A MULTI-OBJECTIVE RECOMMENDATION SYSTEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MAKBULE GÜLÇİN ÖZSOY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

JUNE 2016

Approval of the thesis:

A MULTI-OBJECTIVE RECOMMENDATION SYSTEM

submitted by **MAKBULE GÜLÇİN ÖZSOY** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering** _____

Prof. Dr. Faruk Polat
Supervisor, **Computer Engineering Department, METU** _____

Prof. Dr. Reda Alhajj
Co-supervisor, **Dept. of Computer Science, Uni. of Calgary** _____

Examining Committee Members:

Prof. Dr. Fazlı Can
Computer Engineering Department, Bilkent University _____

Prof. Dr. Faruk Polat
Computer Engineering Department, METU _____

Prof. Dr. İsmail Hakkı Toroslu
Computer Engineering Department, METU _____

Prof. Dr. Ahmet Coşar
Computer Engineering Department, METU _____

Assoc. Prof. Dr. Tansel Özyer
Computer Engineering Department, TOBB ETU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: MAKBULE GÜLÇİN ÖZSOY

Signature :

ABSTRACT

A MULTI-OBJECTIVE RECOMMENDATION SYSTEM

Özsoy, Makbule Gülçin
Ph.D., Department of Computer Engineering
Supervisor : Prof. Dr. Faruk Polat
Co-Supervisor : Prof. Dr. Reda Alhajj

June 2016, 198 pages

Recommendation systems suggest items to the user by estimating their preferences. Most of the recommendation systems are based on single criterion, such that they evaluate items based on overall rating. In order to give more accurate recommendations, a recommendation system can take advantage of considering multiple criteria. Beside combining multiple criteria from a single data source, multiple criteria from multiple data sources can be combined. Recommendation methods can also be used in various application domains involving big data such as marketing, biology, chemistry. In this thesis, four applications are studied: 1) use of multiple criteria from a single source to make recommendations, 2) use of multiple criteria from multiple sources to make recommendations, 3) use of recommendation methods to predict gene regularity networks and 4) use of recommendation methods to identify new indicators for known drugs. Firstly, we propose a new multi-objective optimization based recommendation method that combines multiple criteria, namely past preferences of users, hometown of users, friendship relation among users, check-in time information. We expanded this method by inferring home/center location of users in terms of longitude-latitude pairs, by making dynamic recommendations based on temporal preferences of users and by clustering users by their hometown and friendship relations. These methods are evaluated on a Foursquare check-in dataset. Secondly, we combine information collected from multiple different social networks to create integrated models of individuals and to make recommendations to them. To our knowl-

edge, this is the first work aiming to use information from multiple social networks in recommendation process by modeling the users. For this purpose, we collect and anonymize two data-sets that contain information from BlogCatalog, Twitter, Flickr, Facebook, YouTube and LastFm web-sites. We implement several different types of recommendation methodologies to observe their performance while using single or multiple features from a single source or multiple sources. Thirdly, observing the common features of recommendation systems and gene regularity networks (GRNs), we use the proposed multi-objective optimization based recommendation method to predict the gene relationships; such that which genes regulates the others. Lastly, we adapt the proposed recommendation method to identify new indications for known drugs, i.e. drug repositioning.

Keywords: Recommendation systems, Multi-objective optimization, Multi-source data, Gene regularity networks, Drug re-positioning

ÖZ

ÇOK AMAÇLI ÖNERİ SİSTEMİ

Özsoy, Makbule Gülçin
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Faruk Polat
Ortak Tez Yöneticisi : Prof. Dr. Reda Alhajj

Haziran 2016 , 198 sayfa

Öneri sistemleri, kullanıcıların tercihlerini tahmin ederek onlara yeni ürünler önerirler. Öneri sistemlerinin çoğu hesaplamalarında tek bir ölçüt; kullanıcının ürüne verdiği puan; kullanır. Daha doğru öneriler vermek için öneri sistemleri birden fazla ölçütü bir arada kullanabilirler. Tek bir kaynaktan elde edilen birden fazla ölçütü kullanılmasının yanında, birden çok kaynaktan elde edilen birden çok ölçütün bir arada kullanılması da mümkündür. Ayrıca, öneri sistemleri metotları farklı bilim dallarının; biyoloji, kimya gibi; ilgilendiği çok fazla veri içeren uygulama alanlarında da kullanılabilir. Bu tezde, genel olarak dört alanda çalışmalar yapılmıştır: 1) Tek veri kaynağından elde edilen birden çok ölçütün öneri sistemlerinde kullanılması, 2) birden çok veri kaynağından elde edilen birden çok ölçütün öneri sistemlerinde kullanılması, 3) gen düzenleyici ağ yapılarının tahmin edilmesinde öneri sistemlerinin kullanılması ve 4) ilaç yeniden konumlandırılması amacıyla öneri sistemlerinin kullanılması. İlk olarak birden çok ölçütü; kullanıcıların önceki kayıtları (check-in), konumları, arkadaşlık ilişkileri, check-in zaman bilgisi; bir arada kullanabilen çok hedefli optimizasyon yapabilen bir öneri sistemi metodu tasarlanmıştır. Kullanıcıların konumunu enlemboylam değerlerini bularak, kullanıcıların zaman tercihlerine göre dinamik öneriler üretilerek ve kullanıcıları konum ve arkadaşlık ilişkilerine göre gruplayarak tasarlanan bu metot iyileştirilmiştir. Değerlendirme için bir Foursquare kayıt (check-in) veri-seti kullanılmıştır. İkinci olarak, birden çok sosyal ağdan toplanan birden çok ölçüt bütünleştirilerek model oluşturulmuş ve bu model kullanıcılara öneri sunmak

amacıyla kullanılmıştır. Bildiğimiz kadarıyla bu çalışma birden çok kaynaktan elde edilen verilerin kullanıcıları modelleyen ve öneri sistemlerinde kullanan ilk yöntemdir. Bu amaçla, BlogCatalog, Twitter, Flickr, Facebook, YouTube ve LastFm internet sitelerinden veriler toplanarak anonimleştirilmiş ve toplanan verilerle iki ayrı veri seti oluşturulmuştur. Bir veya daha çok kaynaktan toplanmış, bir veya daha çok ölçütü kullanan birçok farklı öneri sistemi metodu uygulanmış ve oluşturulan veri seti üzerinde değerlendirilmiştir. Üçüncü olarak öneri sistemleri ile gen düzenleyici ağ yapılarının benzerliği gözlemlenerek, bu tezde önerilmiş olan öneri sistemi yöntemi genler arası ilişkilerin, hangi genin diğer geni yönettiği gibi, tahmin edilmesi amacıyla kullanılmıştır. Son olarak bu tezde önerilmiş olan öneri sistemi yöntemi bilinen ilaçların yeni endikasyonlarının bulunması amacıyla uyarlanmıştır.

Anahtar Kelimeler: Öneri sistemleri, Çok Amaçlı Optimizasyon, Çok Kaynaklı Veri, Gen Düzenleyici Ağlar, İlaç Yeniden Konumlandırma

To my family

ACKNOWLEDGMENTS

I have to start by thanking my family and friends for their endless support and encouragement. They helped me at every step of this study in countless ways.

My advisors, Faruk Polat and Reda Alhajj have been very valuable for me. Their experience and wisdom definitely improved my skills in doing research. I have to also state how grateful I am to Reda Alhajj for his efforts in helping me live in Calgary without any problems other than being far from my home.

I am also grateful to my advisory committee members for their constructive comments in improving the research in this thesis.

The research in this thesis was supported by TUBITAK-BIDEB PhD scholarship for research abroad (2214-A).

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xvii

LIST OF FIGURES xx

LIST OF ABBREVIATIONS xxiii

CHAPTERS

1 INTRODUCTION 1

1.1 Problem Definition and Motivation 1

1.2 Organization of the Thesis 4

2 RELATED WORK 7

2.1 Location Aware Recommendation Systems 9

2.2 Social Network Aware Recommendation Systems 11

2.3 Temporal Information Aware Recommendation Systems . . . 12

2.4 Multi-Objective Recommendation Systems 17

2.5 Hybrid Recommendation Systems 18

2.6 Clustering Based Recommendation Systems 20

2.7 Inferring User Locations 21

2.8 Use of Multiple Social Networks 22

2.9 Gene Regularity Network Inference 24

2.10 Drug Repositioning 26

3 MULTI-OBJECTIVE OPTIMIZATION BASED RECOMMENDATION SYSTEM 29

3.1 Multi-Objective Optimization Based Recommendation System 29

3.1.1 Similarity calculations 30

3.1.2 Neighbor selection 30

3.1.3 Item selection 32

3.2 Location and Social Network Aware Multi Objective Recommendation System 34

3.2.1 Converting Binary Check-in Data to Rated Check-in Data 35

3.2.2 Similarity Criteria 36

3.2.3 Methods 37

3.3 Multi-Objective Optimization Based Recommendation Enhanced With Inferred Home Locations 38

3.3.1 Inferring Locations of Users and Items 39

3.3.2 Multi-Objective Optimization Based Recommendation Enhanced With Inferred Home Locations . . 40

3.3.3 Details of the Implementation 40

3.4 Time Preference Aware Dynamic Recommendation Enhanced with Location, Social Network and Temporal Information . . 41

3.4.1 Motivation 42

3.4.2 Modelling Temporal Information and Its Use in Recommendation 46

3.4.3 Details of the Implementation 47

3.5 Multi-Objective Optimization Based Recommendation Enhanced With Clustering 48

3.6 A Location and Social Network Aware Hybrid Recommendation System 50

3.6.1 System Design 50

3.6.2 Hybridization Techniques 51

3.6.3 Details of the Implementation 54

3.7 Evaluation of Multi-Objective Optimization Based Recommendation 55

3.7.1 Evaluation Metrics 55

3.7.2 Checkin2011 Dataset 57

3.7.3 Evaluation Results 57

3.7.3.1 Evaluation Results: Location and Social Network Aware Multi Objective Recommendation System 58

3.7.3.2 Evaluation Results: Multi-Objective Optimization Based Recommendation Enhanced with Inferred Home Locations 69

3.7.3.3 Evaluation Results: Multi-Objective Optimization Based Recommendation Enhanced with Temporal Information . 76

3.7.3.4 Evaluation Results: Multi-Objective Optimization Based Recommendation Enhanced with Clustering 84

3.7.3.5 Evaluation Results: Hybrid Recommendation Enhanced with Clustering . 92

3.7.3.6 Evaluation Results: Comparison of Methods 96

3.8 Conclusion 106

4 MAKING RECOMMENDATIONS USING MULTIPLE SOCIAL NETWORKS 109

4.1 Multi-source Dataset 111

4.2 Recommendations Using Multiple Data Sources 118

4.3 Evaluation of Recommendation using Multi-Source Social Networking 124

4.3.1 Multi-Source Social Networking Dataset 124

4.3.2 Evaluation Results 126

4.3.2.1 Recommendation of Flickr groups . . 127

4.3.2.2 Recommendation of BlogCatalog followees 132

4.4 Conclusion 144

5 INFERRING GENE REGULATORY NETWORKS VIA MULTIPLE DATA SOURCES AND MULTI-OBJECTIVE OPTIMIZATION . . . 145

5.1 Pareto dominance and collaborative filtering based prediction 146

5.2 Evaluation of Gene Regularity Network Structure Prediction 151

5.2.1 Bio-informatics Datasets 151

5.2.1.1 Microarray and Transcription Factor (TF) Binding Datasets 152

5.2.1.2 Dream4 In Silico Network Challenge Dataset 152

5.2.2 Evaluation Results 153

5.2.2.1 Evaluation Results for Microarray and Transcription Factor (TF) Binding Datasets 154

5.2.2.2 Evaluation Results for Dream4 In Silico Network Challenge Dataset 160

5.3 Conclusion 162

6 DRUG REPOSITIONING USING PARETO DOMINANCE AND COLLABORATIVE FILTERING 163

6.1 Pareto dominance and collaborative filtering based prediction 163

6.2 Details of the proposed method 164

6.3 Evaluation of Drug Repositioning 169

6.3.1 Drug Repositioning Datasets 169

6.3.2 Evaluation Results 170

6.4 Conclusion 177

7 CONCLUSION AND FUTURE WORK 179

7.1 Summary and Conclusion 179

7.2 Future Work 181

REFERENCES 183

CURRICULUM VITAE 197

LIST OF TABLES

TABLES

Table 3.1 Rating assignment to binary check-in data 35

Table 3.2 Rating assignment to binary check-in data 36

Table 3.3 Number of users with the given precision upper-bound value 62

Table 3.4 Results for Base setting 63

Table 3.5 Results for Weight Based setting 63

Table 3.6 Results for Rate Based setting 64

Table 3.7 Results for Rate and Weight Based setting 64

Table 3.8 Results for $r = 5km$ 73

Table 3.9 Results for $r = 100km$ 74

Table 3.10 Comparison of methods using travel locality and not 75

Table 3.11 Results for non-dynamic methods using time similarity 77

Table 3.12 Comparison of methods using temporal similarity or not 78

Table 3.13 Upper bound of the metrics based on the given temporal preference 79

Table 3.14 Results for WD_M 80

Table 3.15 Results for WD_A 81

Table 3.16 Results for WD_E 81

Table 3.17 Results for WD_N 82

Table 3.18 Results for WE_M 82

Table 3.19 Results for WE_A 83

Table 3.20 Results for WE_E 83

Table 3.21 Results for WE_N 84

Table 3.22 Comparison of dynamic results 85

Table 3.23 Results of Base setting and Clustering based on hometown 86

Table 3.24 Results of Base setting and Clustering based on friendship 87

Table 3.25 Results of Weight Based setting and Clustering based on hometown 87

Table 3.26 Results of Weight Based setting and Clustering based on friendship 87

Table 3.27 Results of Rate Based setting and Clustering based on hometown . 88

Table 3.28 Results of Rate Based setting and Clustering based on friendship . 88

Table 3.29 Results of Rate and Weight Based setting and Clustering based on hometown 88

Table 3.30 Results of Rate and Weight Based setting and Clustering based on friendship 89

Table 3.31 Comparison of the results of the clustering methods to traditional and multi-objective optimization methods 89

Table 3.32 Time requirements for similarity calculations, recommendtion computation and total in seconds 92

Table 3.33 α_R values 93

Table 3.34 Results of Mixed(M) technique 94

Table 3.35 Results of Weighted(W) and Vote(V)/Weight(W) techniques 95

Table 3.36 Results of Weighted(W) and Parameter&Vote(PV) techniques . . . 95

Table 3.37 Results of Weighted(W) and Parameter&Weight(W-PW) techniques 96

Table 3.38 Comparing results of the hybridizaton methods to traditional and multi-objective optimization methods 97

Table 3.39 Precision upper-bound value for different k values 98

Table 3.40 Precision upper-bound value when travel locality is considered for different k values 98

Table 3.41 Precision upper-bound value when time categories are considered for different k values 99

Table 3.42 Comparison of methods (Precision) 102

Table 3.43 Comparison of methods (Ndcg) 103
Table 3.44 Comparison of methods (Hitrate) 104
Table 3.45 Comparison of methods (Coverage) 105

Table 4.1 Last update dates on Facebook in the given date ranges 115
Table 4.2 The abbreviations used in this section 126
Table 4.3 Comparison of methods (CF_BCF vs. HI-FG_FCC_BCF) 139
Table 4.4 Comparison of methods (CF_BCF vs. MO-FG_FCC_BCF) 143

Table 5.1 The best results for the F1-measure and with the 3F_Exp 155
Table 5.2 The best results for the precision and with the 3F_Exp 156
Table 5.3 The best results for the recall and with the 3F_Exp 156
Table 5.4 The best results for the precision and with the 4F_Exp 157
Table 5.5 The results for the undirected graph 157
Table 5.6 The results for N=6 and k=2 with WAVG item selection method . . 159
Table 5.7 The results for N=12 and k=2 with WAVG item selection method . 160
Table 5.8 The results for ND, WAVG, N=6 and k=2 160
Table 5.9 The top 5 results for InSilico_Size10 challenge (Avg. Time series) 161
Table 5.10 The results for InSilico_Size10 challenge 161
Table 5.11 The top 5 results for InSilico_Size100 challenge (Avg. Time series) 161
Table 5.12 The results for InSilico_Size100 challenge 162

Table 6.1 The best results when different similarity metrics are used 175
Table 6.2 AUC-PR results when different similarity metrics are used 176
Table 6.3 Comparison of the proposed method to other state of the art methods from the literature 176

LIST OF FIGURES

FIGURES

Figure 1.1 General design of the methods and applications used in this thesis 2

Figure 3.1 Example input and non-dominated solutions 31

Figure 3.2 Data with multiple features 34

Figure 3.3 Example check-in and time information of users 42

Figure 3.4 Neighbor selection: Check-in similarity based 43

Figure 3.5 Neighbor selection: Time similarity based 44

Figure 3.6 Neighbor selection: Check-in&Time similarity based 44

Figure 3.7 Neighbor selection: Multi-objective optimization based 45

Figure 3.8 Proposed system design 50

Figure 3.9 N vs Precision when $k = 10$ 59

Figure 3.10 N vs Hitrate when $k = 10$ 59

Figure 3.11 N vs Precision when $k = 30$ 60

Figure 3.12 N vs Hitrate when $k = 30$ 60

Figure 3.13 k vs Precision when $N = 30$ 61

Figure 3.14 k vs Hitrate when $N = 30$ 61

Figure 3.15 Results for different settings: CF-C 65

Figure 3.16 Results for different settings: CF-F 65

Figure 3.17 Results for different settings: CF-I 66

Figure 3.18 Results for different settings: CF-H 66

Figure 3.19 Results for different settings: MO-CI 67

Figure 3.20 Results for different settings: MO-CFI 67
Figure 3.21 Results for different settings: MO-CFIH 68
Figure 3.22 Results for different settings: MO-CH 68
Figure 3.23 Average distances of users to their check-in venues (in km) 70
Figure 3.24 Number of users who visited venues in the given ranges 71
Figure 3.25 User ids and their average distances to their check-in locations (in km) 72
Figure 3.26 Number of home-towns with the given number of users (in log_2 scale) 91
Figure 3.27 Number of users with the given number of friends (in log_2 scale) . 91
Figure 3.28 Comparison of methods according to their Precision performance 100
Figure 3.29 Comparison of methods according to their Ndcg performance . . . 100
Figure 3.30 Comparison of methods according to their Hitrate performance . . 101
Figure 3.31 Comparison of methods according to their Coverage performance 101

Figure 4.1 Number of users on each social network 111
Figure 4.2 The collected data 114
Figure 4.3 The general structure of the system 119
Figure 4.4 Collaborative filtering recommendation method 119
Figure 4.5 The multi-objective optimization based recommendation method . 120
Figure 4.6 The hybrid recommendation method 121
Figure 4.7 The social-historical model based recommendation method 122
Figure 4.8 Evaluation results for N=5 and k=5 (Flickr groups) 129
Figure 4.9 Evaluation results for different values of N 130
Figure 4.10 Evaluation results for different values of k 131
Figure 4.11 Evaluation results for N=16 and k=4 (Flickr groups) 133
Figure 4.12 Evaluation results for N=16 and k=27 (Flickr groups) 134
Figure 4.13 Evaluation results for N=16 and k=12 (Flickr groups) 135

Figure 4.14 Evaluation results for N=5 and k=5 (BlogCatalog followees) . . . 137

Figure 4.15 Evaluation results for HI-FG_FCC_TF_BCF_LG method and k=5 (BlogCatalog followees) . . . 138

Figure 4.16 Evaluation results for HI-FG_FCC_TF_BCF_LG method and N=50 (BlogCatalog followees) . . . 140

Figure 4.17 Evaluation results for N=50 and k=15 (BlogCatalog followees) . 141

Figure 4.18 Evaluation results for N=50 and k=30 (BlogCatalog followees) . 142

Figure 5.1 Example input and non-dominated solutions for prediction of GRN 148

Figure 5.2 Example for regulated genes (item) selection method . . . 151

Figure 5.3 The precision results for ND and WAVG with different N and k (3F_Exp) . . . 158

Figure 5.4 The precision results for ND and WAVG with different N and k (4F_Exp) . . . 159

Figure 6.1 Design of the proposed method . . . 164

Figure 6.2 Example input and non-dominated solutions . . . 167

Figure 6.3 Upper bounds recall, precision and F1 measures . . . 171

Figure 6.4 Upper bounds of recall, precision and F1-measure when random guess is not allowed . . . 172

Figure 6.5 Performance results (Precision) . . . 173

Figure 6.6 Performance results (Recall) . . . 173

Figure 6.7 Performance results (F1-Measure) . . . 174

LIST OF ABBREVIATIONS

LBSN	Location based social network
GRN	Gene regulatory network
POI	Point of interest
API	Application programming interface

CHAPTER 1

INTRODUCTION

In this chapter, the problem definition and motivation and the organization of this thesis are presented in Sections 1.1 and 1.2, respectively.

