	0.1528
	Proposed model	0.1019*	0.1055*	0.1058*
FilmTrust	MSRA-SVD++	0.11	0.1271	0.128
	PAM-SVD++	0.3121	0.3324	0.3337
	Proposed model	0.2767*	0.3030*	0.3138*

Table 7.11. Results of F-measure.

Dataset	Method	Top-2	Top-3	Top-4
Ciao	MSRA-SD++	0.1617	0.1468	0.1324
	PAM-SVD++	0.1996	0.1516*	0.1365
	Proposed model	0.1638*	0.1525	0.1356*
FilmTrust	MSRA-SD++	0.1605	0.1412	0.1165
	PAM-SVD++	0.3748	0.3341	0.2878
	Proposed model	0.3231*	0.2907*	0.2541*

7.5 Summary

In this chapter, the cold-start problem was alleviated by exploiting a novel overlapping clustering method. The last research question of this study is addressed.

RQ5 How can detecting users' communities who belong to several clusters be established using a novel proposed overlapping method?

The overlapping technique is expanded to the clustering technique. The limitation of the clustering technique has been tackled by adopting an overlapping clustering technique. Unlike the clustering technique, the overlapping technique means that users belong to several clusters at the same time. In this method, a novel overlapping was applied. This method was executed using the PAM algorithm to implement the clustering, which is done by exploiting social relations and confidence values (as explained in chapter 6). After acquiring users' clusters, the average distances were computed in each cluster. Afterward, the distance between all users and the computed distance of the clusters are compared. If the comparison result is less than or equals the average distance of a cluster, a new user is added to this cluster. The SVD++ method was then applied to all clusters to compute prediction values. The outcome was calculated by computing the average of MAE and RMSE for all clusters. The model was tested by two datasets: Ciao and FilmTrust. It can be noted that recommending items from various clusters achieves higher accuracy than recommending from a single cluster. These results confirm that users have multiple tastes and we considered this to improve the performance of the prediction. Consequently, the results of the proposed overlapping method outperformed our

previous clustering model (PAM-SVD++) and the-state-of-the-art methods in terms of MAE and RMSE. However, in terms of decision support system results, our clustering model (PAM-SVD++) is better than the current model in most cases. To justify, decision support system metrics consecrate the computation of the related items for users only. And according to this computation, the values of recall and precision are computed, so the PAM-clustering computed the related items more accurately than the proposed overlapping model. On the other hand, in terms of MAE and RMSE, these metrics compute the related and unrelated items for users, so the proposed overlapping model predicted items regardless of related or unrelated, it outperformed the PAM-SVD++ model in these metrics.



8. CONCLUSIONS AND FUTURE WORK

After explaining all models of this research, now, the conclusions of the study will be highlighted and some suggestions for further development will be made. Section 8.1 shows the conclusions and section 8.2 suggests future work.

8.1 Conclusions of the Study

According to the results acquired by the proposed models, we come to the following conclusions:

- 1- Social relations especially implicit social relations ones have a vital role to boost the prediction accuracy. Many algorithms have been used in this field. The second step of the MSRA algorithm is adopted in this study.
- 2- The prediction accuracy can further be enhanced by exploiting explicit and implicit social relations as well as explicit and implicit rating feedback and integrating them into the SVD/SVD++ method.
- 3- Social relations have a significant impact to improve the prediction accuracy when used to alleviate the cold-start problem. It is higher than their impact with the non-cold-start users.
- 4- Clustering users into several groups and determining the cold-start users of each cluster is better than working with them as one cluster. Users are clustered according to their social relations. Next, items are predicted to cold-start users for each cluster separately. This is better than predicting items to cold-start users from the whole dataset.
- 5- Using implicit social relations can reduce the outline nodes in the clustering process, where more nodes can be added in clusters.
- 6- For poor density social datasets, confidence values provide an optimal alternative source to include more users in the clusters. Consequently, the chance of predicting relevant items to cold-start users increases.

8.2 Future Work

Although this research enhanced and developed the performance of the RS, several issues remain to be considered in further studies. These include the following:

- 1- Designing a new model that uses contextual sources such as tag information along with social relations and rating feedback. This may enhance the performance of the RS.
- 2- Designing a hybrid model that uses a particular clustering algorithm and an optimization technique.
- 3- Finding another way to apply the overlapping technique rather than comparing the distance of all users with the average distance of a certain cluster.
- 4- All traditional similarity measures utilized rating feedback to compute the similarity for a pair of users. A hybrid similarity method that exploits another source along with the rating feedback can be used to enhance the prediction accuracy.
- 5- To develop the first and second models, contextual information like time can be exploited with social relations and integrated into the SVD++ method to enhance the prediction performance.

REFERENCES

- Adomavicius, G., and Tuzhilin, A., 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE transactions on knowledge and data engineering*, IEEE, 17, 6, 734–749.
- Ahmadian, S., Afsharchi, M., and Meghdadi, M., 2019. An effective social recommendation method based on user reputation model and rating profile enhancement, *Journal of Information Science*, SAGE Publications Sage UK: London, England, 45, 5, 607–642.
- Ahn, H. J., 2008. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, *Information Sciences*. Elsevier, 178, 1, 37–51.
- Al Falahi, K., Mavridis, N. and Atif, Y., 2012. Social networks and recommender systems: a world of current and future synergies, in *Computational Social Networks*. Springer, 445–465.
- Al-Sabaawi, A. M. A., Karacan, H. and Yenice, Y. E., 2020. Exploiting implicit social relationships via dimension reduction to improve recommendation system performance, *PLOS ONE Public Library of Science*, 15, 4, e0231457.
- Altingovde, I. S., Subakan, Ö. N. and Ulusoy, Ö., 2013. Cluster searching strategies for collaborative recommendation systems, *Information processing & management*, Elsevier, 49, 3, 688–697.
- Anand, D., 2011. Enhancing Accuracy of Recommender Systems Through Various Approaches to Local and Global Similarity Measures. Jawaharlal nehru university new delhi. doctorate's thesis, Jawaharlal Nehru University, School Of Computer And Systems Sciences, New Delhi.
- Bach, S. H., Broecheler, M., Huang, B. and Getoor, L., 2017. Hinge-loss markov random fields and probabilistic soft logic, *The Journal of Machine Learning Research*, JMLR, 18, 1, 3846–3912.
- Basu, C., Hirsh, H. and Cohen, W., 1998. Recommendation as classification: Using social and content-based information in recommendation, in *Aaai/iaai*, 714–720.
- Bedi, P. and Sharma, C., 2016. Community detection in social networks, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Wiley Online Library, 6, 3, 115–135.
- Bell, R. M. and Koren, Y., 2007. Improved neighborhood-based collaborative filtering, in *KDD cup and workshop at the 13th ACM SIGKDD international conference on knowledge discovery and data mining*, Citeseer, 7–14.

- Bergner, Y., Droschler, S., Rayyan, S., Seaton, D., Kortemeyer, G., et al., 2012. Model-Based Collaborative Filtering Analysis of Student Response Data: Machine-Learning Item Response Theory, International Conference on Educational Data Mining (EDM), Greece, 95-102.
- Bi, X., Qu, A., Wang, J. and Shen, X., 2016. A group-specific recommender system, Journal of the American Statistical Association, Taylor & Francis online, 112, 519, 1344–1353.
- Bilge, A. and Polat, H., 2012. An improved privacy-preserving DWT-based collaborative filtering scheme, Expert Systems with Applications, 39, 3, 3841–3854.
- Blei, D. M. Ng, A. Y. and Jordan, M. I., 2003. Latent Dirichlet Allocation, Journal of Machine Learning Research, 3, 993–1022.
- Bo, Y., Yu L., Dayou L. and Jiming L., 2013. Social collaborative filtering by trust, In Proceedings of the Twenty-Third international joint conference on Artificial Intelligence (IJCAI '13), AAAI Press, 2747–2753.
- Bobadilla, J., Hernando, A., Ortega, F. and Bernal, J., 2011. A framework for collaborative filtering recommender systems, Expert Systems with Applications. Elsevier, 38, 12, 14609–14623.
- Canny, J., 2002. Collaborative Filtering with Privacy via Factor Analysis, In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, 238-245.
- Cantador, I., Brusilovsky, P. and Kuflik, T., 2011. Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011)', in Proceedings of the fifth ACM conference on Recommender systems, 387–388.
- CarlKadie, J. B. D., 1998. Empirical analysis of predictive algorithms for collaborative filtering, Microsoft Research Microsoft Corporation One Microsoft Way Redmond, WA, Citeseer, 1-19.
- Celma, Ò., 2006. Foafing the music: Bridging the semantic gap in music recommendation, in International Semantic Web Conference, Springer, 927–934.
- Chen, H., Li, X. and Huang, Z., 2005. Link prediction approach to collaborative filtering, in Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05), IEEE, 141–142.
- Chen, J. and Yuan, B., 2006. Detecting functional modules in the yeast protein–protein interaction network, Bioinformatics, Oxford University Press, 22, 18, 2283–2290.