1.1 Problem Definition and Motivation

Many people share their opinions, comments and ratings using several different web-based social network platforms, such as social networks (Twitter, Facebook etc.) and location based social networks (Foursquare, Facebook Places etc.). These platforms have many active users who provide data, e.g. check-ins, friendship, and ratings. Up to the end of March 2016, Twitter reached 310 million monthly active users and 1 billion unique visits per day [126], Foursquare reached more than 50 million users and 8 billion check-ins [29] and Facebook reached 1.09 billion daily active users on average [28]. Such data are used in many problem domains, such as event detection, traffic forecasting, urban planning, disease spread and recommendation([110, 101, 134]).

One of the problem domains that can benefit from the data provided by the (location based) social networks is recommendation systems. Recommendation systems estimate the users' future preferences based on their historical information. In general, the traditional recommendation systems do not use all the criteria provided by the social networks but are based on a single criterion, namely the overall rating. However, a user may consider more than one criteria while deciding to use an item [2]. In order to give more accurate recommendations, a recommendation system can take

advantage of considering multiple criteria. Location based social networks (LBSNs) are one of the resources that provide many features/criteria at once. In this thesis, we aim to use LBSNs as a source to reach multiple criteria and to combine these criteria with the recommendation purpose.

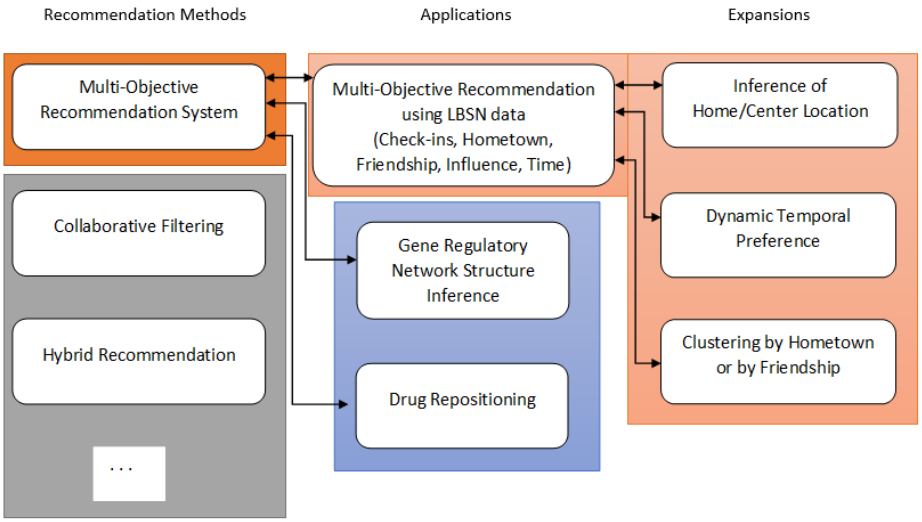


Figure 1.1: General design of the methods and applications used in this thesis

In Figure 1.1, the general design of the methods and applications used in this thesis are shown. In this thesis, we propose a multi-objective recommendation system. The proposed system is used for making recommendations using multiple criteria collected from a LBSN. The recommendation method is further expanded by inferring home/center of users, making dynamic recommendations based on users' temporal preferences and clustering users. The proposed system is also used in bioinformatics and biochemistry domains to infer structure of gene regulatory networks and to identify new indications of known drugs; drug re-positioning.

LBSNs give their users the opportunity to share their current locations via check-ins, to share their comments on venues (e.g., restaurants, cafes), to connect to their friends, and to share personal information such as gender or home-town. With the help of LBSNs, it is possible to embed location, social networks, time information and dynamic preferences of users into recommendation systems. In this thesis, we proposed a new multi-objective optimization based recommendation method, which combines all these features. We expand the proposed method by inferring home/center location of users and recommending venues in a predefined radius, by using temporal infor-

mation and making dynamic recommendations based on user temporal preference, by clustering users by their hometown or friendship. We also implemented user-based hybridization technique to compare the proposed method.

As already expressed, many different web-based platforms; e.g. social networks, review web-sites, e-commerce web-sites; use recommendation services to serve their users. For instance, IMDB is a movie review web-site that has a service named as "Recommended for you" which gives movie recommendation to its registered users. LinkedIn is a social-networking site for professionals and has a service named "Jobs You May Be Interested In" to suggest jobs to members based on their profiles. Most of these platforms use the information existing in their web-site only to model their "user"s [77]. However, they may miss some information about the "person" who is using their services. People tend to use different web-platforms for different purposes. For example, even though both LinkedIn and Facebook are social networking platforms, people use mostly LinkedIn for professional connections and Facebook for personal connections [92]. Thus, combining information from various platforms can help in modeling users better [140]. In this thesis, we combined information collected from multiple different social networks to create integrated model of users and to give recommendations. For this purpose, we collected and anonymized two specific datasets that contain information from BlogCatalog, Twitter, Flickr, Facebook, YouTube and LastFm web-sites.

Besides the web-based platforms, recommendation methods can be used by different applications related to different branches of science. Gene regulatory network (GRN) inference, from biology, is one of these applications. GRNs are biological networks in which the nodes represent genes, proteins, metabolites and other biological components and edges show the interactions among them. In the literature it is stated that GRNs have the following features: sparseness, scale-free topology, modularity and structurality of inferred networks [46]. These features are similar to features of the recommendation systems. For example, both of GRNs and recommendation systems are sparse and have a topology that usually follows power distribution function ([136, 34, 7]). Observing the similarities, we applied our proposed recommendation method to infer the structure of GRNs.

Another application area that recommendation methods can be applied is drug repositioning. Drug repositioning is the process of identifying new indications for known drugs [69]. It can be used to overcome problems associated with traditional drug discovery by adapting existing drugs to treat newly discovered diseases. Thus, it may reduce the risks, cost and required time to identify and verify new drugs. With the advancements in technology, researchers can reach different types of biological data and complex networks composed of different types of interactions among biological components [38]. Using these data sources, many different computational methodologies can be used to predict possible new use-cases for drugs. For this purpose, we adapted our proposed recommendation method to drug re-positioning problem.

1.2 Organization of the Thesis

The rest of this thesis is organized as follows:

In Chapter 2, the literature review on several different aspects of recommendation systems are presented. In this chapter, social network aware, temporal information aware, multi-objective optimization based, clustering based and hybrid recommendation systems are reviewed. Furthermore, systems that infer users' locations, integrate data from multiple social networks and infer gene regularity networks are detailed.

In Chapter 3, the proposed multi-objective optimization based method is explained. The proposed recommendation method is used to give point of interest (POI) recommendations by combining historical preferences of the users, the friendship relationship among them and explicitly shared home-town information provided by the users. Different expansions to the proposed method are also introduced. The first expansion infers the home/center location of users and uses it in recommendation process. The second expansion considers the time of check-in information and the temporal preference of user. The third expansion adds a clustering step as a pre-processing step to the proposed recommendation method. Lastly, a user-based hybridization method is implemented for comparison. Finally, the experimental evaluation of the proposed method and its expansions are presented.

In Chapter 4, we explain the process of collecting data from multiple sources and

integrating them for recommendation purpose. After creating the related data-sets, we used the proposed multi-objective optimization based recommendation method. In this chapter, the experimental set-up and the results of using integrated data collected from multiple data sources are presented.

In Chapter 5, information on how the proposed recommendation method is used for an application in biology, namely to infer the structure of gene regularity networks (GRNs). In this chapter, the experimental set-up and the results for GRN structure inference problem are presented.

In Chapter 6, we describe how we employ our recommendation method to solve the problem of identifying new indications for known drugs (drug re-positioning). The experimental evaluation results are reported.

In Chapter 7, the thesis is concluded and the future research directions are stated.

CHAPTER 2

RELATED WORK

The goal of recommendation systems is suggesting a user the items that might be of interest for him/her [89]. The most common form of giving recommendation is estimating preference for unseen items and choosing the ones with the highest estimation values [124]. In the literature there are three basic approaches to make recommendations, namely content based, collaborative filtering and hybrid approaches. While content based approach uses item similarity to make recommendations, collaborative filtering approach uses past preferences of the users to decide which item to recommend. Hybrid methods basically combines previous two approaches to make recommendations.

The data to be processed by a recommendation system has basically three elements, which are *user*, *item* and *rating*. In most of the algorithms, these elements are represented by a matrix or a graph. Traditional recommendation systems consider neither location nor social networks nor time information. Today many applications, such as LBSNs, provide these pieces of information to the researchers. The data collected from LBSNs provide not only information on locations the users visited, but also the users' preferences and habits [6]. With the help of LBSNs, e.g. Foursquare, Facebook Places, it is possible to embed location, social network and temporal information into recommendation systems. Besides, these applications provide information on users' profiles, such as gender, age and home-town. One problem of home-town information provided by the social networks is that their granularity is not well-defined. Most of the location information is given in terms of city, state or country level. In order to give better recommendations, there is a need for fine-tuned location information.

Recently, methods in the literature started to consider multiple criteria instead of a single criterion to increase the accuracy performance. Also, hybrid recommendation approaches can be used to combine results of multiple single criterion based recommendation methods and give better recommendations. Besides, there are some works in the literature that attack the scalability issue of the recommendation systems by applying clustering techniques. Another set of methods aims to use other information provided by the social networks, such as users' gender, age and home-town. Another trend that can be used for better recommendations is to model users' interests on other domains and use that in the target domain; this is known as cross-domain recommendations. Most cross-domain recommendation methods described in the literature are based on item matches. Alternatively, mapping identities across domains can be useful to figure out how users behave in different domains and use this information in making recommendation. Another idea which is recently introduced to the literature is to use recommendation methods in bioinformatics domain. It is observed that gene regularity networks (GRNs) have similar features that most recommendation systems have, such as sparsity and scalefreeness. Also, recommendation methods can be used to identify new indications for known drugs, which is known as drug re-positioning.

In Sections 2.1, 2.2 and 2.3, information on location, social network and time aware recommendation systems are explained, respectively. In Section 2.4, information on multi-objective recommendation systems is given. In Section 2.5, information on hybrid recommendation systems is given. In this section, we give general information on hybrid recommendation systems, without focusing on location, social network or time aware methods. Afterwards in Section 2.6, the recommendation systems that uses clustering techniques are explored. In Section 2.7, information on related works that aim to infer user locations from the data collected from on-line social networks and use it with recommendation purpose is given. In Section 2.8, information on the literature which focus on using information from multiple domains, e.g. social networks, is given. In Section 2.9, related work about gene regularity network inference, which is proposed to be one of the application domains of recommendation systems, is presented. Lastly, in Section 2.10, related work about drug re-positioning which is another application domain for recommendation methods, is presented.

2.1 Location Aware Recommendation Systems

Analysis show that users from a spatial region prefer different items than users in another region (preference locality) and users prefer location based items (e.g. restaurants) which are in limited/shorter distance (travel locality) [67]. These findings reveal that recommendation systems should consider spatial information such that recommendations should be given by users who are living in the same region and recommended items should be close to the target user.

LARS [67] is a framework that produce location-aware recommendations. The authors introduce three types of settings which are using user location information only, using item location information only and using both user and item locations. For the first setting, they perform user partitioning using an adaptive pyramid structure. For the second setting, they use travel penalty and avoid recommending distant items. For the last setting, the previous two approaches are combined together to make recommendations. Also in LARS* [113], the proposed methods are improved for better scalability and efficiency.

[100] propose a map-based recommendation system that collects context information, location, time, weather and user request. Using these features, they model user preferences by Bayesian network, and inferred the most appropriate item to suggest and show it on map.

[152] use GPS history of users to give location recommendations (E.g. Where should I go?) and activity recommendations (E.g. What else can I do there?). After modeling the users' past location and activity preferences and mining information on locations and activities (e.g. correlations, features), they apply matrix factorization method.

[114] proposed a method, named Extended Feature Combination (EFC), to deal with data containing user, location, activity and rating information. The author constructs 2D data structure from higher dimensional data and introduces a general data combination approach.

[136] propose a point of interest(POI) recommendation method that combines location, social network and past ratings. With the help of these information, they created

three different ranking lists and used a linear fusion framework to integrate them into one.

[16] propose a recommendation system that recommends successive personalized POI. They propose a novel matrix factorization method, FPMC-LR, which uses both location and temporal information. They showed that POI recommendation is very time-critical and its performance is related not only to past preferences but also to current location.

[144] proposes a new framework, iGSLR, which gives location recommendations by considering social and geographical information. The framework uses kernel density estimation and uses personalized geographical influence idea. The performance of iGSLR shows that integration of user preference, social influence and personalized geographical influence provides better results than the state-of-the-art methods.

[10] proposes a method which combines user, activity, time and location information to make recommendation. They use tensor tensor factorization for this purpose. They collect data from multiple sources and fuse them based on the location, i.e. which activity can be done in the target location. The experimental results of this method show that this method is able to perform better than the state-of-the art methods that use collaborative filtering or other tensor based methods.

[44] aims to make recommendation on successive check-in location. They adopt a third-rank tensor to model the users' behavior. While modeling the users, they included context information, such as the category of the location or periodicity of the check-ins. The evaluation results demonstrate improved performance compared to state-of-the-art methods on successive location recommendation.

[141] aims to recommend personalized point-of-interests (POIs) to the users by integrating various features available in the data-set. They model the users preferences by incorporating the the social, categorical, geographical, sequential, and temporal information.

2.2 Social Network Aware Recommendation Systems

Online social network sites bring a new trend to recommendation systems, which relies on the assumption that friends share similar tastes. Using this information, data sparsity problem can be handled more efficiently and quality of recommendations can be increased [78].

[34] proposes a social-historical model to explore users' check-in behavior in LBSNs. Their model integrates users' past preferences and considers the social ties. For historical tie analysis, they introduce a language model, after observing common features of language processing and location-based social networks. They utilize the sequence of check-ins and predicts the probabilities of candidate locations for the next check-in. Their model is based on the Sequence Memorizer [90]. For social ties, they investigate that friends have higher check-in similarity. To model both effects, they add user's social ties as a regularization term to historical ties.

SoCo [78] is a context-aware recommendation system that incorporates social network information. The authors apply random decision tree algorithm to partition the input matrix into subgroups of similar contexts. For predicting ratings, matrix factorization method is used in which social network information was used as a regulator. Also, in this study a context-aware Pearson correlation coefficient calculation method is proposed.

[135] aims to exploit both social and geographical information existing in LBSNs to support location recommendation services. The authors proposed a friend-based collaborative filtering(FCF) approach and its geo-measured version (GM-FCF). In FCF only friends of the target user are considered when performing collaborative filtering. The researchers also observe that nearby friends share common places more than geographically distant friends. Using this information, they propose GM-FCF, where friend similarity was also affected by geographical location.

[81] aims to provide a general method that incorporates social networks to recommendation system. The authors introduce two different regularization terms to matrix factorization. The first regularization, namely average-based regularization, uses the assumption that target user's taste is close to the average of the his/her friends' taste.

While taking the average, they weighted the taste's based on target user-friend similarity. The second regularization, namely individual-based regularization, uses friends' taste individually.

[93] proposes a new similarity metric based on user's past preferences and his/her social activities. Besides, they propose a new method. For the similarity calculations they consider similarity based on common tags on common items, friendship, and membership to the same group. The proposed algorithm applies both item similarity and user similarity.

2.3 Temporal Information Aware Recommendation Systems

Analysis on LBSN data shows that users tend to visit locations periodically [19] and that their behavior differs depending on the hour of the day (daytime vs. night) and day of the week (weekdays vs. weekend)([79, 96, 19, 21]). These results can be used for different applications, such as predicting the next check-in location or giving location recommendation. In this section, we don't only focus on recommendation systems, but also some other example works that uses temporal information obtained from check-ins.

[138] aims to recommends point-of-interests (POIs) to the target user at a specified time in a day. They perform temporal analysis by splitting the time into hourly slots and considering the check-in time slots. They use the temporal information to find out the user similarities in a collaborative filtering setting. They further enhance their approach by considering the fact that users tend to visit nearby locations. They used Bayes rule to decide on the probability that a user will visit a candidate location. At the end, they combine the two approaches using a linear interpolation method. They state that their approach performs better than all the baselines, and improves accuracy by 37% over the state-of-the-art methods.

[79] gives category aware POI recommendation. Their approach first models the preference transition of users in terms of categories, then predicts the preferred categories of the target user and finally recommends the locations in the predicted categories, also by considering geographical influence. In order to learn users' preference tran-

sition, they use the similarity of users taking into account both category information and temporal effects. To do that, they divide the time into four partitions as a combination of "working hours, leisure time" of a day and "weekdays, weekend days" of a week. Then for each temporal partition, they use the check-in frequency of the users for each location category to calculate the similarity. For all the predictions, namely prediction of next preferred categories of the target user and prediction of next check-in locations, the matrix factorization (MF) is used.

[139] gives time-aware POI recommendations by considering both geographical and temporal influences. Observing the facts that users prefer to visit nearby places, they tend to visit different places in different time slots and they periodically visit the same places, they first create a graph model for check-ins, locations and time information. Then they use this graph model to propagate the preferences with their proposed algorithm, Breadth-first Preference Propagation (BPP). Their experimental results show that the proposed method outperforms the state-of-the-art POI recommendation methods.

[75] improves the methodology introduced in [34] by including recency affect and place links. According to recency effect, users prefer to visit locations that they have visited recently rather than more distant past. According to place links, users prefer to visit locations that their explicit friends visit or people who have temporally common check-ins as they are. The experimental results show that the improvements introduced in this study provides better accuracy performance.

[13] proposes a new approach that uses Formal Concept Analysis (FCA) for context-aware recommendation. The method uses multiple contextual information, namely user, social, location and temporal information. In their work, they differentiate negative and positive ratings and split contextual information in different types. The evaluation results show that FCA is an appropriate solution for recommendation systems, differentiation of negative and positive ratings is promising and inclusion of context information is useful.

[32] explores to use temporal patterns of users to improve the location recommendation performance. Observing that users have different check-in preferences in different hours of a day (non-uniformness) and each user has similar check-ins in consec-

utive hours (consecutiveness), they propose a location recommendation method with temporal effects. They first divide the check-ins data into slots based on the hour of a day, then they use matrix factorization on each time slot. Afterwards, they combine the results in each time slot by using four different aggregation strategies, namely sum, mean, maximum and vote. Their evaluation results show that considering temporal effects for location recommendation is very effective.

[48] aims to recommend right locations at the right time. They capture the spatial and temporal aspects of user check-in information in a single probabilistic model, called Spatio-Temporal Topic (STT). They use latent topics to represent the users' interest, semantic relationship among locations, preferred regions for each user and temporal activity patterns. Their evaluation results on real life data sets from Twitter, Gowalla, and Brightkite show that STT outperforms the state-of-the-art models in recommendations systems and topic modeling.

[125] gives route suggestions by considering spatial and temporal behavior of the visitors of a theme park. After creating sequence of Location-Item-Time (LIT) information based on the visitors' behavior analysis, they mine the frequent LIT sequences by their proposed algorithm, namely the Location-Item-Time PrefixSpan (LIT-PrefixSpan). In order to recommend a route, they retrieve the appropriate LIT sequence based on the given constraints, such as intended total visit time and intended visit times for each facility. In their work they noted that this is the first work which includes location, item and time information together to represent a sequence.