- Chen, R., Hua, Q., Chang, Y., Wang, B., Zhang, L., et al., 2018. A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks, *IEEE Access*, IEEE, 6, 64301–64320.
- Choi, K., Yoo, D., Kim, G. and Suh, Y., 2012. A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis, *electronic commerce research and applications*, Elsevier, 11, 4, 309–317.
- Cleverdon, C. W., Mills, J. and Keen, E. M., 1966. Factors determining the performance of indexing systems, Cranfield: College of Aeronautics, University of Michigan.
- Cohen, W. W., Ravikumar, P. and Fienberg, S. E., 2003. A Comparison of String Distance Metrics for Name-Matching Tasks, in *IIWeb*, 73–78.
- Cover, T. and Hart, P., 1967. Nearest neighbor pattern classification, *IEEE transactions on information theory*, IEEE, 13, 1, 21–27.
- Cristianini, N. and Shawe-Taylor, J., 2000. An introduction to support vector machines and other kernel-based learning methods. Cambridge university press, United Kingdom.
- Dai, B., Wang, J., Shen, X. and Qu, A., 2019. Smooth neighborhood recommender systems, *The Journal of Machine Learning Research*, JMLR, 20, 1, 589–612.
- Dang, T. A. and Viennet, E., 2013. Collaborative filtering in social networks: A community-based approach, in *2013 International Conference on Computing, Management and Telecommunications (ComManTel)*, IEEE, 128–133.
- Deerwester, S. Dumais, S. T., Furnas, G. W., Landauer, T. K. and Harshman, R., 1990. Indexing by latent semantic analysis, *Journal of the American society for information science*, Wiley Online Library, 41, 6, 391–407.
- Demir, Ö. Yakushev, A. R., Keddo, R. and Kallio, U., 2017. Item-item music recommendations with side information, *arXiv preprint arXiv:1706.00218*.
- De Meo, P., Ferrara, E. Fiumara, G. and Provetti A., 2013. Enhancing community detection using a network weighting strategy, *Information Sciences*. Elsevier, 222, 648–668.
- Durante, F., Sánchez, J. F. and Trutschnig, W., 2013. On the interrelation between Dempster–Shafer belief structures and their belief cumulative distribution functions, *Knowledge-Based Systems*. Elsevier, 52, 107–113.
- Feng, H. Tian, J., Wang, H. J. and Li, M., 2015. Personalized recommendations based on time-weighted overlapping community detection, *Information & Management*. Elsevier, 52, 7, 789–800.

- Fernández, Y. B., Arias, J. J. P., Nores, M. L., Solla, A. G. and Cabrer, M. R., 2006. AVATAR: an improved solution for personalized TV based on semantic inference, *IEEE transactions on Consumer Electronics*, IEEE, 52, 1, 223–231.
- Finch, H., 2005. Comparison of distance measures in cluster analysis with dichotomous data, *Journal of Data Science*, 3, 1, 85–100.
- Fortunato, S., 2010. Community detection in graphs, *Physics reports*. Elsevier, 486, 3, 75–174.
- George, T. and Merugu, S., 2005. A scalable collaborative filtering framework based on co-clustering, in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, Houston, TX, 2005, IEEE.
- Ghazanfar, M. A. and Prügel-Bennett, A., 2014. Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems, *Expert Systems with Applications*, Elsevier, 41, 7, 3261–3275.
- Girvan, M. and Newman, M. E. J., 2002. Community structure in social and biological networks, *Proceedings of the national academy of sciences*, National Acad Sciences, 99, 12, 7821–7826.
- Goldberg, K., Theresa R., Dhruv G. and Chris P., 2001. Eigentaste: A constant time collaborative filtering algorithm, *information retrieval*, Springer, 4, 2, 133–151.
- Golub, G. H. and Reinsch, C., 1970. Singular value decomposition and least squares solutions, *Numer Math*, 14, 403–420.
- Gose, E., Johonsonbrugh, R. and Jost, S., 2005. Pattern recognition and image analysis, New Delhi, PHI learning.
- Guo, G. R., Zhang, J., Thalmann, D., Basu, A. and Yorke-Smith, N., 2014. From ratings to trust: an empirical study of implicit trust in recommender systems, in *Proceedings of the 29th annual acm symposium on applied computing*, 248–253.
- Guo, G., Zhang, J. and Thalmann, D., 2012. A simple but effective method to incorporate trusted neighbors in recommender systems, in *International conference on user modeling, adaptation, and personalization*, Springer, 114–125.
- Guo, G., Zhang, J. and Yorke-Smith, N., 2013. A novel bayesian similarity measure for recommender systems, in *Twenty-Third International Joint Conference on Artificial Intelligence*, Beijing, China, 2619–2625.
- Guo, G., Zhang, J. and Yorke-Smith, N., 2015. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings, in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 123–129.