[72] aims to predict the next check-in location of the target user after his/her a few (three in the application) consecutive check-ins. The proposed approach as Collaborative Exploration and Periodically Returning model (CEPR), first forecasts if a user is looking for a new place to visit -by proposed Exploration Prediction (EP)-, and then according to the results either Markov model or collaborative filtering or their combinations are used. They use the Markov model when user is not looking for a new place but follow his/her regular behavior and the collaborative filtering enhanced with social and geographical information otherwise. The results of CEPR shows that first deciding on the behavior of the user by applying EP improves the overall performance. Also for recommendation process, it improves the performance up to 30%

compared to the traditional recommendation systems.

[55] aims to predict the location of the target user by choosing the most influential friends. They first perform sequential random walk with restart procedure (SeqRWR) to rank the friends, and then use a Bayesian model which combines temporal and spatial information(TSBM) to predict users' locations by using friends' locations. They conclude that the proposed method increases performance in terms of accuracy.

[33] studies on temporal effects on LBSNs and models user mobile behavior. They propose a framework that analyses and models the temporal cyclic patterns of human behavior (temporal preferences) and their relationship with spatial and social data (temporal correlations). They consider three layers of LBSNs, namely social, geographical and temporal layers. For these layers, they consider the social friendship, historical check-ins and temporal stamps of the check-ins, respectively. They use Gaussian distribution to model the temporal patterns. They observe that users' behavior is affected by multiple cyclic patterns, and they are correlated with the users' friends. They conclude that temporal and spatial information complements each other and improves the location prediction performance.

[96] predicts the next location that the user will visit given the historical data and the current location of the target user. They apply two machine learning methods, linear regression and decision trees, using the features of individual users (historical visits, social ties), global knowledge (popularity of places, their distances and user transitions between them) and temporal information of users' movements. In terms of temporal information, they consider the "category hour", "category day" and "place day", and the frequency of check-ins in terms of hour of day (in the range of 0-23) and hour of week (in the range of 0-167), as well as the day of the week for specific venues. In terms of temporal features, they find that predicting the next check-in location of a user in weekdays at working hours performs better than predicting at night or at the weekends. They explain this by the observation that during night and the weekends people deviate more from their regular mobility patterns.

[19] studies the social and temporal characteristics of users by analyzing their check-ins. They conclude three base results: The users visit places periodically; their mobility is affected by geographic constraints, economic constraints and social status and

content and sentiment analysis can provide rich contextual information. As a result of their analysis on when users check in, they find that there are three main check-in peaks at around 9 am, 12 pm and 6 pm. In terms of weekly pattern, they observe that the behavior of users differs in weekdays and weekends. In the weekdays, there are two peaks; lunch time and dinner time; while at the weekends the peaks are blended.

[21] aims to understand the laws of human motion and dynamics by analyzing relationship among the human geographic movement, daily routine on human mobility patterns and the effect of social ties. Their analysis on one cell phone location data and two LBSN data shows that people generally move periodically in a bounded region, but sometimes visit distant places. The distant places they go are usually in proximity of an existing friend. In terms of temporal information, they find that most of the users visit the same locations that they once checked in before. Another finding they observe is that during the daytime the variability of check-ins is less, while during rush hours and at the weekends it is higher. They conclude that during these time, the movement of human is less predictable.

[142] proposes a method that uses Temporal Influence Correlations (TIC) to make time-aware recommendations and to recommend time-to-visit that location. It combines user-based and item-based (location-based) correlations.

[153] learns context similarities to make context-aware recommendation, where one of the contexts can be time information. The idea of the proposed method is that the similarity among contextual situations should produce similar recommendation lists. In this work, various similarity calculation methods are used, such as Independent Context Similarity (ICS), Latent Context Similarity (LCS), Weighted Jaccard Context Similarity (WJCS) and Multidimensional Context Similarity (MCS).

[146] aims to recommend the successive locations to the user based on the user's current location. For this purpose, they proposed a new model called location and time aware social collaborative retrieval model (LTSCR). In order to make recommendation, it combines the current location of the target user, the friendship relations among users and the time information.

[150] makes successive point of interest(POI) recommendation using a ranking-based

pairwise tensor factorization method. They incorporate user's previous check-ins, user's last check-in location and time information, and model user-user, location-location and location-time information. They also employ a newly proposed interval-aware weight utility function to differentiate check-in correlations based on time interval.

2.4 Multi-Objective Recommendation Systems

Most of the recommendation systems are based on single criterion, such that they aim to evaluate the item based on overall rating. A user may consider more than one criteria while deciding to use an item. For example, while choosing a movie to watch, the user may not only consider the overall rating, but also the genre, the actors/actresses, the director and etc. In order to give more accurate recommendations, a recommendation system can take advantage of considering multiple criteria. In [2], [3], [1] and [84], the importance of multi-criteria recommendation systems is highlighted. The common techniques in the literature are as follows: Taking a linear combination of multiple criteria, finding Pareto optimal solutions, optimizing for only the most important criterion and consecutive optimization for one criterion at a time.

Taking a linear combination of multiple criteria technique combines the results which are obtained in prediction step for each criterion. In [63], marginal utility value for each criterion is used while taking sum of the prediction scores. Similarly, in [85] the sum of prediction scores is calculated. Also, in the same study another approach which considers weights assigned by users to indicate importance of the criteria are used while getting the sum. After combining the multiple criteria ratings, the items that maximize the value are recommended to the user. [54] introduces a recommendation framework that is based on constrained linear optimization techniques. In the method each item is assigned a utility score which is depended on the item's rating and the predefined operational objectives. In the experiments, they use promotion of long tail items and resource constraints as operational objectives. [109] expands [54] by explicitly controlling any potential loss while adding new utility aspects. They experimented their approach on a talent-match setting where potential employees are suggested to the talent-seekers. The evaluation results show that significant improve-

ment in user engagement is achieved while keeping the degradation in relevance in acceptable levels.

Finding Pareto optimal solutions technique discovers several good items among large number of candidates [2]. This approach does not require priory-assigned weights to the criteria. In [65] this approach is used for a restaurant recommendation system where users indicate their preference. For example, in this system a user may look for cheap and Italian food, and the system tries to find the optimal restaurant based on these criteria. In [2], it is stated that these systems may suffer from scalability problem when number of criteria gets larger.

Optimizing for only the most important criterion technique filters recommendations using a single criterion, which is indicated by the user. In the fourth technique, consecutive optimization for one criterion at a time, the ranked list of criteria is used for filtering the recommendations.

A recent study focuses on finding the most representative neighbors with the help of multi-optimization based algorithms. [97] explains that the traditional collaborative filtering methods are insufficient at finding the representative users as neighbors. They propose to use Pareto dominance to eliminate less representative users and to select the most promising ones. This work considers only one criterion, namely the rating. After deciding on the most promising candidates, the neighbors are decided by calculating the similarities between the target user and the candidate users and the candidates with the highest similarity are assigned as neighbors. For the item selection, the past preferences of the neighbors are used.

2.5 Hybrid Recommendation Systems

In recommendation systems, considering more than one criterion can be handled by hybrid recommendation systems. In [11], it is stated that all recommendation methods have strengths and weaknesses, and different methodologies to combine them are used to increase the recommendation performance.

In [11], different recommendation methods and hybridization techniques are ana-

lyzed. The analyzed recommendation methods are collaborative filtering, content-based filtering, demographic recommendation systems, utility based recommendation systems and knowledge-based recommendation systems. The analyzed hybridization techniques are weighted, switching, mixed, feature combination, cascade, feature augmentation and meta-level techniques. According to the analysis results, the writer states that some of the hybridization methods are not studied as much as other and needs further exploration. Based on this observation, the writer proposes a hybrid system that uses collaborative filtering and knowledge-based recommendation system.

[3] states that many hybrid recommendation systems combine collaborative and content filtering methods to avoid the limitations of them. Some of the techniques that are used to combine them are listed as follows: Combining separate recommendation systems, adding content-based characteristics to collaborative models, adding collaborative characteristics to content-based models, and developing a single unifying recommendation model. Besides giving information on collaborative filtering, content filtering and hybrid recommendation systems, the researchers also give ideas on possible extensions to improve the recommendation performance. Some possible extensions are improvements in understanding users and items, using contextual information, using multi-criteria ratings and producing more flexible recommendation systems.

In [56], a hybrid recommendation system for e-commerce is proposed. The system dynamically assigns a personalized weight for each method and combines them. The methods used in the research are the content based filtering, collaborative filtering and demographic filtering.

In [39], performance of hybrid recommendation systems is analyzed. They combine four different Web usage mining methods using four different combination techniques. One of the combination methods is proposed newly in this study. They conclude choice of the hybridization technique and the single recommendation methods to be used affect the overall performance result. In general, they state that the hybrid system performs better than a single method.

[123] aims to predict click-through rate (CTR) of web objects by exploring ensemble

of recommendation strategies. They model recommendation as a contextual bandit problem. They state that the objective of recommendation systems and bandit algorithms are maximizing the total user response and total reward, respectively, and are equivalent. In their study, they propose two combination techniques and show that their method is robust in terms of CTR.

2.6 Clustering Based Recommendation Systems

The recommendation systems need to perform calculations on thousands of users and/or items in less than a second. This means that these systems have to deal with scalability issues. In the literature clustering based methods are proposed to make recommendation systems work faster.

[111] partitions the users into subgroups and uses the members in a subgroup as neighbors. In this work, they aim to give faster recommendation without losing the quality of recommendations. They observe from the evaluations that clustering decreases the quality slightly but increases the throughput; the number of recommendations generated per second; sharply. This means that clustering is promising for solving the scalability problem of the recommendation systems.

[62] extends their previous work UTA-Rec [63] by adding a clustering step. UTA-Rec [63] uses Multi-Criteria Decision Analysis(MCDA) techniques in recommendation. Instead of using a single criterion, such as overall rating, the writers consider multiple criteria; namely story, acting, direction and visuals. In [62], they use the output of their MCDA technique to calculate similarities among users. Then they perform clustering on users based on their similarities.

[4] clusters the users and then uses the members in a cluster in the process of recommendation. They adapted a data structure from information retrieval, namely cluster-skipping inverted index structure. Using inverted index, the researchers can use as many users and/or items as possible without any memory problem, while having the same representation as the traditional data structures; e.g. matrix representation. The evaluation results of [4] show that their method gives high accuracy and reasonable scalability results.

2.7 Inferring User Locations

Users' home location can be used by recommendation systems to increase the performance. Many social networks provide this information by asking their users to fill a free format area. The users mostly fill this area with information in city, state or country level if not with some artificial/fake information ([35, 45]). In order to obtain fine-tuned location information, it should be inferred from the other fields of the available data. In the literature, researchers focused on inferring home location from different web-sites, social networks, LBSNs. Besides, there are some other works that focus on privacy issues revealed by home location inference.

[82], [45], [18] and [24] aim to infer the location of Twitter users. [82] use machine learning techniques to infer user home locations in different levels, namely city, state or time zone. The researchers use textual context of tweets, users' tweeting behavior and external dictionaries containing names of cities/states as the features of their system. [45] use machine learning methods to infer the users' home state or country using only the textual content of the tweets. [18] infers the users' location in city-level and performs analysis only on the content of the tweets. For the analysis, they use the common vocabulary of users with the intuition of users from the same region use specific place names and similar idioms or expressions. [24] infer users' home locations using their following-follower information. They propose if a user follows another user and is followed by that user, then these users are friends. They explain that people become friends who live nearby and obtained friendship information can be used to infer the users' home location.

Some other works aim to infer home location of users from other web-sites/social networks, such as Wikipedia, Flickr and Facebook. [74] aim to infer locations of Wikipedia users based on their edit histories. They find out that many of the users in Wikipedia tend to contribute about small geographic regions, which are correspond to the place where users' currently live or where they were born. [104] predicts the home country and gender of Flickr users. They use photo annotation tags, titles, date taken, upload date, and geo-tags. [5] aims to predict the location of Facebook users. The researchers find out that friendship of users drops as the distance between them increases. Using this analysis result, they used social structure existing in the

Facebook to predict the location of users.

In terms of privacy, some of the researchers study on inferring user locations even if the users don't reveal this information publicly. [43], [151] and [76] don't specifically focus on inferring the users' home location information, but on predicting users' private attributes in general. [43] use friendship links and apply a Bayesian network approach to model the relationships among users. They show that private attributes can be inferred especially when users are strongly connected. [151] aim to infer users' private attributes using friendship and group membership information. They find that especially group membership information leaks private attributes.[76] apply a modified Naive Bayes algorithm to predict privacy sensitive attributes. They use both users' attributes and the social network's link structure. Besides, they propose a technique to decide on the most effective attributes or links to be removed to protect privacy. Some other works, whose main concern is privacy, focus on inferring users' home location. [103] analyze the publicly available data in Foursquare to infer users' home location in terms of city, state or country level. They find that with 78% accuracy users' home location can be found within 50 kilometers. They extended their work in [102]. This work focus on inferring users' home location on popular social networks, namely Foursquare, Google+ and Twitter. They use the publicly available attributes together with friendship information. They explain that home location of users in city level can be found with accuracy around 67%, 72% and 82% for Foursquare, Google+ and Twitter. They also aim to find the geographic coordinates of the residences and achieve accuracy around 60% within a radius of six kilometers. They conclude that sharing location information(check-ins) on social networks can reveal the users' residence location, which may mean a privacy leak.

2.8 Use of Multiple Social Networks

Recommendation systems aim to make recommendations to users based on their interests. Recently, most of the research on recommendation system focus on combining different kinds of information, as exemplified in the previous sections. These systems mostly use either linear combination of features or multi-objective optimization methods. Even though these works use multiple features at once, none of them

use data from multiple data sources. Recently, in a challenge [15] related to recommendation systems, using diverse data from multiple sources is used as the main purpose of the challenge. Methods which ranked higher on different tasks of the challenge used hybridization and ensemble methods [25]. Even though the idea of the challenge is similar to ours, unlike our work it is based on using diverse data from multiple sources about items, not about users.

Another set of research focuses on cross-domain recommendation, which models users in a domain and employs the model in a target domain. Works described in [131], [121], [148], [60], [50] and [80] are some examples from cross-domain recommendation systems. These systems mostly use item-based matches and do not consider users' identities or they use data from a single source and assume different categories, such as books and movies, as different domains. One of the first research efforts on cross-domain recommendation belong to [131]. In that work, users were surveyed on category names and ratings they give. The collected data was analyzed both in group and at individual levels. Results showed that multiple information sources for recommendations is promising. The work described in [121] found correlation between objects by using a Bayesian hierarchical approach based on Latent Dirichlet Allocation (LDA) method by modeling users' interests and objects' topics. Output correlations were used to make recommendations to target users based on their interests. Zhang et al. [148] aimed to make recommendations across websites by using browsing information of users. This idea is similar to ours in the sense that we aim to use multiple social networks and they used multiple browsing history. However, browsing history of users may not be always available. Kumar el al. [60] used textual information of items to map them across domains. Then these mappings were used to give cross-domain recommendations. Hu et al. [50] modeled users, items and domains together with the assumption that users behave similarly across domains. They evaluated their method on books and movies data-sets collected from Amazon web-site. The work described in [80] modeled users' preferences separately on each domain using types of items. Then using factorization machines, they combined separate models into one. Li el al. [68] identified user and item mapping across rating matrices and used the out mapping in the recommendation process. In their work, they assumed similar rating behavior of users on both domains, and there were

some overlapping users/items. They evaluated their method on a synthetic data-set and Yahoo! Music data-set.

An alternative to cross-domain recommendation can be using identity resolution across domains, such as mapping users on different domains. This approach can be useful to analyze users' behavior on different domains and analysis results can be used by other applications, e.g., recommendation systems. Works described in [77], [92],[140],[53] and [122] are some example works that aim to connect identities across social networks, namely identity resolution. They mostly focus on mapping users across domains, but not on their preferences or interactions with the related social network, i.e., they do not make any recommendation. Liu el al. [77] used two different social networking platforms to collect user descriptions. Then using co-occurrence of words, the authors built a network which connects interests and identities. They used this network to make recommendations. They did not aim to figure out individual identities but generic groups, such as Dog Lovers. Authors of [92] searched and matched users across online social networking platforms. For matching purposes, they used several different attributes of users; such as age, gender, location, country and name. Zafarani et al. [140] mapped individuals across social media sites by first identifying users' unique behavior patterns, such as using similar names or typing patterns, then constructing features based on the captured behavior, and lastly identifying users using machine learning techniques. Jain et al. [53] used content and network features additional to previously used features to map users across Facebook and Twitter. They concluded that using different attributes provides distinct aspects of the identity of users, and helps to improve performance of the identity resolution process. Finally, Tan et al. [122] proposed a semi-supervised manifold alignment method to map users across social networking platforms. Even though they used social structures only, they stated that names of users can also be used to boost performance of the system.

2.9 Gene Regularity Network Inference

The properties of GRNs are sparseness, scale-free topology, modularity and structurality of inferred networks [46]. In the GRNs the number of connections among genes are limited, such that the GRNs are sparse. GRNs follow the power distribu-

tion for the connectivity ([94, 147]), such that some genes regulates many other genes while some others regulate only few or no other genes. This property is related to the scale-free topology feature of the GRNs. They are structurally decomposable into network motifs [46] and they can be clustered into groups such that the genes in a group are highly co-expressed or have similar functions [108].

The methods to infer (reverse engineer) GRNs commonly use Boolean networks, Bayesian networks, relevance networks, differential and difference equations [108]. Recently, integration of prior knowledge to the GRN inference gained attention the literature ([108, 91, 120, 46]).

In Boolean networks, based on the gene expression levels and the input parameter of threshold, the gene interactions are represented as a boolean function. The aim of the reverse engineering is to find out the related boolean function for each gene [108]. A REVerse Engineering ALgorithm (REVEAL) [73], [61] and [117] are some example approaches that are based on Boolean networks.

The Bayesian networks are the most commonly used model to infer GRNs [108]. These networks are based on the conditional dependence of the nodes (e.g. genes), where the conditional probabilities are based on the parent nodes only. Using this feature the probability of the graph can be calculated by a joint probability distribution, which is dependent on the probability of existence of edges between nodes. [30], [149] [41], [120] and [116] are example approaches that are based on Bayesian networks.

The use of differential and difference equations can be appropriate to use to infer GRNs, since the concentration of biological components changes over time [129]. The equations are based on the input gene expression data, the time, the model parameters and external effects. It aims to find the changes in the gene expression data and the relations among the genes. [129], [36] and [70] are example approaches that are based on differential and difference equations.

In the relevance networks, using the similarity metrics such as Pearson coefficient or mutual information the connection among genes are decided. The constructed graph is undirected, but by introducing a threshold as a parameter, it is possible to decide

on the direction of the edges. ARACNE [88] and [115] are example approaches that are based on relevance networks.

Recently, integration of prior knowledge and multiple types of data to the GRN inference process attracts attention in the literature. [149], [70], [120], [51] [147] are example approaches that combines multiple data sources to infer the GRNs more accurately.

2.10 Drug Repositioning

Identifying new indications for known drugs, namely drug repositioning, has recently received more attention from industry and academia. The work described in [27] classifies computational drug repositioning methods into two categories: namely drug-based and disease based approaches. Drug-based repositioning methods initiate their analysis from the chemical or pharmaceutical features of drugs. Disease-based repositioning methods initiate the analysis from symptomatology or pathology features of diseases. Drug repositioning methods use various features for the computations [145], e.g., Chemical structure of drugs, protein targets interaction networks, side-effect of drugs, gene expressions and textual features.

There are many drug repositioning methods described in the literature, However, they mostly use only one feature: The structure and chemical properties of a drug is directly related to which diseases it affects. Drugs with high chemical similarity can be used for drug repositioning [27]. The works described in [95] and [57] are example methods that use chemical similarity for drug repositioning. Authors of the work described in [22] states that common segments in protein-protein interaction and protein targets interaction networks can reveal cross-reactions and can be used for drug-repositioning. The works described in [71] and [58] use protein targets interactions networks. Side effects are physiological consequence of drugs' biological activity; they can provide information on underlying pathways or physiological systems to which drugs are related [27]. Side-effect similarity between drugs may indicate physiological relatedness between them. The works described in [12] and [133] use side-effect similarity of drugs for drug repositioning. Similarities in molec-

ular level can also be used for drug repositioning [27]. For this purpose, the works described in [118], [52] and [49] use gene expressions and molecular activity signatures. Some of the works described in the literature rely on text mining tools to connect drugs and diseases [106]. One such method is described in [17]. It applies text mining methods to associate query and matching terms about diseases, genes, drugs, mutations and metabolites. It also ranks related sentences and abstracts.

Recent drug-repositioning methods combine multiple features to achieve better performance. For instance, the work described in [69] combines chemical and molecular features to find out similar drugs. The authors applied a bipartite graph based method to predict novel indications of drugs. Gottlieb et al. [40] used chemical structures, side effects and drug targets to calculate pairwise similarity of drugs. They used the calculated similarities as input features to a machine learning method, namely logistic regression. They predicted new drug-disease relations. Zhang et al. [145] used chemical, biological and phenotypic features to calculate drug-drug similarities which are used to find out k-nearest-neighbors. Then known indications of neighbors are used for drug re-positioning. Qabaja et al. [106] combined information collected from gene expression profiling and text mining. They apply logistic regression to predict associations among drugs and diseases. Ozgur et al. [98] uses text mining techniques to create a parse tree which is used to create a protein-protein network. They also applied some social network analysis techniques (e.g., degree centrality, closeness) to prioritize genes' effect on diseases.

In this thesis, we realize drug repositioning as a recommendation process. In other words, we argue that it is possible to recommend existing drugs for treating emerging diseases based on characteristics of new diseases as compared to characteristics of existing diseases in relationship with associated effective drugs. Thus, we apply a method from recommendation systems to tackle the drug repositioning problem. The employed method is able to integrate multiple data-sources and multiple features. Similar to the work of Zhang et al. [145], the proposed method first identifies drugs most similar to the target drug and uses known relations of neighbor drugs to predict new indications of the target drug. Unlike the work of Zhang et al. [145], we use a Pareto dominance and collaborative filtering based method, which has been already used as part of adapting recommendation systems to other domains, like venue recom-

mendation and in bioinformatics to predict the structure of gene regulatory networks. Also, we applied several settings for the calculations and compared their performance to each other.

CHAPTER 3

MULTI-OBJECTIVE OPTIMIZATION BASED RECOMMENDATION SYSTEM

Traditional recommendation systems do not consider location or social network information and they are usually based on a single-criterion, namely overall rating. In this thesis, we proposed a method that considers multiple criteria at once. We also expanded the base proposed method by inferring location of users, including time information and temporal preference of users, clustering and hybridization techniques. The base methodology and the expansions are explained in Sections 3.1-3.6. The evaluation process and performance results of the methods are presented in Section 3.7. The chapter is concluded in Section 3.8.

3.1 Multi-Objective Optimization Based Recommendation System

Most of the recommendation related researches on the data containing multiple criteria aggregate the preferences into a utility function, usually by getting the weighted sum. Unlike previous works, we combine all the criteria into a vector representation and decide results using multi-objective optimization methods. The proposed method is composed of three steps:

- Similarity calculations: Using each criterion; such as item ratings, user locations, friendship; similarity among users is calculated.
- Neighbor selection: The most similar users (neighbors) to the target user is selected by multi-objective optimization methods.

- Selection of items: Using traditional approach used in user-based collaborative filtering, such that past preferences of the neighbors, the items are selected. Then, the items scores are calculated and the top ranked items are recommended to the target user.

The details of the steps are given in the following sections, namely Sections 3.1.1,3.1.2 and 3.1.3:

3.1.1 Similarity calculations

The first step of the method is to calculate user-user similarities based on several different contexts, such as past preferences, location, friendship, home-town, gender and age. The calculation of similarities can be based on any similarity measure used in the literature, such as Cosine similarity, Pearson correlation, Jaccard similarity.

The equations for the common similarity metrics to calculate user similarities, namely Pearson correlation, Cosine similarity, Adjusted Cosine similarity and Jaccard similarity, are given in the Equations 3.1, 3.2, 3.3 and 3.4, respectively.

In the equations the following notation is used. Users are shown with *u* and *v* and the items are shown with *i*. *rat(u,i)* indicates the rating of the item *i* given by the user *u*. $rat_{avg}(u, .)$ indicates the average rating given by the user *u*. $rat_{avg}(u, *)$ and $rat_{avg}(v, *)$ are used to denote the average rating given by user *u* to the common items rated by the user *v*, or vice versa. *sim(u,v)* indicates the similarity of user *u* to user *v*.

3.1.2 Neighbor selection

Having the similarities for each user to the others, the next step is to decide the most similar users to the target user. For this purpose, non-dominated users are found out by finding the Pareto optimal points. We propose that non-dominated users are the ones that will affect the target most. Non-dominated users are founded by deciding on which user dominates which other and selecting the users who are never been dominated.

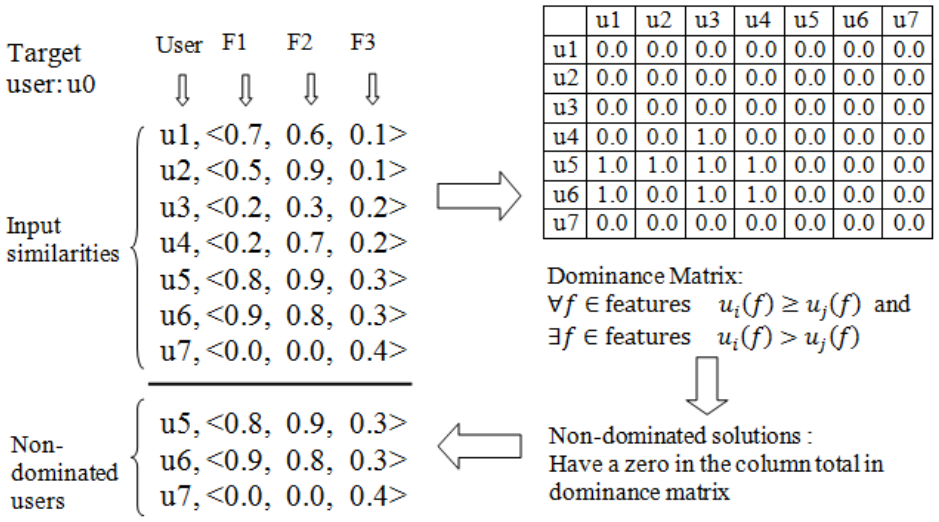
$$sim(u,v) = \frac{\sum_i (rat(u,i) - rat_{avg}(u,*))(rat(v,i) - rat_{avg}(v,*))}{\sqrt{\sum_i (rat(u,i) - rat_{avg}(u,*))^2}\sqrt{\sum_i (rat(v,i) - rat_{avg}(v,*))^2}} \tag{3.1}$$

$$sim(u,v) = \frac{\sum_i rat(u,i)rat(v,i)}{\sqrt{\sum_i rat(u,i)^2}\sqrt{\sum_i rat(v,i)^2}} \tag{3.2}$$

$$sim(u,v) = \frac{\sum_i (rat(u,i) - rat_{avg}(u,.))(rat(v,i) - rat_{avg}(v,.))}{\sqrt{\sum_i (rat(u,i) - rat_{avg}(u,.))^2}\sqrt{\sum_i (rat(v,i) - rat_{avg}(v,.))^2}} \tag{3.3}$$

$$sim(u,v) = \frac{|rat(u,.) \cap rat(v,.)|}{|rat(u,.) \cup rat(v,.)|} \tag{3.4}$$

In Figure 3.1 an example of multi-dimensional data is given. In this example, the similarity values of seven users to the target user, $u0$, are given for three different criteria, namely F_1-F_3. To make the example more concrete, one can assume that these similarities are check-in, home-town and friendship similarity, which are calculated in the previous step.



	u1	u2	u3	u4	u5	u6	u7
u1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
u2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
u3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
u4	0.0	0.0	1.0	0.0	0.0	0.0	0.0
u5	1.0	1.0	1.0	1.0	0.0	0.0	0.0
u6	1.0	0.0	1.0	1.0	0.0	0.0	0.0
u7	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 3.1: Example input and non-dominated solutions

In order to find the non-dominated users, the first step is to create the dominance matrix. In a dominance matrix, the cell values indicate if the user in the row dominates the user in the column. The equation to fill the cells is given in the Equation 3.5. In the equation, f indicates the features used and u and v indicates the users. According

to the equation, a user dominates another, name them as u and v respectively, if the u's scores (similarities in our example) are greater than or equal to the v's, and there exists at least one feature score of u that is greater than the v's. In Figure 3.1, based on the example similarities, the output dominance matrix is given.

$$dom(u,v) = \begin{cases} 1.0 & \forall fu(f) \geq v(f) \text{and} \exists fu(f) > v(f) \\ 0.0 & \text{otherwise} \end{cases} \tag{3.5}$$

The second step of finding the non-dominated users is selecting the non-dominated users. Having the dominance matrix, non-dominated users are decided by looking at the column sums of the dominance matrix. The users whose column sum equals to 0.0 are the non-dominated users. In the example, $u5$, $u6$ and $u7$ are selected as the non-dominated users.

Neighbor selection step can be terminated in one iteration or in multiple iterations. If it is terminated in a single step, the neighbors' count can be limited, i.e. less than the input parameter value. In order to collect as many neighbors as given, an iterative process of neighbor collection can be applied. First, we apply the method of finding non-dominated users explained previously. If the number of non-dominated users is less than the given neighbor count, we remove the selected users from the data representation and re-apply the method of the finding non-dominated users. We continue this process until the predefined number of neighbor count is reached.

3.1.3 Item selection

The last step is to make recommendation of items. The items preferred by the neighbors are considered as candidate recommendations. The more neighbors recommend an item, the more the score of the candidate item is. At the end, *top-k* items with the highest score are suggested to the user.

The score calculation is performed according to the Equation 3.6. In the equation, *s(u,i)* is the predicted score of item *i* that the target user *u* will give. *v* is a user who is chosen as a neighbor to the user *u*. *sim(u,v)* is the similarity of users *u* and *v* and *rat(v,i)* is the rating given to the item *i* by the user *v*.

$$s(u,i) = \sum_{v \in neighbors} sim(u,v) rat(v,i) \tag{3.6}$$

For the similarity and rating scores used the Equation 3.6, different values can be used, i.e. they can be considered binary or multiple-valued. We considered four different settings:

- Basic: Input rating information is considered to be binary. For example, a user likes a place or not. In this setting, the selected neighbors are considered to have same level of effect on the target user. The similarity value of each neighbor is assigned to 1.0. The rating score is also assigned to 1.0.

- Weight Based: In this setting, the neighbors are given different weights, so that they have different level of effect on the target user. For example, if first neighbor has weight value 0.4 and the second neighbor has weight value of 0.8; then the recommendations from the second neighbor will be considered to be more (twice) important than the first one. We assigned the weight of the neighbors according to their similarities to the target user. The weight of a neighbor is assigned by taking the average of similarities that are used in the related method (The Equation 3.7). In the equation, $w(u,v)$ is the calculated weight, the sim_k is one of the similarities that is calculated in the first step, and the $|Similarities|$ is the total number of similarities calculated on the first step. In this setting, we considered the input rating information as binary.

$$w(u,v) = \frac{\sum_{sim_k \in Similarities} sim_k(u,v)}{|Similarities|} \tag{3.7}$$

- Rate Based: In this setting, the rating scores are used in the range that is either given in the input data or that is calculated in a pre-processing step. The similarity values of neighbors are assigned to 1.0 as in the Base setting.

- Rate and Weight Based: This setting is a combination of rate based and weight based methods. In this setting the input ratings are given in a range and the neighbors are assigned different weights.

3.2 Location and Social Network Aware Multi Objective Recommendation System

In this section, we used the proposed method in Section 3.1 to make recommendations by considering not only users, check-ins (as items) and ratings, but also home-town (as location), friendship and social network information (Figure 3.2).

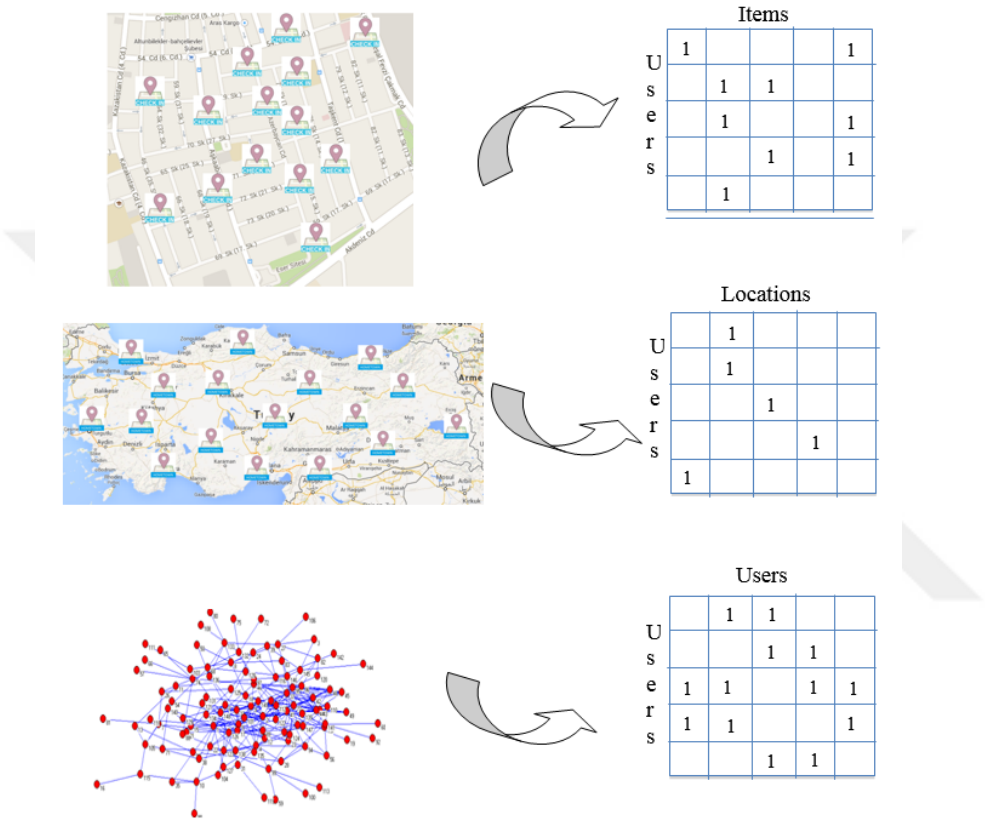


Figure 3.2: Data with multiple features

For the implementation, the steps given in Section 3.1 are followed. However, some of the calculations, such as conversion of binary data to rated data and similarity calculations, are dependent on the data set characteristics, such that they are calculated based on the available features/criteria. Obviously, the features to be used during the recommendation process is also dependent on the data, as it provides the related feature or not.

In this thesis, if not explicitly stated otherwise, we use the Checkin2011 data-set [31] during implementation and test. The data contains information of users' check-in,

Table 3.1: Rating assignment to binary check-in data

Range	#Check-ins	Rating score
$x \leq 1$	64510	1.0
$1 < x \leq 5$	17753	2.0
$5 < x \leq 11$	2688	3.0
$11 < x \leq 23$	1052	4.0
$23 < x$	372	5.0

friendship and home-town. The check-in information is binary by definition, such that a user checks in a location or not. In order to apply rate based calculations, we introduced a step that assigns ratings to the system. We detail this in Section 3.2.1. In this section, we implemented a recommendation method that uses the historical preferences of the users, the relationship among users and their home-towns. The similarity measures among users are calculated using these kinds of information (Detailed in Section 3.2.2). We implemented not only our proposed method with different combinations of the criteria but also the traditional collaborative filtering methods. The details on the implemented methods are given in Section 3.2.3.

3.2.1 Converting Binary Check-in Data to Rated Check-in Data

Analysis of the input data showed us that there are many locations that a user checks in only once, but there are some others that the users checks in several more times. If a user checks in at the same location very frequently, this shows that the user prefers that location more than a location where he/she checks in less. So, we assigned rating scores to the check-ins depending on the number of check-ins for a location.

We start by assigning 1.0 as the rating score to the check-ins where the user visits only once. Then we take the average of the check-in counts discarding the one timers, and assigned 2.0 as the rating score to the range in between the one to the average. For example, in our data-set the average of check-ins of users is 5 when single check-ins are excluded; such that if users check in at a location more than once, they check in at the same location 5 times, on the average. We continue in the same manner until we reach 5.0 as the rating score. Table 3.1 shows the related ranges of check-in counts, total number of check-ins in those ranges and the assigned rating scores.

Table 3.2: Rating assignment to binary check-in data

Range	#Check-ins	Rating score
$x \leq 1$	64510	1.0
$1 < x \leq 5$	17753	2.0
$5 < x$	4112	3.0

Looking at the table we observe that the number of check-ins which have 3.0 or more as ratings are less than the number of the ones with ratings 1.0 and 2.0. So we decide to give rating 3.0 to all of the ratings which are assigned to 3.0 or more previously. We end up with a rating scale in between 1.0-3.0, and with the thresholds 1 and 5. At the end the ratings are assigned as presented in Table 3.2.

3.2.2 Similarity Criteria

We used four different similarity criteria which are based on check-in, home-town, friendship and social influence features.

The first criterion is user-user similarity based on user check-in matrix. The assumption is that similar users prefer to check in at similar places. For the user check-in based similarity calculations, we used Cosine similarity metric.

The second criterion is user-user similarity based on home-town. The assumption is that users from the same home-town prefer similar locations to check in. This similarity is set to 1.0 if the users are from the same home-town, and set to 0.0 otherwise.

The third criterion is user-user similarity based on friendship. The assumption is that friends prefer check in at similar locations. This similarity is set to 1.0 if the users are friends, and set to 0.0 otherwise.

The last criterion is related to the influence of the users on another. Given a social network, global influence of users can be calculated by several different algorithms. The most intuitive one is to use the count of connected friends, such that if many users are connected to a single user (count of inward edges is high), then that user is considered to have more influence on the social network. Another method to calculate influence is to use PageRank [99]. PageRank is proposed to decide the importance

of the web-pages and rank them. Similarly, finding out influencers is the process of finding the important users/nodes in the social network and ranking them. However, PageRank method needs global information of the input network, which leads to scalability problems in a large data-set. Instead, there is a need for influence analysis on local information.

In this thesis, we used a local influence model that uses friendship information only. We modeled influence of a user on the target user by finding out the common friends. The idea is that if a user has many common friends with the target, this user will be able to influence the target more. For the influence calculations Cosine similarity metric on friendship data is used.

3.2.3 Methods

We implemented several different versions of the proposed multi-objective optimization based method and traditional collaborative filtering based methods:

- Collaborative Filtering on User x Check-in (CF-C): The traditional user based collaborative filtering algorithm is used to make recommendations. The most similar users based on past check-in information are defined as the neighbors.
- Collaborative Filtering on Friends(CF-F): The friends are defined as the neighbors.
- Collaborative Filtering on Influence(CF-I): The users who have most influence on the target user are defined as the neighbors.
- Collaborative Filtering on Home-town(CF-H): The users from the same home-town as the target user are defined as the neighbors.
- Multi-Objective User Similarity: Multiple context and/or similarity measures are used to decide the non-dominated users. These users are considered as neighbors. We used different combination of contexts to observe the effects of them:
  - Check-in & Influence (MO-CI): The criteria are check-in similarity and influence.

- Check-in & Friends & Influence (MO-CFI): The criteria are check-in similarity, friendship and influence.

- Check-in & Friends & Influence & Home-town (MO-CFIH): The criteria are check-in similarity, friendship, influence and home-town.

- Check-in & Home-town (MO-CH): The criteria are check-in similarity and home-town.

3.3 Multi-Objective Optimization Based Recommendation Enhanced With Inferred Home Locations

LBSNs give their users the opportunity to give information of their current locations via check-ins, to share their comments on venues (e.g restaurants, cafes), to connect to their friends and to share personal information such as gender or hometown. However, in terms of users' hometown information these social networks are not enough in terms of their granularity. Most of them lets users to fill a free format area related to their hometown information. The users mostly fill this area with information in city, state or country level if not with some artificial/fake information ([35, 45]). In order to obtain fine-tuned location information, there is a need for inferring users' home-town/central point.