- Guo, G., Zhang, J. and Yorke-Smith, N., 2016. A novel recommendation model regularized with user trust and item ratings, *IEEE transactions on knowledge and data engineering*, IEEE, 28, 7, 1607–1620.
- Gupta, M., Kumar, P. and Bhasker, B., 2017. Personalized item ranking from implicit user feedback: A heterogeneous information network approach, *Pacific Asia Journal of the Association for Information Systems*, 9, 2, p. 3.
- Herlocker, J. L., Konstan, J. A., Terveen, L. G. and Riedl, J. T., 2004. Evaluating collaborative filtering recommender systems, *ACM Transactions on Information Systems (TOIS)*. ACM New York, NY, USA, 22, 1, 5–53.
- Herlocker, J. L., Konstan, J. A., Borchers, A. and Riedl, J., 2017. An algorithmic framework for performing collaborative filtering, in *ACM SIGIR Forum*, ACM New York, NY, USA, 227–234.
- Hernando, A., Bobadilla, J., Ortega, F. and Tejedor, J., 2013. Incorporating reliability measurements into the predictions of a recommender system, *Information Sciences*, Elsevier, 218, 1–16.
- Hofmann, T., 2003. Collaborative filtering via gaussian probabilistic latent semantic analysis, in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, 259–266.
- Huo, Y., Yongfeng, H., Bilian, C., Jing T. and Yifeng Z., 2020. Privacy-preserving Point-of-Interest Recommendation Based on Geographical and Social Influence, *Information Sciences*, Elsevier, 543, 2021, 202–218.
- Hussein, M., Naji, H. and Bhaya, W., 2016. Influential Nodes Based Alleviation of User Cold-Start Problem in Recommendation System, *Research Journal of Applied Sciences*, 11, 1107–1114.
- Jalili, M., Ahmadian, S., Izadi, M., Moradi, P. and Salehi, M., 2018. Evaluating collaborative filtering recommender algorithms: a survey, *IEEE access*, IEEE, 6, 74003–74024.
- Jamali, M. and Ester, M., 2010. A matrix factorization technique with trust propagation for recommendation in social networks, in *Proceedings of the fourth ACM conference on Recommender systems*, 135–142.
- Jannach, D., Zanker, M., Felfernig, A. and Friedrich, G., 2010. *Recommender systems: an introduction*. Cambridge University Press, United Kingdom.
- Järvelin, K. and Kekäläinen, J., 2002. Cumulated gain-based evaluation of IR techniques, *ACM Transactions on Information Systems (TOIS)*. ACM New York, NY, USA, 20, 4, 422–446.
- Jiao, J., Zhang, X., Li, F. and Wang, Y., 2019. A Novel Learning Rate Function and Its Application on the SVD++ Recommendation Algorithm, *IEEE Access*, IEEE, 8, 14112–14122.

- Jiawei, H. and Micheline, K., 2006. *Data Mining: Concepts and Techniques*. 2nd edition ed, Elsevier, United States of America.
- Jones, K. S., 1972. A statistical interpretation of term specificity and its application in retrieval, *Journal of documentation*, MCB UP Ltd.
- Kim, M.-G. and Kim, K., 2016. Recommender Systems using SVD with Social Network Information, 22, 4, 1–18.
- Koohi, H. and Kiani, K., 2016. User based Collaborative Filtering using fuzzy C-means, *Measurement*, Elsevier, 91, 134–139.
- Koren, Y., 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model, *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 08*, 426–434.
- Koren, Y. and Bell, R., 2015. Advances in collaborative filtering, in *Recommender systems handbook*, Springer, Boston, MA, 77–118.
- Kouki, P., Fakhraei, S., Foulds, J., Eirinaki, M. and Getoor, L., 2015. Hyper: A flexible and extensible probabilistic framework for hybrid recommender systems, in *Proceedings of the 9th ACM Conference on Recommender Systems*, 99–106.
- Krulwich, B., 1997. Lifestyle finder: Intelligent user profiling using large-scale demographic data, *AI magazine*, 18, 2, p. 37.
- Ku, Y.-C., Tai, Y.-M. and Chan, C.-H., 2016. Effects of Product Type and Recommendation Approach on Consumers' Intention to Purchase Recommended Products, *ICEIS 2013 - Proceedings of the 15th International Conference on Enterprise Information Systems*, 2, 475–480.
- Lam, S. K. and Riedl, J., 2004. Shilling recommender systems for fun and profit, in *Proceedings of the 13th international conference on World Wide Web*, 393–402.
- Lam, X. N., Vu, T., Le T. D. and Duong, A. D. 2008. Addressing cold-start problem in recommendation systems, in *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, 208–211.
- Lancichinetti, A., Fortunato, S. and Kertész, J., 2009. Detecting the overlapping and hierarchical community structure in complex networks, *New journal of physics*, IOP Publishing, 11, 3, 1-20.
- Lee, D. D. and Seung, H. S., 1999. Learning the parts of objects by non-negative matrix factorization, *Nature Publishing Group*, 401, 6755, 788–791.
- Lee, S. K., Cho, Y. H. and Kim, S. H., 2010. Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations, *Information Sciences*, Elsevier, 180, 11, 2142–2155.