In this section, we extend Section 3.1 by considering the travel locality preference of the users. [67] explains the travel locality as users' preference of location based items/venues (e.g. restaurants) which are in limited/shorter distance. We calculated the users' fine-tuned home-town and venue's locations, as longitude-latitude pairs. Then, distance based on the calculated longitude-latitude values are used as a measure for travel locality to give better recommendations.

In Section 3.3.1, the approach that we used for inferring home-town of the users and exact location of venues in terms of longitude-latitude pairs is explained. In Section 3.3.2, the extended recommendation system is explained. In Section 3.3.3, the implementation details based on the characteristics of the input data is given.

3.3.1 Inferring Locations of Users and Items

Our intuition is that if a user prefers to be at similar locations in terms of longitude-latitude pairs up until now, then this user will tend to visit similar or closer locations in the future. [67] supports this idea by emphasizing "travel locality" idea in their work. Travel locality can be defined as users' preference of location based items (e.g. restaurants) which are in limited/shorter distance to them.

In this thesis, we refer the commonly visited locations or regions as the user's home location or center point. We inferred the home/center locations of users by averaging their previous check-ins. [19] and [35] attract attention to the fact that there can be outliers in the check-ins, such as check-ins during a trip in a new city, and that averaging check-ins can be negatively affected by them. we are aware that outliers in the check-ins may affect the home location inference, and we want to apply different approaches beside averaging, such as introducing personal distance thresholds, as a future work.

The process of averaging contains three basic steps. In the first step, for each user the average of longitude values of all check-ins is calculated. In the second step the same calculations performed for latitude values. In the last step, each user's home/center location is assigned as the calculated average values of longitude-latitude pair.

We observed from the data that users who check in at the same venue (based on the location ids) provide different longitude-latitude pair values, but these values are in a limited range. This most probably occurs because of the area of the venue and the different seating of the users. Our intuition is that average of check-ins will provide us the center location of the related venue. Similar to the process of inferring home location of users, we calculated the exact location of venues (e.g restaurants, cafés) by averaging the check-ins. The average longitude value of check-ins in each venue is assigned as the longitude of the center location of that venue. Similarly, the average latitude value of check-ins in each venue is assigned as the latitude of the center location of that venue.

3.3.2 Multi-Objective Optimization Based Recommendation Enhanced With Inferred Home Locations

The extended system uses the same steps as the previous section, Section 3.1, such that it is composed of the following steps:

- Similarity calculations: Using each criterion; such as item ratings, user locations, friendship; similarity among users is calculated.
- Neighbor selection: The most similar users (neighbors) to the target user is selected by multi-objective optimization methods.
- Selection of items: Using traditional approach used in user-based collaborative filtering, such that past preferences of the neighbors, the items are selected. Then, the items (e.g restaurants, cafés) which are in the range of interest area are recommended to the target user. The interest area of a user is decided by creating a circle around the user's inferred home/center location. The radius of the circle is decided based on the input data, which is detailed in the evaluation section.

Note that unlike the previous method, in this method we also consider the travel locality by considering only the items (e.g. venues) which are in a limited radius.

3.3.3 Details of the Implementation

In general, the same approach followed in the previous section is followed. Here, we use the same data as in the previous one, namely Checkin2011 data-set [31]. In the implementation step, several different methods using inferred home/center location of users/items and the travel locality idea are implemented. The methods use either traditional collaborative filtering idea or the proposed multi-objective optimization based recommendation method. The implemented methods are as follows:

- Inferred Location + Collaborative Filtering on User x Check-in (IL-CF-C): The traditional user based collaborative filtering algorithm is used to make recom-

mendations. The most similar users based on past check-in information are defined as the neighbors.

- Inferred Location + Collaborative Filtering on Friends(IL-CF-F): The friends are defined as the neighbors.
- Inferred Location + Collaborative Filtering on Influence(IL-CF-I): The users who have most influence on the target user are defined as the neighbors.
- Inferred Location + Multi-Objective User Similarity: Multiple context and/or similarity measures are used to decide the non-dominated users. These users are considered as neighbors. We used different combination of criteria to observe the effects of them:
  - Check-in & Influence (IL-MO-CI): The criteria are check-in similarity and influence.
  - Check-in & Friends & Influence (IL-MO-CFI): The criteria are check-in similarity, friendship and influence.

3.4 Time Preference Aware Dynamic Recommendation Enhanced with Location, Social Network and Temporal Information

In this section, we extended the method explained in Section 3.1 in two folds: First, we added temporal information to the method to get into account the time based preferences of different users. Second, we take the target users' temporal preference into account to give dynamic recommendations. Giving personal recommendations to each user using the temporal preference is a recent idea and only few works in the literature used it.

In Section 3.4.1, our motivation on why we propose to use time information in a multi-objective optimization setting and why we aim to give dynamic recommendations are explained. Then, in Section 3.4.2, information on how we model the temporal information and the extended recommendation method are explained. Lastly, in Section 3.4.3, the implementation details of the the extended recommendation method based on the characteristics of the input data is given.

3.4.1 Motivation

Motivated by the previous analysis on human behavior in the literature and our intuition, we aim to add temporal information to the recommendation process. As mentioned in related work section, it has been shown that humans tend to behave differently depending on the time of the day (daytime vs. night) or day of the week (weekdays vs. weekends)([79, 96, 19, 21]).

Similar to the analysis results, our intuition is that we should differentiate users who check in at similar locations at similar times than users who check in at similar locations at different times, as well as users who check in at different locations. We give Figure 3.3 as an example. In the figure, there are four users, namely $u1$, $u2$, $u3$ and $u4$, and two criteria, $c1$ and $c2$, which are the location and time of the check-ins, respectively. The locations are represented by their ideas, e.g. $L1$, and temporal information is given in terms time of the day, namely daytime(D) and night(N). In the example, the target user, to whom we want to make recommendations, is $u1$. Just by looking at the input data, we observe that $u3$ and $u4$ have visited the similar places as the target user, $u1$. However, we observe that $u3$ and $u1$ have been at the same place at the same time more often than $u4$ and $u1$. From these observations we can conclude that $u3$ is the most similar user (the neighbor) for the target user $u1$.

u1					
Check-in Loc.	L1	L2	L 3	L 4	L 5
Check-in Time	D	N	N	N	D

u2				
Check-in Loc.	L 3	L 4	L 5	L 6
Check-in Time	D	D	D	D

u3						
Check-in Loc.	L 1	L 2	L 3	L 4	L 5	L 6
Check-in Time	N	N	D	N	D	D

u4						
Check-in Loc.	L1	L2	L3	L4	L5	L6
Check-in Time	N	D	D	D	D	D

Figure 3.3: Example check-in and time information of users

Different approaches using temporal information behave differently while choosing

the neighbors and giving recommendations. The following figures show how the similarities are calculated and used to decide on the best neighbor for the target user, $u1$, based on our example given in Figure 3.3.

The first method is a collaborative filtering based recommendation method which only takes into account the past preferences (the check-in venues). In Figure 3.4, the check-in based matrix is created and the similarities among users are calculated. Based on the similarities, the users who are most similar to the target user are $u3$ or $u4$, and one of them is selected as the best neighbor. When we consider the space used for the calculations, we observe that we have to create a matrix of size $|U|x|C_1|$, where $|U|$ is the number of users and $|C_1|$ is the size of the first criterion. In this example, $|U|$ is 4 and $|C_1|$ is 6.

Check-in Similarities Based

Users vs. Check-in Loc.	**L1**	**L2**	**L3**	**L4**	**L5**	**L6**
u1	1	1	1	1	1	0
u2	0	0	1	1	1	1
u3	1	1	1	1	1	1
u4	1	1	1	1	1	1

Size: |U| x $|C_1|$

Similarities to
the target user (u1):
u1-u2: 0.670
u1-u3: 0.912
u1-u4: 0.912

u3 = u4 > u2

Figure 3.4: Neighbor selection: Check-in similarity based

The second method is a collaborative filtering based recommendation method which only takes into account the temporal information of the check-ins. In Figure 3.5, the time based matrix is created and the similarities among users are calculated. Based on the similarity values, $u3$ is selected as the neighbor as he/she has the highest similarity to the target user. The used space is calculated by $|U|x|C_2|$, where $|C_2|$ is the size of the second criterion and is equal to 2 in this example.

The third method is a collaborative filtering based recommendation method which takes into account both the location and the temporal information of the check-ins. In Figure 3.6, both check-in and time are considered together as a combination and the related matrix is created. Based on the matrix, the similarities among users are calculated and based on the results $u3$ is selected as the neighbor. The used space is

Temporal Similarities Based

Users vs. Time	D	N
u1	2	3
u2	4	0
u3	3	3
u4	5	1

Size: |U| x $|C_2|$

Similarities to
the target user (u1):
u1-u2: 0.554
u1-u3: 0.980
u1-u4: 0.707

u3 > u4 > u2

Figure 3.5: Neighbor selection: Time similarity based

calculated by $|U|x(|C_1|x|C_2|)$. Note that, we are able to combine both of the criteria by getting all the combinations. This approach is not applicable in real world data, since the number of users in the system and the size of each criteria is large.

Check-in & Temporal (Combination) Similarities Based

Users vs. Check-in Loc. & Time	L1 D	L1 N	L2 D	L2 N	L3 D	L3 N	L4 D	L4 N	L5 D	L5 N	L6 D	L6 N
u1	1	0	0	1	0	1	0	1	1	0	0	0
u2	0	0	0	0	1	0	1	0	1	0	1	0
u3	0	1	0	1	1	0	0	1	1	0	1	0
u4	0	1	1	0	1	0	1	0	1	0	1	0

Size: |U| x ($|C_1|$. $|C_2|$)

Similarities to
the target user (u1):
u1-u2: 0.223
u1-u3: 0.547
u1-u4: 0.182

u3 > u2 > u4

Figure 3.6: Neighbor selection: Check-in&Time similarity based

The fourth method uses the proposed multi-objective optimization based recommendation method and takes into account both the location and the temporal information of the check-ins. In Figure 3.7, the similarities among users based on location and temporal information are given on the left and the related dominance matrix is given on the right. The similarities are calculated as shown in Figures 3.4 and 3.5. The details on how dominance matrix is created is not given in the figure, but it is explained in Section 3.1. The non-dominated user is selected as the neighbor based on the dominance matrix, such that the user who is never dominated (whose column sum is zero) is selected. The used space is calculated by $|U|x(|C\_1| + |C\_2|) + |U - 1|x|C|$. The first part of the equation is related to the similarity calculations, which are defined in the previous paragraphs. The second part of the equation is related to the matrix which holds the similarities and the dominance matrix. The $|U - 1|$ is the number of

users except the target user and the $|C|$ value indicates the count of different criteria used, which is 2 for this example.

Multi-Objective Optimization Based

Users vs. Criteria	Check-in Loc Similarity	Temporal Similarity
u2	0.670	0.554
u3	0.912	0.980
u4	0.912	0.707

Size: (|U|-1) x |C| + |U| x ($|C_1|$ + $|C_2|$)

Dominance Matrix	u2	u3	u4
u2	0	0	0
u3	1	0	1
u4	1	0	0
Total:	2	**0**	1

Figure 3.7: Neighbor selection: Multi-objective optimization based

Previous works in the literature, our intuition and the example analyzed in this section show that using temporal information provides information on the users' behavior and can increase the performance of the recommendation system, by assisting to choose neighbors more effectively. While using the temporal information, other important features, such as social network and geographical information which are already shown to be effective in recommendation performance, should not be discarded. In order to combine all these kinds of information effectively, we believe that the multi-objective optimization based approach is a good choice.

Beside considering temporal information as a source for similarity calculations, we aim to use this information as a tool for giving dynamic recommendations. A user may ask the recommendation system to make recommendation for a specific hour of a day or day of a week. For example, to have a breakfast, a user may prefer to visit a venue which serves brunch on a weekend morning, but prefers to visit a coffee shop on a weekday morning. Also a user may require a recommendation independent from the current time (the time of asking for the recommendation). For example, on a weekday evening a user may look for brunch locations for the weekend. Traditional recommendation systems generally produce the same recommendations for a target user even if the user indicates his/her preference of time. Using the users' temporal preferences in recommendation process is started to be researched and our work is one of the first of these systems and it gives dynamic recommendations based on temporal preferences of the users.

3.4.2 Modelling Temporal Information and Its Use in Recommendation

Some of the recent works, [79], [138], [139], [32], incorporate temporal information in recommendation by dividing the time into partitions. While [79] divides the time into four by considering working hours and leisure time of a day and weekdays and weekends of a week, [138], [139] and [32] split time into hourly slots.

In our work we divided time into eight different slots, such as the combination of four partitions of the day (i.e. morning, afternoon, evening, night) and two partitions of the week (i.e. weekdays and weekend). We assigned Saturday and Sunday as the weekends and the rest as the weekdays. We assigned the hours in between 06.00 - 11.59 (6.00 am - 11.59 am) as morning, 12.00 - 17.59 (12.00 pm - 5.59 pm) as afternoon, 18.00 - 23.59 (6.00 pm - 11.59 pm) as evening and 00.00 - 05.59 (00.00 am - 5.59 am) as night.

Using these time slots we can differentiate users who socialize in different times of a day and a week; such as in the morning (i.e. morning person) or at the night (i.e. night owl). For this purpose, the temporal preference based similarity among users is used. Also in the proposed system, the time slots can be used by the target users to indicate their temporal preferences to get recommendations. As a result, our proposed method will make recommendations of venues that can be visited specifically in the given time slot.

The proposed system is composed of the same steps as explained in Section 3.1: similarity calculations, neighbor selection and item selection. The same approach for all steps are used as before, except two extensions proposed in this section:

First extension is to use temporal information of the check-ins. In the similarity calculations step, past check-in times of the users can be taken into account. After mapping the check-in times of the users to the time slots, the similarities of users based on their frequency of check-ins in the related time slot is calculated. The similarities are used in a multi-objective optimization setting to decide the non-dominated users, who are assigned as the neighbors of the target user.

The second extension is related to the use of dynamic temporal preferences of the

users in the item selection step. With this extension, it is possible for the target users to indicate their time category (slot) preferences to get recommendations, such that they can be made recommendations on venues that can be visited specifically in the requested time slot. For this extension, item selection step is modified as follows: The candidate items (venues) are eliminated according to their status on being visited on the given time slot, such that if the candidate item has never been visited in the given time slot by any of the neighbors, then it is removed from the candidate list. Then, the original item selection step is performed on the rest of the candidate items and the *top-k* items with the highest score are suggested to the user.

3.4.3 Details of the Implementation

In this section, we used the same data as in Section 3.1, namely Checkin2011 data-set [31]. The data contains information of users' check-in, friendship and home-town. Additional to the criteria used in the previous sections, we took the check-in time-stamp information into account.

The user-user similarity based on the temporal information is calculated as follows: As explained in the previous sections, we mapped the times of previous check-ins into eight different categories, which are the combination of four partitions of the day (i.e. morning, afternoon, evening, night) and two partitions of the week (i.e. weekdays and weekend). Then, frequency of check-ins in the temporal categories are used to calculate the similarities among users. For the calculations Cosine similarity metric is used.

Several different methods which are using temporal information together with different combination of features are implemented. We implemented not only non-dynamic versions of the methods but also the dynamic versions which are based on time category preference of the target users. The methods with dynamic recommendation process is initialized with Dynamic Temporal Preference(DTP) while presenting the evaluation results. The explanation of the implemented methods are as follows:

- Collaborative Filtering on Time(CF-T): The most similar users based on past check-in time information are defined as the neighbors.

- Multi-Objective User Similarity: Multiple criteria are used to decide the non-dominated users. These users are considered as neighbors. We used different combination of criteria to observe the effects of them:
  - Check-in & Time (MO-CT): The criteria are check-in similarity and time similarity.
  - Check-in & Home-town & Time (MO-CHT): The criteria are check-in similarity, home-town and time similarity.
  - Check-in & Home-town & Friends & Time (MO-CHFT): The criteria are check-in similarity, home-town, friendship and time similarity.
  - Check-in & Home-town & Friends & Influence & Time (MO-CHFIT): The criteria are check-in similarity, home-town, friendship, influence and time similarity.

3.5 Multi-Objective Optimization Based Recommendation Enhanced With Clustering

In this section, we extended the proposed recommendation method explained in Section 3.1 by introducing a pre-processing step. In this step, clustering of users based on a single criterion is used. By this process, we aimed to observe the effect of clustering on the accuracy and scalability of the system.

In the clustering step, the user set is divided into clusters based on a single criterion, such as home-town or friendship. Clustering helps the system to reduce scaling problem by considering only the users that are from the same cluster as the target user instead of considering all of the users in the system. The rest of the method is remained same, except only the cluster that the target user belongs to is used for the calculations.

Clustering can be performed on any kind of feature, however based on the available data we preferred to use either home-town or friendship information. Even though it is possible to use complex clustering methodologies, we preferred to use a simple method which directly groups users based on the feature value. By this way we believe that we can capture the changes of the users' state, such as moving to a new

city, easily without needing complex calculations. The clustering methods we used are as follows:

- Clustering based on Home-town (CLH): Users from the same home-town are grouped into the same cluster.
- Clustering based on Friendship (CLF): Users who are labeled as friends of the target user are grouped into the same cluster.

Afterwards the following methods are used to make recommendations. Note that for the calculations only the information of the users from the same cluster as the target user are used:

- Collaborative Filtering on User x Check-in (CF-C): The traditional user based collaborative filtering algorithm is used to make recommendations. The most similar users based on past check-in information are defined as the neighbors.
- Collaborative Filtering on Friends(CF-F): The friends are defined as the neighbors.
- Collaborative Filtering on Influence(CF-I): The users who have most influence on the target user are defined as the neighbors.
- Collaborative Filtering on Home-town(CF-H): The users from the same home-town as the target user are defined as the neighbors.
- Multi-Objective User Similarity: Multiple context and/or similarity measures are used to decide the non-dominated users. These users are considered as neighbors. We used different combination of criteria to observe the effects of them:
  - Check-in & Influence (MO-CI): The criteria are check-in similarity and influence.
  - Check-in & Friends & Influence (MO-CH): The criteria are check-in similarity and home-town.
  - Check-in & Friends & Influence (MO-CFI): The criteria are check-in similarity, friendship and influence.

– Check-in & Influence & Home-town (MO-CIH): The criteria are check-in similarity, influence and home-town.

3.6 A Location and Social Network Aware Hybrid Recommendation System

There are many ways to integrate multiple features into a recommendation method, such as hybrid systems or multi-criteria based optimization systems, which are exemplified in related work section. In this section, we used techniques from [11] to combine single criterion based recommendation methods, such as only location based, only friendship based or only rating based recommendation systems.

In Section 3.6.1, the system design is explained. In Section 3.6.2, the hybridization techniques from [11] are explained and the information on how we applied the chosen techniques are given. In Section 3.6.3 details of the implementation based on the dataset characteristics is given.

3.6.1 System Design

In our hybrid recommendation system, we combine multiple single criterion based methods. Unlike the other hybrid recommendation systems in the literature, we combine the selected neighbors instead of recommended items. We believe that if the best neighbors can be selected, the output recommendation would be better. After having combined list of neighbors, we decide on the items to recommend as it is done in traditional collaborative filtering method.