- Lee, T. Q., Park, Y. and Park, Y.-T., 2007. A similarity measure for collaborative filtering with implicit feedback, in International Conference on Intelligent Computing ICIC 2007, Springer, Berlin, Heidelberg, 385–397.
- Leskovec, J., Rajaraman, A. and Ullman, J. D., 2014. Mining of Massive Datasets, Cambridge University Press, United Kingdom.
- Li, Q. and Kim, B. M., 2003. Clustering approach for hybrid recommender system, in Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003), IEEE, 33–38.
- Linden, G., Smith, B. and York, J., 2003. Amazon. com recommendations: Item-to-item collaborative filtering, IEEE Internet computing, IEEE, 7, 1, 76–80.
- Liu, J., Wu, C. and Liu, W., 2013. Bayesian probabilistic matrix factorization with social relations and item contents for recommendation, Decision Support Systems, Elsevier, 55, 3, 838–850.
- Liu, Y. and Kou, Z., 2007. Predicting who rated what in large-scale datasets, ACM SIGKDD Explorations Newsletter, ACM New York, NY, USA, 9, 2, 62–65.
- Lops, P., De Gemmis, M. and Semeraro, G., 2011. Content-based recommender systems: State of the art and trends, in Recommender systems handbook, Springer, Boston, MA, 73–105.
- Lu, J., Wu, D., Mao, M., Wang, W. and Zhang, G., 2015. Recommender system application developments: a survey, Decision Support Systems, Elsevier, 74, 12–32.
- Lu, Z., Dou, Z., Lian, J., Xie, X. and Yang, Q., 2015. Content-based collaborative filtering for news topic recommendation, in AAAI, 217–223.
- Ma, H., Zhou, D., Liu, C., Lyu, M. R. and King, I., 2011. Recommender systems with social regularization, in Proceedings of the fourth ACM international conference on Web search and data mining, New York, USA, 287–296.
- Ma, H., King, I. and Lyu, M. R., 2007. Effective missing data prediction for collaborative filtering, in Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, 39–46.
- Ma, H., King, I. and Lyu, M. R., 2009. Learning to recommend with social trust ensemble, in Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, 203–210.
- McCarthy, J. F. and Anagnost, T. D., 1998. MusicFX: an arbiter of group preferences for computer supported collaborative workouts, in Proceedings of the 1998 ACM conference on Computer supported cooperative work, 363–372.

- McCrae, J., Piatek, A. and Langley, A., 2004. Collaborative filtering, [http:// www.imperialviolet.org](http://www.imperialviolet.org).
- Melville, P., Mooney, R. J. and Nagarajan, R., 2002. Content-boosted collaborative filtering for improved recommendations, *Aaai/iaai*, 23, 187–192.
- Miranda, T., Claypool, M., Gokhale, A., Mir, T., Murnikov, P., et al. .1999. Combining content-based and collaborative filters in an online newspaper, in *In Proceedings of ACM SIGIR Workshop on Recommender Systems*, Citeseer.
- Mnih, A. and Salakhutdinov, R. R., 2008. Probabilistic matrix factorization, in *Advances in neural information processing systems*, 1257–1264.
- Moffat, A. and Zobel, J., 2008. Rank-biased precision for measurement of retrieval effectiveness, *ACM Transactions on Information Systems (TOIS)*. ACM New York, NY, USA, 27, 1, 1–27.
- Mooney, R. J. and Roy, L., 2000. Content-based book recommending using learning for text categorization, in *Proceedings of the fifth ACM conference on Digital libraries*, San Antonio, Texas, USA, 195–204.
- Moradi, P. and Ahmadian, S., 2015. A reliability-based recommendation method to improve trust-aware recommender systems, *Expert Systems with Applications*. Elsevier, 42, 21, 7386–7398.
- Newman, M. E. J., 2001. Clustering and preferential attachment in growing networks, *Physical review E. APS*, 64, 2, 1-13.
- Newman, M. E. J., 2004. Fast algorithm for detecting community structure in networks, *Physical review E, APS*, 69, 6, 1-5.
- Nilashi, M., 2016. An overview of data mining techniques in recommender systems, *Journal of Soft Computing and Decision Support Systems*, 3, 6, 16–44.
- O’Doherty, D., Jouili, S. and Van Roy, P., 2012. Towards trust inference from bipartite social networks, in *Proceedings of the 2nd ACM SIGMOD Workshop on Databases and Social Networks*, Scottsdale, Arizona 13–18.
- O’Mahony, M., Hurley, N., Kushmerick, N. and Silvestre, G. 2004. Collaborative recommendation: A robustness analysis, *ACM Transactions on Internet Technology (TOIT)*, ACM New York, NY, USA, 4, 4, 344–377.
- Palla, G., Derényi, I. Farkas, I. and Vicsek, T., 2005. Uncovering the overlapping community structure of complex networks in nature and society, *nature*. Nature Publishing Group, 435, 7043, 814–818.
- Papagelis, M., Rousidis, I., Plexousakis, D. and Theoharopoulos, E., 2005. Incremental collaborative filtering for highly-scalable recommendation algorithms, in *International Symposium on Methodologies for Intelligent Systems*. Springer, Berlin, Heidelberg, 553–561.