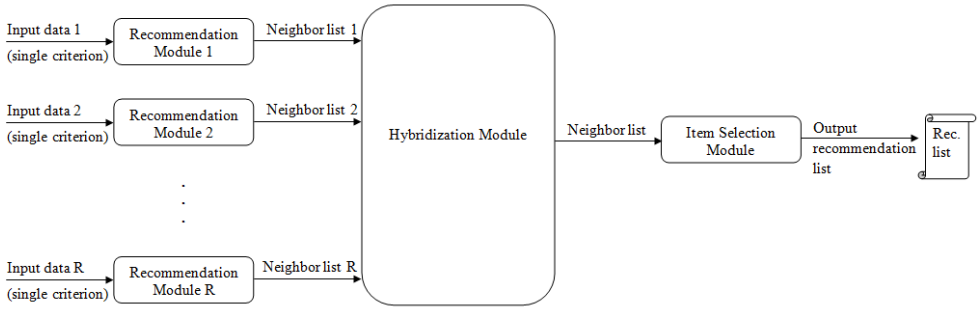


Figure 3.8: Proposed system design

The proposed system is composed of the following modules (Figure 3.8):

- Recommendation modules: Each recommendation module uses a single criterion; e.g. rating, friendship; and produces a candidate neighbors list by calculating the similarities among users. For this purpose, each recommendation module uses the following two steps:
  - Similarity calculations: Using each criterion; such as item ratings, user locations, friendship; similarity among users is calculated. The calculations can be measured by any similarity measure known in the literature, such as Cosine similarity, Pearson correlation, Jaccard similarity.
  - Neighbor selection: Based on the output similarities, the most similar users to the target user are selected as the candidate neighbors.
- Hybridization module: The chosen candidate neighbors are combined by using the selected hybridization techniques. Detailed information on hybridization techniques and how we applied them are given in Section 3.6.2.
- Item selection module: Using the combined neighbors, the items to be recommended are selected following the approach in the traditional user-based collaborative filtering method. The items previously preferred by the neighbors are considered as the candidate items. The more neighbors recommend an item, the more the score of the candidate item is. At the end, *top-k* items with the highest scores are recommended to the target user. For the score calculations, the selected neighbors are considered to have same level of effect on the target user. So, overall score of an item is calculated by counting the number of neighbors who have suggested (e.g. used previously) that item.

3.6.2 Hybridization Techniques

In hybrid recommendation systems two or more recommendation methods are combined to reduce the drawbacks of each method and to give better recommendations [11]. In that work seven different hybridization techniques are explained:

- Weighted: In this technique, the scores of each recommendation method is considered to create a single score for each item. For example, linear fusion of scores technique combines the scores of each recommendation method using a linear formula, such as summation of scores. Another example is using recommendations of each method as votes and giving higher score to the items with higher number of votes.

- Switching: In this technique, the system switches between recommendation methods based on some criteria. The switching criteria need to be decided beforehand and it may introduce additional complexity to the system.

- Mixed: In this technique, the recommendations from multiple recommendation methods are presented side-by-side.

- Feature combination: In this technique, one of the recommendation method is used as an additional feature by the other one. In [11], this technique is explained as a combination method for content filtering and collaborative filtering, where collaborative information is used as an input for the content based filtering method.

- Cascade: In this technique, recommendation methods are employed in stages. First method creates candidate items, then the second one chooses the recommendations from the candidate set. The system using the cascade technique is more efficient (than a weighted hybrid), since the second method does not deal with less important items which are not chosen by the first recommendation method.

- Feature augmentation: In this technique, the result of the first recommendation method is used as a feature by the second method.

- Meta-level: In this technique, the model produced by the first recommendation method is used as an input by the second method. The difference from the feature augmentation is that in this technique the meta-level information, such as the learned model, is used by the second method.

Even though in [11] different types of recommendation methods; such as collaborative filtering and content based filtering; are considered to be combined all the time,

we believe that multiple single type recommendation methods which selects items according to different criteria can be combined together using the explained techniques. In this section, we used several different collaborative filtering recommendation methods each of which uses a single criterion, e.g. rating, friendship. We used each method to produce the candidate neighbors, then the chosen candidate neighbors are combined by using the selected hybridization techniques. We chose to use the following techniques:

Mixed: Predefined number of neighbors are collected by getting each neighbor from different method according to the ranks. For example, let's assume that the first recommendation method, $R1$, finds the neighbors as $u1, u2, u3$, and the second one, $R2$, find it as $u4, u5, u6$. The output list will first collect the first user from the first list, which is $u1$, and then the first user from the second list, which is $u4$. The process will continue until the predefined number of users are collected.

Weighted: For this technique, we used different three different settings:

- Vote based: Each recommendation method gives a vote for each user, without considering any importance weight of methods or similarity of users. The score given for each candidate neighbor is calculated by counting the number of methods which listed it in the candidate neighbors list.

- Weight based: The similarity values calculated by each recommendation method is considered in this setting. The score of the candidate neighbor is assigned as the average similarity value.

- Parameter based: The result of each method is combined using a linear fusion method, as in [136] given in the Equation 3.8. In the equation, R represents the set of recommendation methods and r represents each recommendation method. The α_r is the parameter which is the importance weight of the related method. The total value of α_rs is 1.0, Equation 3.9. The $sim(u, v)$ value is the calculated similarity between users u and v, where u is the target user and v is one of the elements of the candidate neighbors. In this technique user similarities are either considered or not. So, we can further subdivide this technique as Parameter&Vote based and Parameter&Weight based.

$$s(u,v) = \sum_{r \in \mathrm{R}} \alpha_r * sim(u,v) \tag{3.8}$$

$$\sum_{r \in \mathrm{R}} \alpha_r = 1.0 \tag{3.9}$$

We assigned the α_r values using the following approaches:

- Equal values: Each α_r is set as $1/|R|$, where $|R|$ represents the number of recommendation methods used.
- Rank based values: The ranks of each criterion is given to the system as an input. Based on the rankings the α_r values are assigned in decreasing order. For example, assume that the ranking of three criteria is given as $c1 > c2 > c3$. Then for these criteria we assign the scores as $c1 = 3$, $c2 = 2$, $c3 = 1$ and then we divide the scores to the total score, which is 6 in this example, to get the α_r values. As a result, in this example the assigned α_r values are calculated as $c1 = 0.50$, $c2 = 0.33$, $c3 = 0.17$.
- Search based values: We search the best α_r values by assigning different values to them, such that by increasing the values with 0.1 increments in between [0.0-1.0]. Since the search becomes very complex as the number of criteria increase, we decided to search the best α_r values only for two criteria all the time, and we incrementally assigned the α_r values for all the criteria. For example, assume that we have three criteria, $< c1, c2, c3 >$. We first calculated the best α_r value for the $c1$, the first ranked criterion, by assigning different α_r values for $< c1, c2 >$ and $< c1, c3 >$. Based on the results, we decide the α_1 value. Then, knowing the first α_r value, we search the space only for α_r values of $c2$ and $c3$ by using all the three criteria in recommendation process.

3.6.3 Details of the Implementation

Generally, the steps given in the system design section are followed. However, similarity calculation step is dependent on the data-set characteristics, such that the similarities are calculated based on the available features/criteria. In this section, we used

the same data as in Section 3.1 [31] and the same similarity measures among users are calculated as done in Section 3.1. The implemented methods are as follows:

Hybrid Techniques: Multiple recommendation methods are combined using the Mixed (M) or Weighted (W) techniques. The settings used in Weighted technique are coded as: Vote (V), Weight (W), Parameter&Vote (PV) and Parameter&Weighted (PW). The different approaches used in Parameter based settings are coded as: Equal (E), Rank (R), Search (S). For example, a hybrid technique using Weighted and Parameter&Vote based using equal assignments of alpha values is coded as: W-PV-E. The produced code is added to the recommendation method's code (see below), when we give the evaluation results. We used different combination of criteria to observe the effects of them:

- Check-in & Home-town (HT-CH): The criteria are check-in similarity and home-town.
- Check-in & Influence (HT-CI): The criteria are check-in similarity and influence.
- Check-in & Friends & Influence (HT-CFI): The criteria are check-in similarity, friendship and influence.
- Check-in & Friends & Influence & Home-town (HT-CFIH): The criteria are check-in similarity, friendship, influence and home-town.

3.7 Evaluation of Multi-Objective Optimization Based Recommendation

In this section we presented the evaluation process and results of recommendation by using multiple criteria from the same source. The evaluation metrics, the data-set and the evaluation results are presented in Sections 3.7.1, 3.7.2 and 3.7.3.

3.7.1 Evaluation Metrics

We used the following evaluation metrics:

- **The Precision@k** metric is presented in Equation 3.10. In the equation tp_k represent true positives and fp_k represents false positives in the given output list with size k. True positives are the ones which are listed in the output list and are actually true, such that the recommended venue visited by the target user in the future or the predicted gene is really connected to the target gene. False positives are the ones listed in the output list but are not actually true, such as the recommended venue is not visited in the future by the target user or the predicted gene is not actually connected to the target gene. While giving the evaluation results, we presented the average of the $Prec_k$ values. Here it is worth mentioning that it is common for recommendation methods to have low precision results as the data is very sparse. For instance, in [143] the authors gave several examples of low precision results, which are in the range [0.030, 0.035], for different data-sets.

$$Prec_k = \frac{tp_k}{tp_k + fp_k} \tag{3.10}$$

- **The Ndcg**(Normalized discounted cumulative gain) metrics decides the relevance of the listed items depending on their rank. It is calculated by Equation 3.11. The Dcg (Discounted cumulative gain) value is calculated by Equation 3.12. In the equation k is the size of the returned list and j is the item's position in the list. The Idcg (Ideal discounted cumulative gain) is the Dcg value in the ideal case, where the resulting list is sorted by the relevance. While giving the evaluation results, we presented the average of the $Ndcg_k$ values.

$$Ndcg_k = \frac{Dcg_k}{Idcg_k} \tag{3.11}$$

$$Dcg_k = rel_1 + \sum_{j=2}^{k} \frac{rel_j}{log_2 j} \tag{3.12}$$

- **The Hitrate** metric shows the ratio of the users who are given at least one true recommendation. The average Precison@k value of a method can be high even though it is able to make recommendations just to a few users. For example, assume that we have two different recommendation methods, RM_1 and RM_2, two users u and v, and the output list size is 3. Consider the case where RM_1 gives

2 true recommendations to user u and no true recommendation to user v, and RM_2 gives one true recommendation to each user. Both methods' Precision@k will be 0.33, on average. However, RM_2 can give true recommendations to both users; this means we can say that it is better than RM_1. It is calculated by Equation 3.13. In the equation, M is the set of target users, and m is one of those users. $HitRate_m$ is set to 1.0 if the output list contains at least one true recommendation and to 0.0 otherwise.

$$HitRate = \frac{\sum_{m \in M} HitRate_m}{|M|} \tag{3.13}$$

- **The User Coverage** is defined as the ratio of the users who are given any recommendation by the system. As stated in [8], some of the algorithms in the literature loose coverage in order to gain more accuracy. These algorithms are usually suffering from giving poor recommendations to the cold start users. In [47] it is stated that coverage and accuracy should be analyzed together.

3.7.2 Checkin2011 Dataset

We use the Checkin2011 data-set [31] which contains 11326 users, 187218 locations, 1385223 check-ins, 47164 friendship links. The data is collected from Foursquare web-site in between January 2011 - December 2011. Since the size of the data is large, a sub-sample of the data is created by extracting check-ins made in January 2011, which is named as CheckinsJan. In CheckinsJan data, there are 8308 users, 49521 locations and 86375 check-ins. Using the CheckinsJan data, a list of venues is recommended to the users. For the evaluation, the check-ins made in February is used as the test set.

3.7.3 Evaluation Results

In this section, we presented the evaluation results of multi-objective optimization based recommendation using the following approaches: Using location and social network information; Enhanced with inferred home locations; Enhanced with temporal information, enhanced with clustering and Hybrid recommendation using location

and social network information. We used the Precision@k, Ndcg, Hitrate and Coverage metrics to analyze the performance of the methods. The results of each approach are presented in Sections 3.7.3.1 - 3.7.3.5. Lastly, the comparison of all the methods to the baseline method belonging to the creator of the data-set we use, namely Gao et al.[34], is presented in Section 3.7.3.6.

3.7.3.1 Evaluation Results: Location and Social Network Aware Multi Objective Recommendation System

In this section, first, we present the configurations that are necessary for the evaluation and then we present the evaluation results and discussions.

Configurations We aimed to predict future check-ins of each user using the Check-insJan data. In the prediction step, it is necessary to limit the number of neighbors, N, and the output list size, k. In order to decide the N and k values, we performed the following analysis.

In order to decide N, we fixed k value and performed the recommendation process using the Base setting. For the analysis, we set k value in the range of 10-30 with 5 increments and we set N value in between 10-80 with 10 increments. Then, we analyzed the results in terms of precision and hit-rate. Figures 3.9 and 3.10 shows the results when k is set to 10 and Figures 3.11 and 3.12 shows the results when it is set to 30. In all of the figures, we observed that increasing N value increases the precision and hit-rate performance. However, the acceleration of the increase reduces after a certain N value. This value can be assigned as N in the rest of the analysis. Using the outputs of our tests, we assigned N as 30, for this evaluation configuration.

Deciding the N value, the next step is to decide the k value. For this purpose, we used N as 30 and gave k different values in the range 10-30 with 5 increments, as we did previously. The precision and hit-rate results for this setting is given in Figures 3.13 and 3.14. From the results, we observed that the precision is affected by the k significantly and reduces sharply as it increases. The hit-rate increases, as expected, when the k value increases. However, the ratio of the change is not as sharp as in the precision. Considering all these, we decided to assign k value to 10.