- Parvin, H., Moradi, P., Esmaeili S. and Qader, N. N., 2019. A scalable and robust trust-based nonnegative matrix factorization recommender using the alternating direction method, *Knowledge-Based Systems*, Elsevier, 166, 92–107.
- Patel, B., Desai, P. and Panchal, U., 2017. Methods of recommender system: a review, in 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, IEEE, 1–4.
- Paterek, A., 2007. Improving regularized singular value decomposition for collaborative filtering, in *Proceedings of KDD cup and workshop*, 5–8.
- Pazzani, M. and Billsus, D., 1997. Learning and revising user profiles: The identification of interesting web sites, *Machine learning*. Springer, 27, 3, 313–331.
- Pazzani, M. J., 1999. A framework for collaborative, content-based and demographic filtering, *Artificial intelligence review*, Springer, 13, 393–408.
- Pham, M. C., Cao, Y., Klamma, R. and Jarke, M., 2011. A Clustering Approach for Collaborative Filtering Recommendation Using Social Network Analysis, *Journal Of Universal Computer Science*, 17, 4, 1–21.
- Psorakis, I., Roberts, S., Ebden, M. and Sheldon, B., 2011. Overlapping community detection using bayesian non-negative matrix factorization, *Physical Review E*. APS, 83, 6, p. 66114.
- Qi, L. et al. Zhang, Z., Dou, W., Hu, C. and Yang C., 2018. A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment, *Future Generation Computer Systems*. Elsevier, 88, 636–643.
- Quinlan, J. R., 1986. Induction of decision trees, *Machine learning*, Springer, 1, 1, 81–106.
- Ramakrishnan, N. Keller, B., Mirza, B., Grama, A. and Karypis, G., 2001. When being weak is brave: Privacy in recommender systems, *arXiv preprint cs/0105028*.
- Reafee, W., Salim, N. and Khan, A., 2016. The power of implicit social relation in rating prediction of social recommender systems, *PloS one*, Public Library of Science, 11, 5: e0154848.
- Reddy, M. S. and Adilakshmi, T., 2014. Music recommendation system based on matrix factorization technique-SVD, in 2014 International Conference on Computer Communication and Informatics, IEEE, 1–6.
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J., 1994. GroupLens: an open architecture for collaborative filtering of netnews, in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 175–186.

- Resnick, P. and Varian, H. R., 1997. Recommender systems, *Communications of the ACM*. ACM New York, NY, USA, 40, 3, 56–58.
- Rezaeimehr, F., Moradi P., Ahmadian, S. and Qader, N. N., 2018. TCARS: Time-and community-aware recommendation system, *Future Generation Computer Systems*, Elsevier, 78.
- Ricci, F., Rokach, L., Shapira, B. and Kantor, P .B., 2011. Introduction to recommender systems handbook, in *Recommender systems handbook*, Springer, 1–35.
- Ricci, F., Rokach, L. and Shapira, B., 2015. *Recommender Systems : Introduction and Challenges*.
- Salton, G. and McGill, M., 1983. *Introduction to modern information retrieval*, McGraw-Hill, New York.
- Sánchez-Moreno, D., González, A. B. G., Vicente M. D. M., Batista V. F. L. and Moreno-García M. N., 2016. A collaborative filtering method for music recommendation using playing coefficients for artists and users, *Expert Systems with Applications*, Elsevier, 66, pp. 234–244.
- Sarwar, B., Karypis, G., Konstan, J. A., and Riedl, J. T., 2000. Application of dimensionality reduction in recommender system-a case study, *Minnesota Univ Minneapolis Dept of Computer Science*.
- Sarwar, B., Karypis, G., Konstan, J. and Riedl, T., 2001. Item-based collaborative filtering recommendation algorithms, in *Proceedings of the 10th international conference on World Wide Web*, 285–295.
- Sarwar, B., Karypis, G., Konstan, J. and Riedl, T., 2002. Incremental singular value decomposition algorithms for highly scalable recommender systems, in *Fifth international conference on computer and information science*, Citeseer, 27–28.
- Sarwar, B., Karypis, G., Konstan, J. and Riedl, T., 2002. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering, in *Proceedings of the fifth international conference on computer and information technology*, 291–324.
- Semeraro, G., Lops, P., Basile, P. and Gemmis M., 2009. Knowledge infusion into content-based recommender systems, in *Proceedings of the third ACM conference on Recommender systems*, 301–304.
- Shi, W., Wang, L. and Qin, J., 2020. User Embedding for Rating Prediction in SVD++-Based Collaborative Filtering, *Symmetry, Multidisciplinary Digital Publishing Institute*, 12, 1, p 121.
- Shi, Y., Larson, M. and Hanjalic, A., 2009. Exploiting user similarity based on rated-item pools for improved user-based collaborative filtering, in *Proceedings of the third ACM conference on Recommender systems*, 125–132.