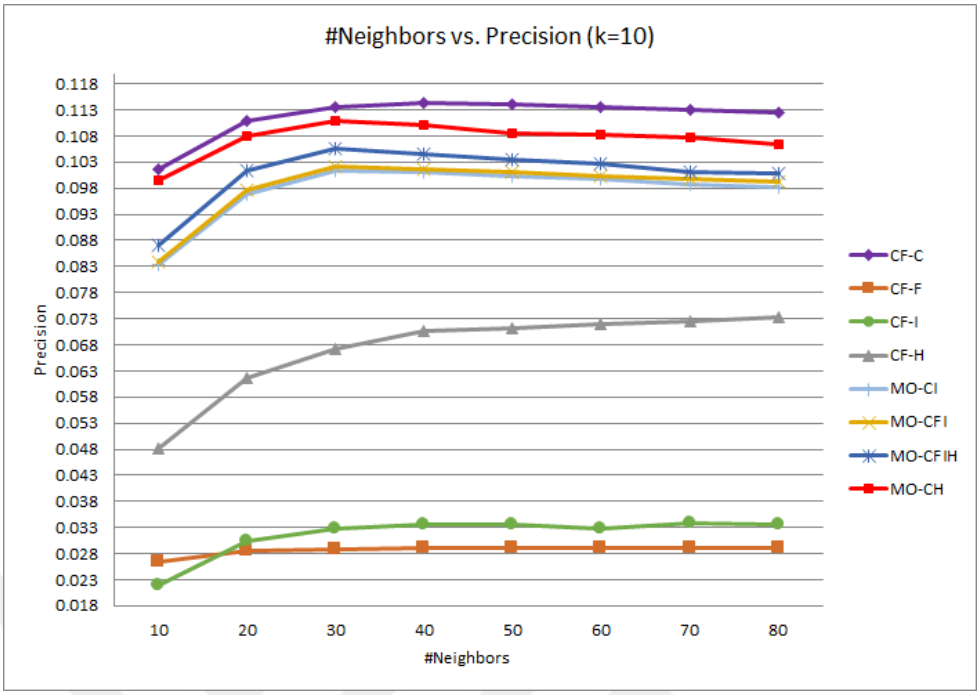


Figure 3.9: N vs Precision when $k = 10$

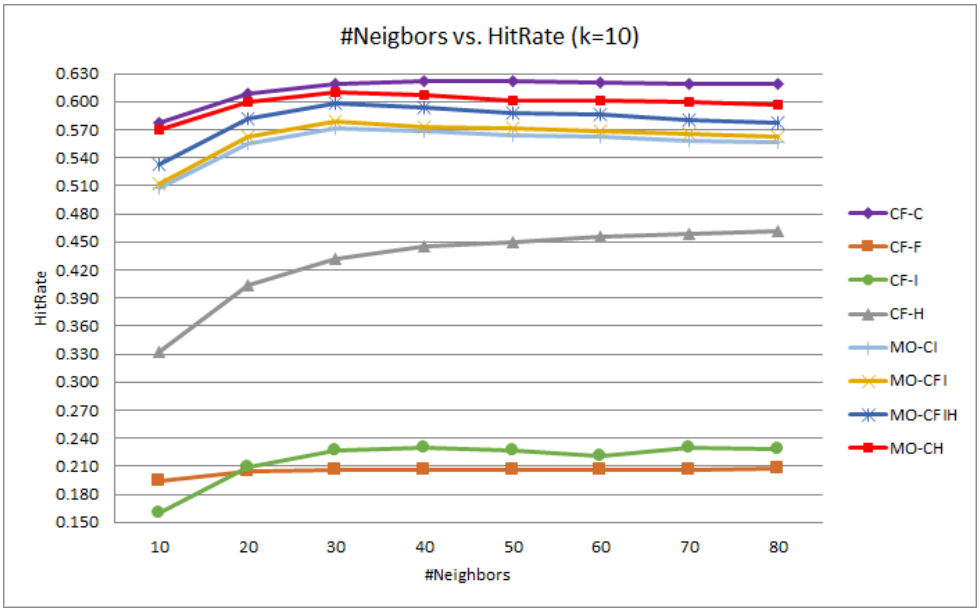


Figure 3.10: N vs Hitrate when $k = 10$

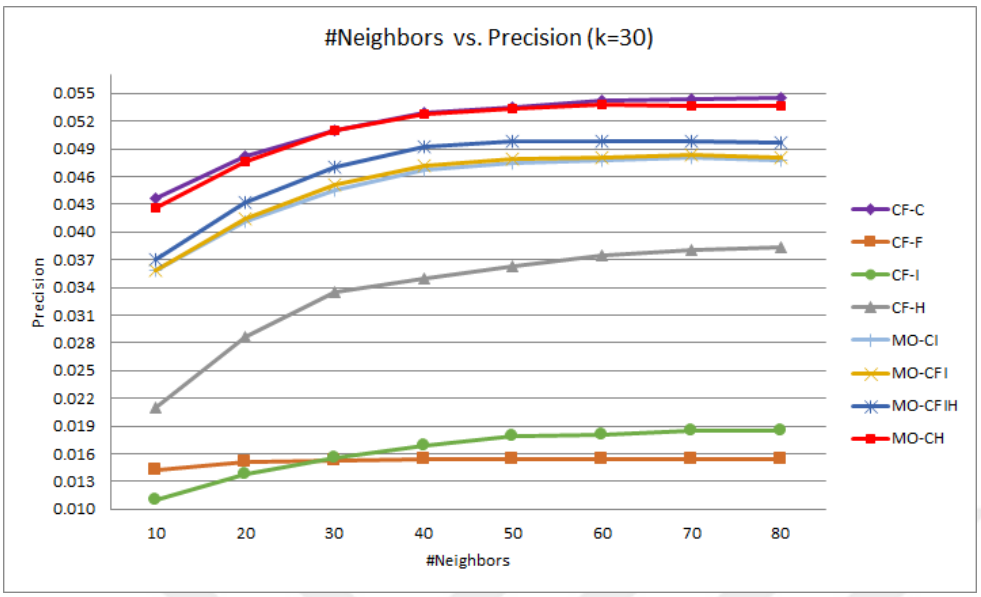


Figure 3.11: N vs Precision when $k = 30$

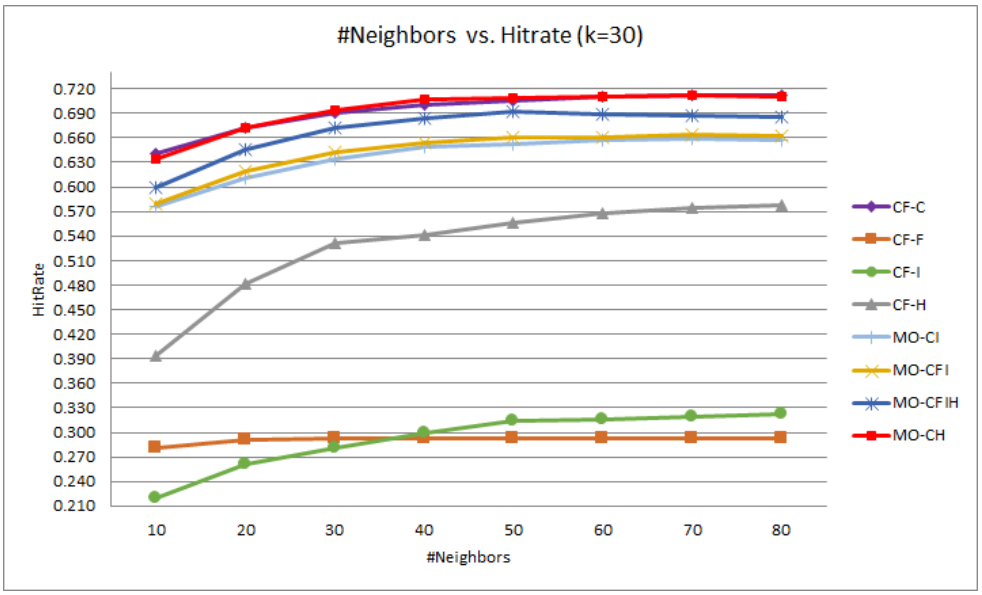


Figure 3.12: N vs Hitrate when $k = 30$

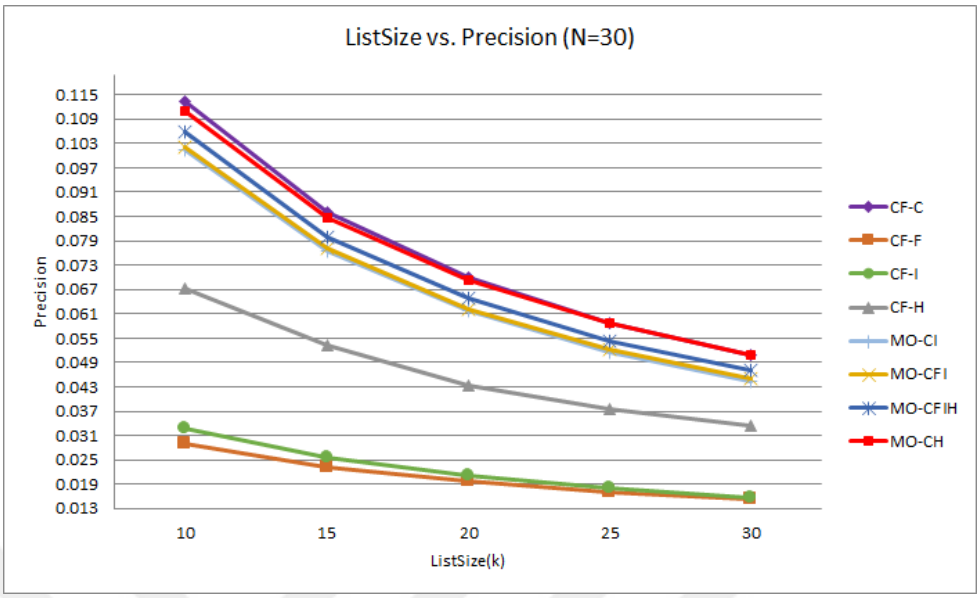


Figure 3.13: k vs Precision when $N = 30$

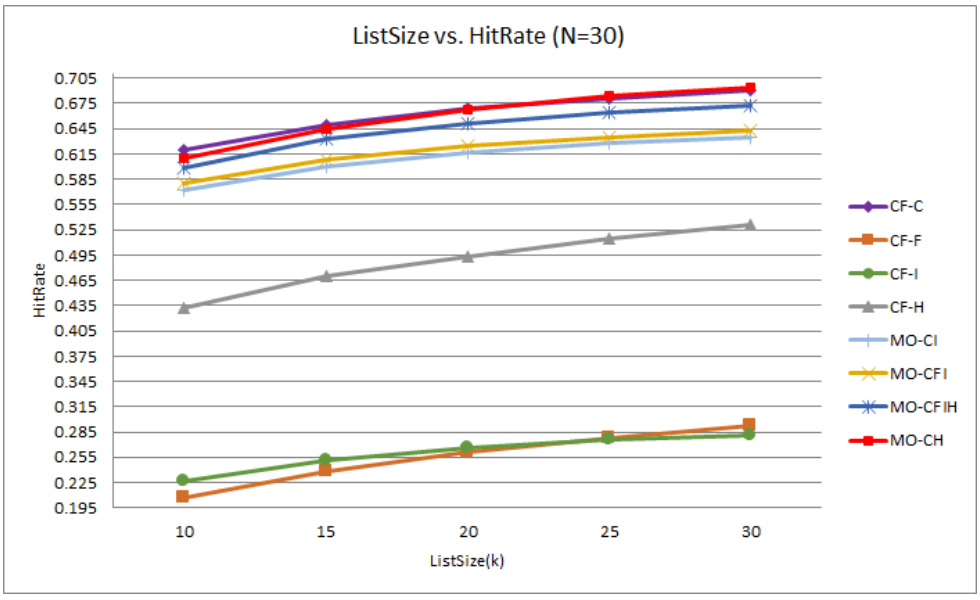


Figure 3.14: k vs Hitrate when $N = 30$

Table 3.3: Number of users with the given precision upper-bound value

Upper-bound Precision	No. of Users	Ratio of Users
0.1	1152	16.0%
0.2	1048	14.6%
0.3	877	12.2%
0.4	753	10.5%
0.5	634	8.8%
0.6	491	6.8%
0.7	421	5.9%
0.8	322	4.5%
0.9	262	3.6%
1.0	1227	17.1%

Results The upper-bounds of the related metrics are as follows: The Ndcg, Hitrate and Coverage metrics' upper-bounds are 1.0. The upper-bound for Precision metric is 0.489. For the upper bound calculations, only the check-in locations that are seen in the CheckinsJan data is considered. For all of the metrics, the users are also limited to the ones who check in both in January and February. The number of users with the precision upper-bound values are given in Table 3.3. Most of the users have 0.1 or 0.2 precision upper-bound values, which shows that these users check in only once or twice during February, our test period. These users are the most challenging ones to make recommendation.

In Table 3.4, the results for Base setting is given. According to the table, the best two methods are CF-C and MO-CH in terms of Precision, Ndcg and Hitrate. The best Coverage is obtained by MO-CFIH method. We observed that in CheckinsJan data with Base settings, the check-in and the home-town features carry more information than the friendship and influence features. Combining all the features increases the coverage near to 1.0, to the upper bound. Even though MO-CH method performs worse than the traditional method, CF-C, its coverage is better. This method (and the other multi-objective based methods) can make recommendations even to the cold start users.

In Table 3.5, the results for Weight Based setting is given. According to the table, the best two methods are CF-C and MO-CFI in terms of Precision, Ndcg and Hitrate.

Table 3.4: Results for Base setting

Method	Precision	Ndcg	HitRate	Coverage
CF-C	**0.114**	**0.242**	**0.621**	0.955
CF-F	0.030	0.064	0.221	0.845
CF-I	0.033	0.067	0.226	0.873
CF-H	0.068	0.132	0.435	0.965
MO-CI	0.102	0.213	0.572	0.993
MO-CFI	0.103	0.213	0.577	0.993
MO-CFIH	0.105	0.218	0.596	**0.999**
MO-CH	0.112	0.227	0.616	0.996

Table 3.5: Results for Weight Based setting

Method	Precision	Ndcg	HitRate	Coverage
CF-C_W	**0.119**	**0.250**	**0.630**	0.955
CF-F_W	0.030	0.064	0.221	0.845
CF-I_W	0.034	0.068	0.229	0.873
CF-H_W	0.068	0.132	0.435	0.965
MO-CI_W	0.105	0.218	0.597	**0.999**
MO-CFI_W	0.111	0.225	0.607	0.996
MO-CFIH_W	0.098	0.207	0.549	0.993
MO-CH_W	0.089	0.193	0.522	0.993

The best Coverage is obtained by MO-CI method. Combining multiple features via multi-objective optimization methods leads to increase in the coverage more than 3%. Comparison with the Base setting results shows us that using weights of neighbors in the calculations slightly increases the performance of the methods.

In Table 3.6, the results for Rate Based setting is given. According to the table, the best Precision, Ndcg and Hitrate results belong to CF-C. It is followed by the multi-objective optimization methods. In terms of Coverage the best result is obtained by MO-CFIH. Introducing rates to the system decreases the performance of the methods which use check-in feature. We observe that introduction of weights does not affect the results significantly.

In Table 3.7, the results for Rate and Weight Based setting is given. According to the table, the best Precision, Ndcg and Hitrate results are obtained by CF-C. The

Table 3.6: Results for Rate Based setting

Method	Precision	Ndcg	HitRate	Coverage
CF-C_R	**0.114**	**0.242**	**0.624**	0.955
CF-F_R	0.034	0.073	0.240	0.845
CF-I_R	0.030	0.063	0.205	0.873
CF-H_R	0.068	0.134	0.436	0.965
MO-CI_R	0.098	0.209	0.559	0.993
MO-CFI_R	0.098	0.210	0.564	0.993
MO-CFIH_R	0.102	0.218	0.587	**0.999**
MO-CH_R	0.111	0.229	0.611	0.996

Table 3.7: Results for Rate and Weight Based setting

Method	Precision	Ndcg	HitRate	Coverage
CF-C_RW	**0.119**	**0.257**	**0.637**	0.955
CF-F_RW	0.034	0.073	0.240	0.845
CF-I_RW	0.031	0.063	0.210	0.873
CF-H_RW	0.068	0.134	0.436	0.965
MO-CI_RW	0.096	0.210	0.546	0.993
MO-CFI_RW	0.089	0.197	0.533	0.993
MO-CFIH_RW	0.103	0.219	0.592	**0.999**
MO-CH_RW	0.109	0.226	0.601	0.996

best Coverage result belongs to MO-CFIH. Combining multiple features via multi-objective optimization methods leads high coverage performance, more than 95%. Comparing the results to the rate based setting, we observe that the performance of the methods changes slightly.

In order to observe how different settings have affected the results, we presented Figures 3.15 to 3.22. The figures show the behavior of different methods on different settings. For the traditional collaborative filtering based methods; namely CF-C, CF-F, CF-I and CF-H; the results are not affected significantly by the weight or rate information. For the multi-objective optimization based methods using home-town information, namely MO-CFIH and MO-CH, the performance decreases with weight usage. However, for the other multi-objective optimization based methods the weight usage increases the performance. All the multi-objective optimization based methods

achieve nearly 1.0 coverage, which is the upper-bound.

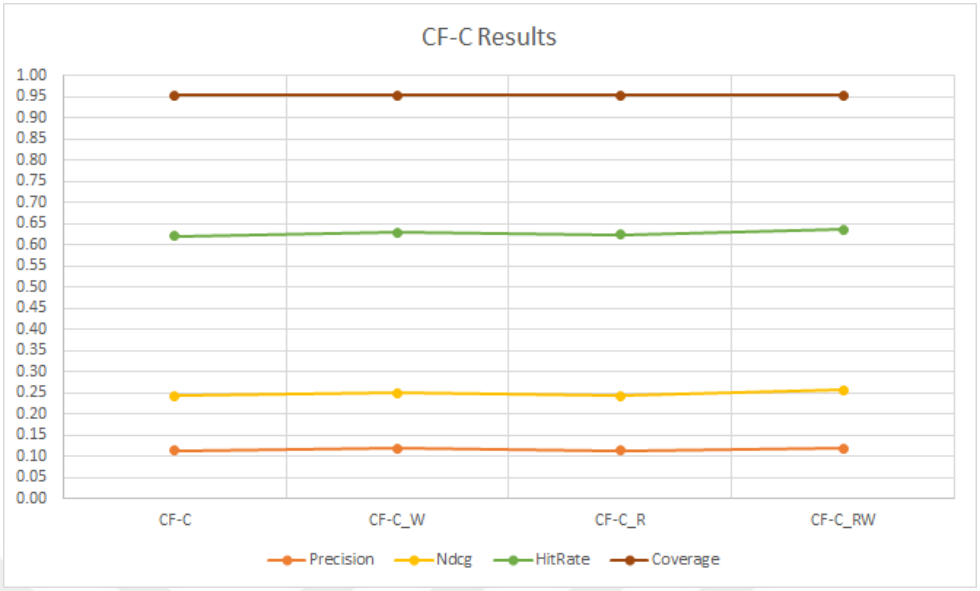


Figure 3.15: Results for different settings: CF-C

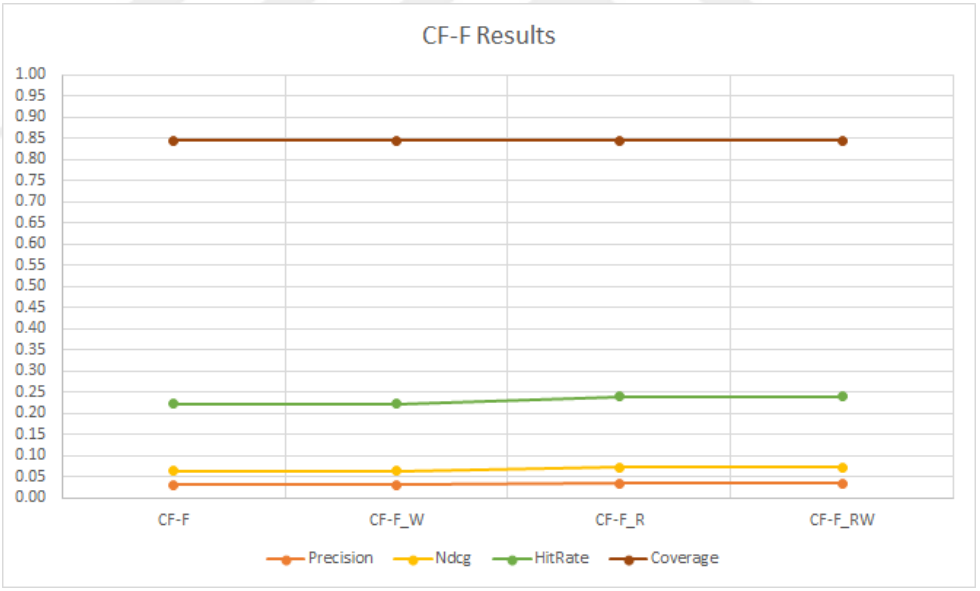


Figure 3.16: Results for different settings: CF-F

In summary, multi-objective optimization based methods use the information provided by different features. This leads to increase in coverage while preserving precision. As stated in [54] and [109], some accuracy loss while adding additional aspects to increase the overall utility is expected.