- Shi, Y. U., Larson, M. and Hanjalic, A., 2014. Collaborative Filtering beyond the User-Item Matrix : A Survey of the State of the Art and Future Challenges, *ACM Computing Surveys (CSUR)*, 47, 1, 1–45.
- Si, L. and Jin, R., 2003. Flexible mixture model for collaborative filtering, in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 704–711.
- Smyth, B. and Cotter, P., 2000. A personalized television listings service, *Communications of the ACM*. ACM New York, NY, USA, 43, 8, 107–111.
- Son, J., Choi, W. and Choi, S.-M., 2020. Trust information network in social Internet of things using trust-aware recommender systems, *International Journal of Distributed Sensor Networks*, SAGE Publications Sage UK: London, England, 16, 4, 1-12.
- Son, L. H., 2016. Dealing with the new user cold-start problem in recommender systems: A comparative review, *Information Systems*. Pergamon-elsevier science ltd the boulevard, langford lane, kidlington , 58, 87–104.
- Su, X., Greiner, R., Khoshgoftaar, T. M. and Zhu, X., 2007. Hybrid collaborative filtering algorithms using a mixture of experts, in *IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*, IEEE, 645–649.
- Su, X. and Khoshgoftaar, T. M., 2009. A survey of collaborative filtering techniques, *Advances in artificial intelligence*, Hindawi, 2009, 1-19.
- Sun, H.-F., Chen J.-L., Yu, G. and Liu, C.-C., 2012. JacUOD: a new similarity measurement for collaborative filtering, *Journal of Computer Science and Technology*, Springer, 27, 6, 1252–1260.
- Sun, Z., Han, L. H., Huang, L., Wang, W., Zeng, et al., 2015. Recommender systems based on social networks, *Journal of Systems and Software*, Elsevier, 99, 109–119.
- Suryana, N., Hanafi, H. and Basari, A. S., 2018. An understanding and approach solution for cold start problem associated with recommender system: a literature review, *Journal of Theoretical & Applied Information Technology*, 96, 9, 2677-2695.
- Tang, J., Gao, H., Liu, H. and Sarma, A., D., 2012. eTrust: Understanding trust evolution in an online world, in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 253–261.
- Tsai, C.-F. and Hung, C., 2012. Cluster ensembles in collaborative filtering recommendation, *Applied Soft Computing*, Elsevier, 12, 4, 1417–1425.
- Ungar, L. H. and Foster, D. P., 1998. Clustering methods for collaborative filtering, in *AAAI workshop on recommendation systems*, Menlo Park, CA, 114–129.

- Unger, M., Bar, A., Shapira, B. and Rookach L., 2016. Towards latent context-aware recommendation systems, *Knowledge-Based Systems*. Elsevier B.V., 104, 1–14.
- Wang, H., Wang, N. and Yeung, D.-Y., 2015. Collaborative deep learning for recommender systems, in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1235–1244.
- Wang, J., De Vries, A. P. and Reinders, M. J. T., 2006. Unifying user-based and item-based collaborative filtering approaches by similarity fusion, in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 501–508.
- Wasid, M. and Ali, R., 2018. An improved recommender system based on multi-criteria clustering approach, *Procedia Computer Science*. Elsevier, 131, 93–101.
- Wei, J., He, J., Chen, K., Zhou Y. and Tang, Z., 2017. Collaborative filtering and deep learning based recommendation system for cold start items, *Expert Systems with Applications*, Elsevier, 69, 29–39.
- Wu, Z. and Li, Y. 2014. Link prediction based on multi-steps resource allocation, in *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, IEEE, 355–360.
- Xie, F., Chen, Z., Shang, J., Huang W. and Li J., 2015. Item similarity learning methods for collaborative filtering recommender systems, in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*. Gwangju, 896–903.
- Xue, G.-R., Lin, C., Yang, Q., Xi, W., Zeng, H.-J., Yu, Y. and Chen, Z., 2005. Scalable collaborative filtering using cluster-based smoothing, *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '05*, 114–121.
- Yang, B., Liu, D. and Liu, J., 2010. Discovering communities from social networks: Methodologies and applications, in *Handbook of social network technologies and applications*. Springer, 331–346, Springer, Boston, MA.
- Yang, J., Yang J., Kim J., Kim W. and Kim Y. H., 2012. Measuring user similarity using electric circuit analysis: Application to collaborative filtering, *PloS one*, Public Library of Science, 7, 11, p. e49126.
- Yang, X., Steck, H., Guo, Y. and Liu, Y., 2012. On top-k recommendation using social networks, in *Proceedings of the sixth ACM conference on Recommender systems*, 67–74.
- Yang, X., Guo, Y. and Liu, Y., 2013. Bayesian-inference-based recommendation in online social networks, *IEEE Transactions on Parallel and Distributed Systems*, IEEE, 24, 4, 642–651.

- Yu, J., Gao, M., Rong, W., Song, Y. and Xiong, Q., 2017. A social recommender based on factorization and distance metric learning, *IEEE Access*, IEEE, 5, 21557–21566.
- Yu, J., Gao, M. and Song, Y., 2017. Connecting factorization and distance metric learning for social recommendations, in *International Conference on Knowledge Science, Engineering and Management*, Springer, 389–396.
- Yu, J. and Gao, M., Li, J., Yin, H. and Liu, H., 2018. Adaptive Implicit Friends Identification over Heterogeneous Network for Social Recommendation, 357–366.
- Yu, P., Lin, L. and Wang, J., 2017. A novel framework to alleviate the sparsity problem in context-aware recommender systems, *New Review of Hypermedia and Multimedia*, Taylor & Francis, 23, 2, 141–158.
- Zhang, J. and Piramuthu, S., 2018. Product recommendation with latent review topics, *Information Systems Frontiers*. Springer, 20, 3, 617–625.
- Zhang, S., Wang, W., Ford, J., Makedon, F. and Pearlman, J., 2005. Using singular value decomposition approximation for collaborative filtering, in *Seventh IEEE International Conference on E-Commerce Technology (CEC'05)*, Munich, Germany, 257–264.
- Zhang, Y., Chen, W. and Yin, Z., 2013. Collaborative filtering with social regularization for TV program recommendation, *Knowledge-Based Systems*, Elsevier, 54, 310–317.
- Zhao, F. and Guo, Y., 2017. Learning Discriminative Recommendation Systems with Side Information, in *IJCAI*, 3469–3475.
- Zhi-Xiao, W., Li Z.-C., Ding, X.-F. and Tang, J.-H., 2016. Overlapping community detection based on node location analysis, *Knowledge-Based Systems*, Elsevier, 105, 225–235.
- Zhou, W., Wen, J., Xiong, Q., Gao, M. and Zeng, J., 2016. SVM-TIA a shilling attack detection method based on SVM and target item analysis in recommender systems, *Neurocomputing*, Elsevier, 210, 197–205.
- Zhu, T., Greiner, R. and Häubl, G., 2003. Learning a model of a web user's interests, in *International Conference on User Modeling*, Springer, Berlin, Heidelberg, 65–75.

CURRICULUM VITAE

Name and Surname : Ali Mohsin Ahmed AL-SABAAWI

EDUCATION INFORMATION

Undergraduate : Mosul University, College of computer science and Mathematics, 1998-2003

Master : University Technology Malaysia, department of computer science, 2011-2013

Doctorate : Aksaray University, department of electrical-electronic and computer engineering, 2017-2021

PROFESSIONAL EXPERIENCE AND AWARDS

1. Award of the published paper from International Journal of Computer Science Issues (IJCSI) in 2012
2. Award of the excellent students from University Technology Malaysia (UTM) in 2013

PUBLICATIONS FROM THE THESIS

1. Ali M. Ahmed Al-Sabaawi, H. Karacan, Y.E. Yenice, Exploiting implicit social relationships via dimension reduction to improve recommendation system performance, PLoS One. 15, (2020) e0231457. <https://doi.org/10.1371/journal.pone.0231457>.
2. Ali M. A. Al-Sabaawi, H. Karacan, and Y. E. Yenice, Two models based on social relations and SVD++ method for recommendation system.vol. 15, No. 01, 2021.
3. Ali M. A. Al-Sabaawi, H. Karacan, and Y. E. Yenice, "SVD++ and Clustering Approaches to Alleviate the Cold-Start Problem for Recommendation Systems", in International Journal of Innovative Computing, Information and Control, Volume 17, Number 2, (tentative) April 2021.
4. Ali M. Al-Sabaawi , H. Karacan, Y.E. Yenice. " A Novel Overlapping Method To Alleviate The Cold-Start Problem In Recommendation Systems " International Journal of Software Engineering and Knowledge Engineering. **Under review**.
5. Ali M. Al-Sabaawi , H. Karacan, Y.E. Yenice, Collaborative Filtering: Review, Pamukkale Journal University of Engineering Sciences. **Under review**.