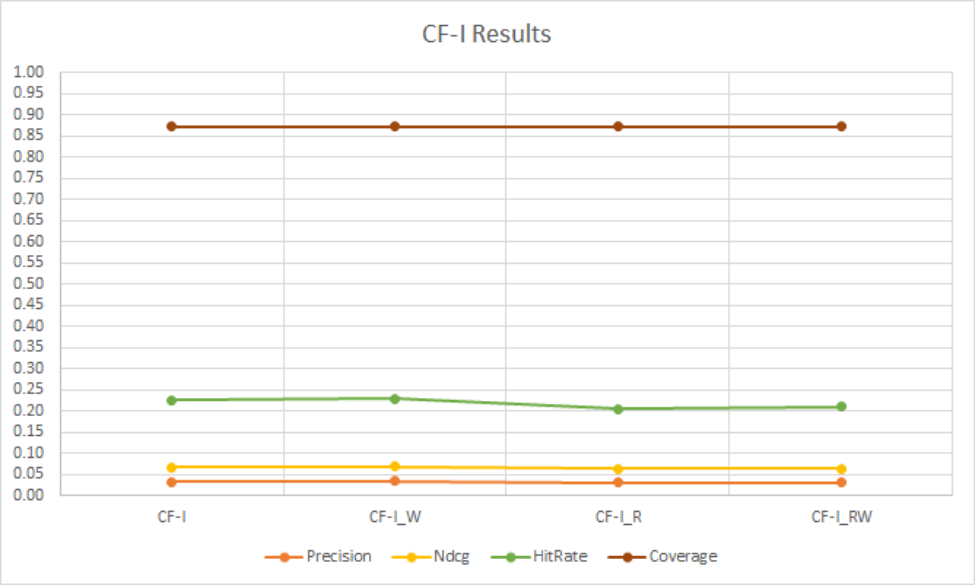


Figure 3.17: Results for different settings: CF-I

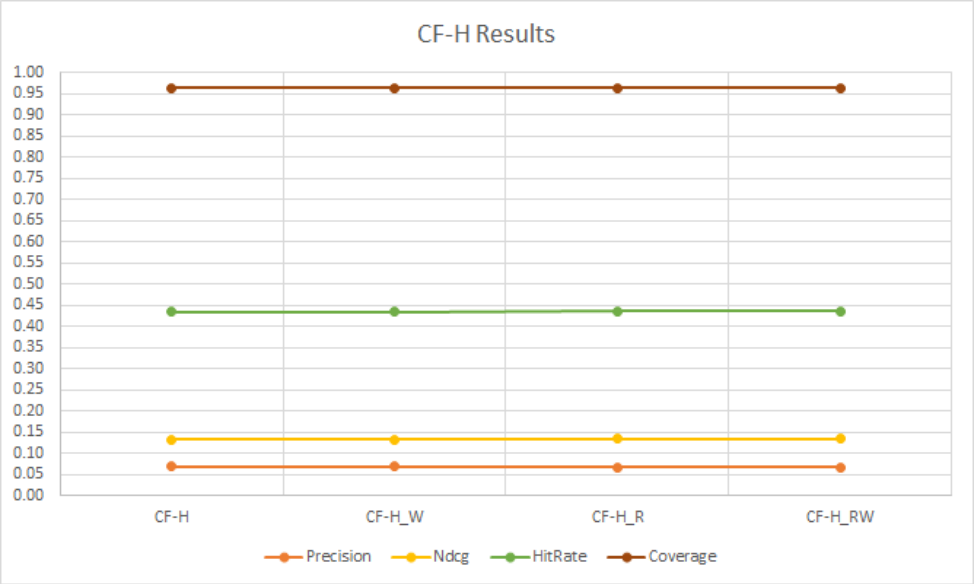


Figure 3.18: Results for different settings: CF-H

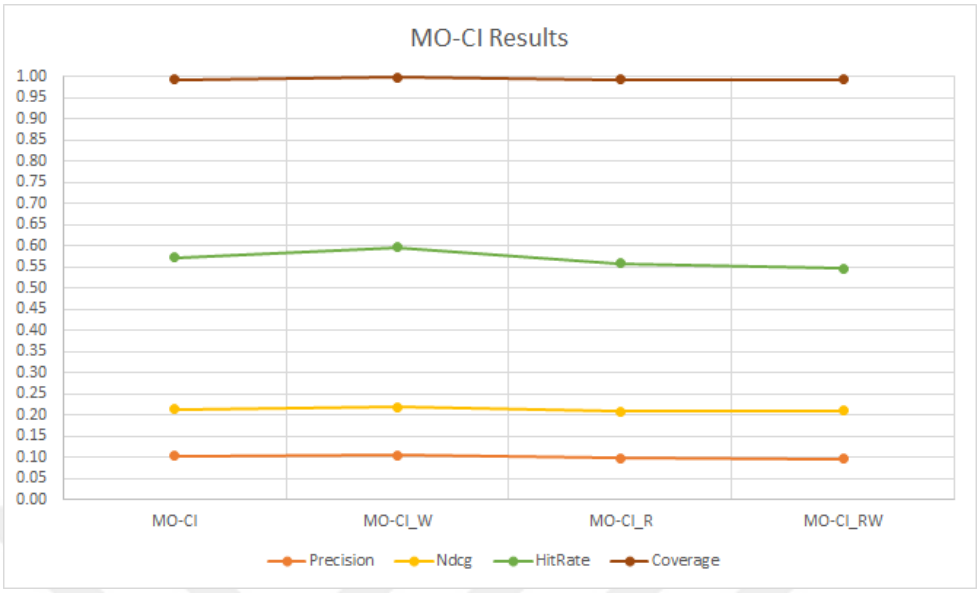


Figure 3.19: Results for different settings: MO-CI

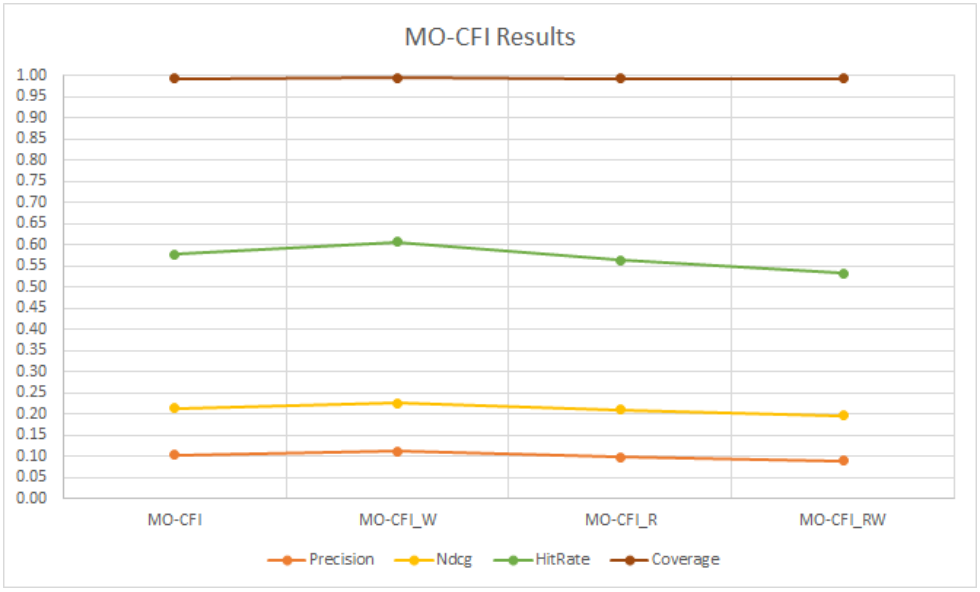


Figure 3.20: Results for different settings: MO-CFI

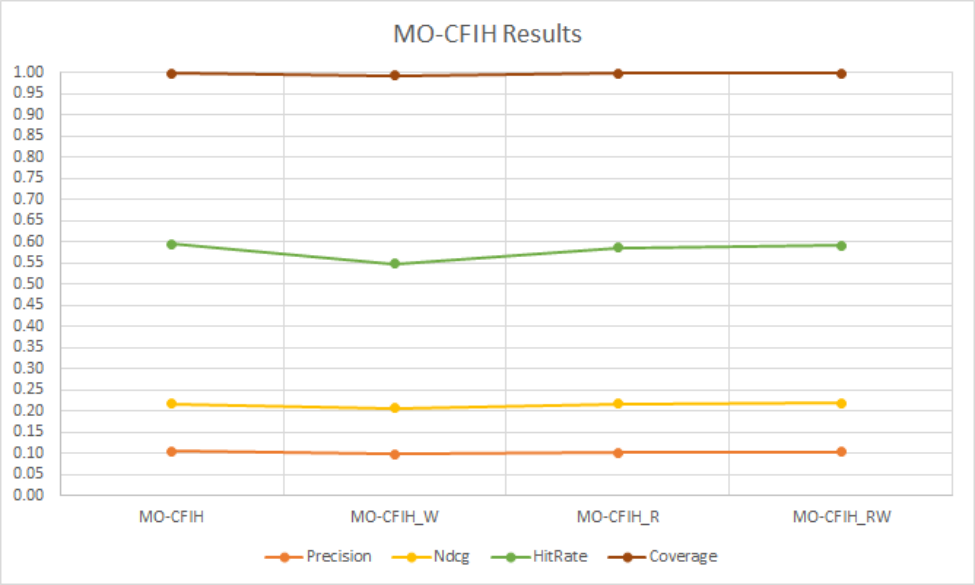


Figure 3.21: Results for different settings: MO-CFIH

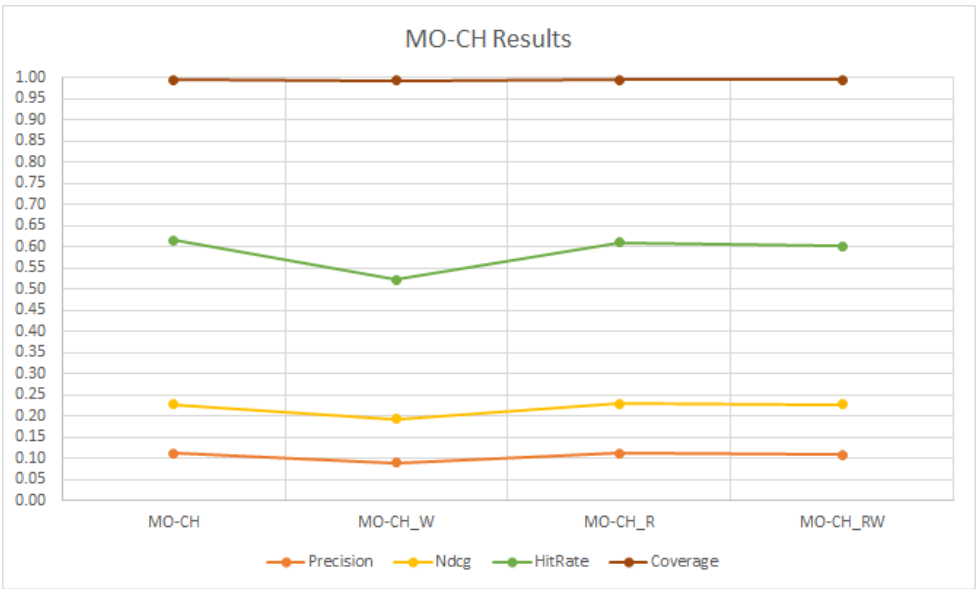


Figure 3.22: Results for different settings: MO-CH

3.7.3.2 Evaluation Results: Multi-Objective Optimization Based Recommendation Enhanced with Inferred Home Locations

In this section, first, we present the configurations that are necessary for the evaluation and then we present the evaluation results and discussions.

Configurations We aim to predict future check-ins of each user using the Checkins-Jan data. In the prediction step there is a need for setting the number of neighbors, N, the output list size, k and the radius, r (related to travel locality). In Section 3.7.3.1 analysis on the data are performed and it is decided to set N to 30 and k to 10. We directly used these values in this section. In order to decide on the radius, r, we performed the following experiments.

The distance between the user and the venue location is decided by Haversine distance, which is calculated as given in [42], by Equation 3.14. In the equation, lon and lat refers to the longitude and latitude of the first and the second locations, and the R is the radius of the Earth and set to 6373 km. Haversine distance gives the great-circle (shortest) distance between two points on a sphere [130]. For the calculation, the inferred home/center locations of the users and the venues are used.

$$
\begin{aligned}
&dlon = lon2 - lon1 \\
&dlat = lat2 - lat1 \\
&a = (\sin(\frac{dlat}{2}))^2 + \cos(lat1) * cos(lat2) * (sin(\frac{dlon}{2}))^2 \\
&c = 2 * atan2(\sqrt{a}, \sqrt{1-a}) \\
&d = R * c
\end{aligned}
\tag{3.14}
$$

We first calculated the average Haversine distance of each user to the locations that they checked as shown in Figure 3.23. In the figure, the calculated average distances are shown as the vertical axis and the anonymous user ids are shown in the horizontal axis. This figure shows that there are few users that checks-in in larger distance, and most of the users usually visit places closer to them. For example, the number of users whose average distance is more than 500 km is less than 1000 and the number of users whose average distance is around 0 km is more than 5000. In our data-set, the average Haversine distance between users' home location and venues center locations

is calculated as 240.44 kilometers (km).

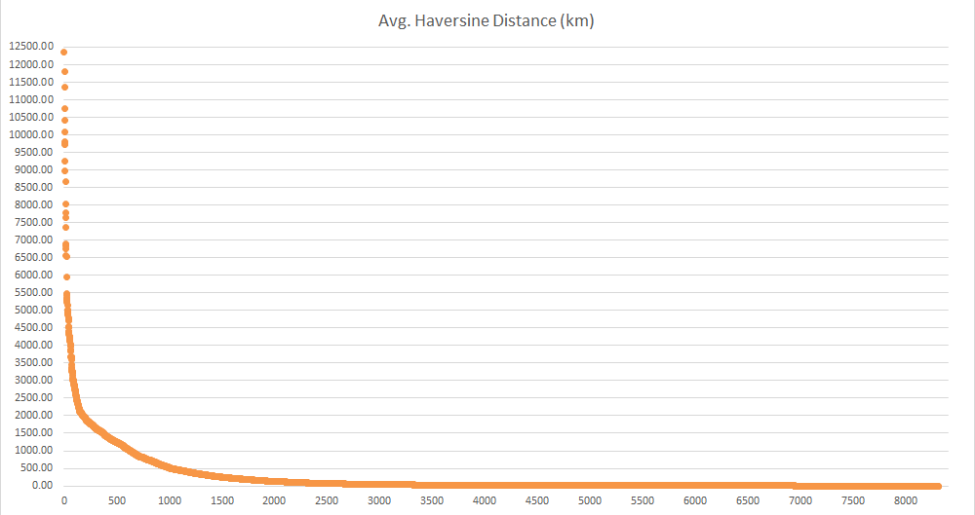


Figure 3.23: Average distances of users to their check-in venues (in km)

We plotted the number of users who visited venues in limited ranges in Figure 3.24. In the figure, the horizontal axis gives the ranges of distance in kilometers and the vertical axis indicates the number of users in the given range. The percentages on the figure indicates the percentage of users whose average distance is in the given range. In the figure, we set the upper value as 2000 km, because in Figure 3.23, we observed that the number of users who visited venues further than 2000 km is small.

In Figure 3.24, we set the ranges using three different approaches. In the first approach, the left part of the figure, we set the ranges in 250 km intervals. After reaching the value 250 km as the lower bound, we set the intervals of 25 km, as seen in the middle part of the figure. After reaching the lower value, 25 km, we changed the interval into 5 km, as seen in the right part of the figure. This figure shows us that most of the users visit venues which are in distance less than 100 km, i.e. about 75% of the users visit venues in 100 km range. Nearly half of those users, i.e. about $1/3$ of all users, visit locations in the range of 5 km. Based on these observations, we decided to set our radius, r, value to 5 km. In order to observe the effect of radius, r, we also set its value to a larger radius. We chose the second value for r as 100 km.

Evaluation Results For the upper bound calculation, the venues are limited to the ones that are seen in the CheckinsJan data and the users are limited to the ones that check in both in January and February. The upper-bounds of the related metrics

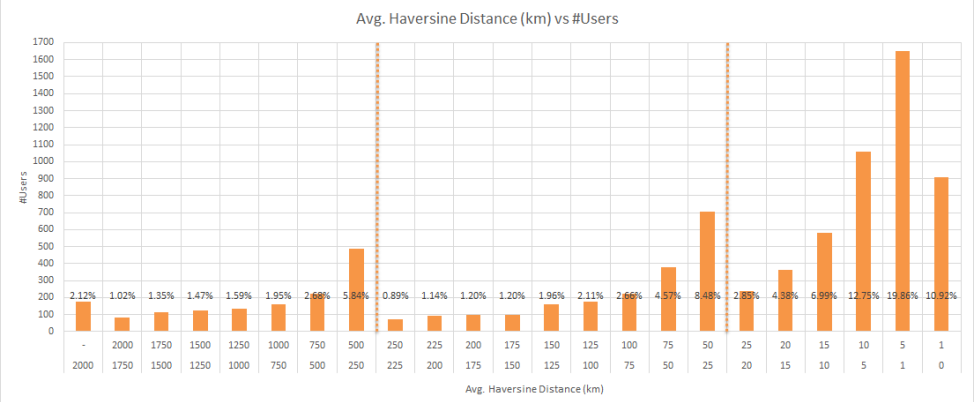


Figure 3.24: Number of users who visited venues in the given ranges

are as follows: When travel locality is not used and all the users in the test set are considered, the upper-bound for the Precision metric is 0.489. When we consider travel locality, the number of users in the given range reduces to 3732 and 5695 for $r = 5km$ and $r = 100km$, respectively. When we consider these users as our target user set, the upper bounds are 0.299 and 0.432 for $r = 5km$ and $r = 100km$, respectively. For the Ndcg, Hitrate and Coverage metrics upper bounds are 1.0, such that the recommendation system should be able to make recommendation to any user. While using travel locality, if we consider all the available users on the test set as the target user set, the upper-bounds of the evaluation metrics reduces. These upper-bounds while considering all the users are presented in the 3.7.3.6. In this section, the evaluation results are given for the users who have at least one check-in in the given travel radius.

Firstly, we analyzed if our home/center location inference method performs well: The Checkins2011 data-set provides us only hometown-ids, without any real world information, i.e. the name of the city, state or country are unknown. The creators of this data-set stated in their work [35] that the hometowns in the data-set are given as either city or state level. Considering this fact, we tried to figure out if the inferred home locations in this section are really close to each other, at the city or state level. In order to decide on the distances in a city and a state we referred to their land area. From [64] we collected the largest 20 cities in the world, and calculated average land area of a city. Then, we calculated the distance by taking the square root of land area, which is about 60 km on average. This value gives us only the information of the

city center. We assumed that a city will be three more times larger than its city center when its suburban areas are also considered. We concluded that a city will have users in 180 km distance at most. Having the city level distance, we decided to set the state-level distance to 720 km with the assumption that a state will have four cities at least. One should note that the given distances are mostly based on assumption and can be larger or smaller in real life.

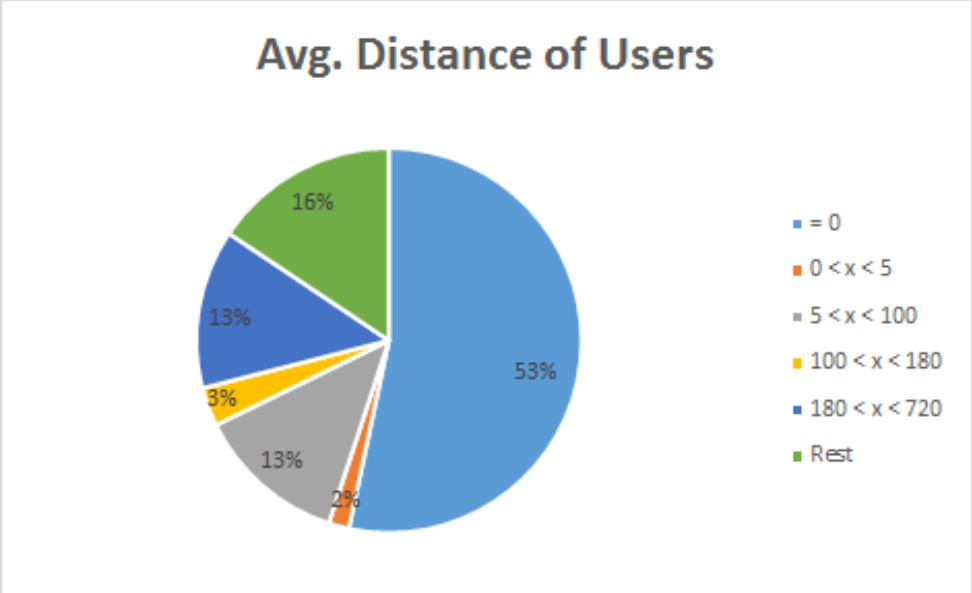


Figure 3.25: User ids and their average distances to their check-in locations (in km)

We calculated the average distance among users who have the same home-town id, which is the only home-town information given in the data-set. In Figure 3.25, we give the ratio of the home-town-ids which have the distance in the given ranges. For example, among its users 2% of the home-towns have distance less than 5 km, on average. We set the ranges as the decided radius values, explained in 3.7.3.1, and city-level and state-level distances, explained in the previous paragraph. In the figure, we observed that more than half of the home-town-ids are used by only one user, so that have 0 km average distance. These users are the ones who either misspelled their location or gave fake/artificial home-town names. When we take into account the distances larger than 0 km, 38% of the home-town-ids have city level distances and 27% of them have state level distances. These results show that our method is mostly able to infer the users' home/center location.

Secondly, we analyzed the performance of multi-objective optimization based recom-

Table 3.8: Results for $r = 5km$

Method	Precision	Ndcg	HitRate	Coverage
IL-CF-C	**0.110**	**0.269**	**0.671**	**0.891**
IL-CF-F	0.037	0.085	0.259	0.514
IL-CF-I	0.040	0.092	0.273	0.533
IL-MO-CI	0.102	0.251	0.633	0.870
IL-MO-CFI	0.104	0.252	0.643	0.876
IL-CF-C_W	**0.113**	**0.273**	**0.678**	**0.891**
IL-CF-F_W	0.037	0.085	0.259	0.514
IL-CF-I_W	0.039	0.092	0.272	0.533
IL-MO-CI_W	0.103	0.252	0.634	0.870
IL-MO-CFI_W	0.100	0.246	0.634	0.876
IL-CF-C_R	**0.113**	**0.275**	**0.682**	0.**895**
IL-CF-F_R	0.038	0.091	0.267	0.514
IL-CF-I_R	0.040	0.093	0.275	0.533
IL-MO-CI_R	0.104	0.257	0.646	0.879
IL-MO-CFI_R	0.106	0.260	0.653	0.885
IL-CF-C_RW	**0.114**	**0.280**	**0.685**	**0.895**
IL-CF-F_RW	0.038	0.091	0.267	0.514
IL-CF-I_RW	0.039	0.093	0.273	0.533
IL-MO-CI_RW	0.105	0.260	0.648	0.879
IL-MO-CFI_RW	0.103	0.255	0.650	0.885

mendation methods while using travel locality or not: In Tables 3.8 and 3.9, the results for different settings when $r = 5km$ and $r = 100km$ are given, respectively. In the tables, the methods with suffix _W, _R and _RW represent the Weight Based, Rate Based, and Rate and Weight Based settings. The methods without any suffix form the Base setting. All other abbreviations, which indicates the type of the methods, are given in Section 3.3.3 while the methods are explained.

In Table 3.8, the results for different settings when $r = 5km$ are given. According to the table, the best method is IL-CF-C. As observed in the previous experiments; the different settings, such as use of weights or rates, do not change the trend of the performance. Even though the best performance is obtained when only check-on information is used, multi-objective optimization based methods which uses a combination of criteria perform nearly as good as the best one. Compared to the upper bounds, different setting for the proposed recommendation system using $r = 5km$

Table 3.9: Results for $r = 100km$

Method	Precision	Ndcg	HitRate	Coverage
IL-CF-C	**0.123**	**0.269**	**0.667**	0.943
IL-CF-F	0.038	0.080	0.263	0.629
IL-CF-I	0.043	0.092	0.288	0.687
IL-MO-CI	0.114	0.251	0.637	0.953
IL-MO-CFI	0.115	0.251	0.642	**0.957**
IL-CF-C_W	**0.127**	**0.278**	**0.675**	0.943
IL-CF-F_W	0.038	0.080	0.263	0.629
IL-CF-I_W	0.044	0.093	0.289	0.687
IL-MO-CI_W	0.112	0.248	0.626	0.953
IL-MO-CFI_W	0.106	0.235	0.609	**0.957**
IL-CF-C_R	**0.125**	**0.275**	**0.674**	0.944
IL-CF-F_R	0.041	0.091	0.283	0.629
IL-CF-I_R	0.043	0.094	0.289	0.687
IL-MO-CI_R	0.115	0.256	0.643	0.955
IL-MO-CFI_R	0.116	0.257	0.648	**0.958**
IL-CF-C_RW	**0.129**	**0.286**	**0.680**	0.944
IL-CF-F_RW	0.041	0.091	0.283	0.629
IL-CF-I_RW	0.044	0.095	0.292	0.687
IL-MO-CI_RW	0.115	0.258	0.641	0.955
IL-MO-CFI_RW	0.109	0.246	0.630	**0.958**

reaches about 38% Precision performance, about 28% Ndcg performance, about 69% Hitrate performance and about 89% Coverage performance.

In Table 3.9, the results for different settings when $r = 100km$ are given. According to the table, the best methods are IL-CF-C and IL-MO-CFI. The performance of the methods increases slightly compared to the when the radius is set to 5 km, shown in the previous table. In terms of coverage using larger range performs about 5% better. Compared to the upper bounds, different setting for the proposed recommendation system using $r = 100km$ reaches about 29% Precision performance, about 28% Ndcg performance, about 67% Hitrate performance and about 96% Coverage performance.

In order to observe how inferring home locations and considering travel locality affected the performance of the recommendation system, we reported results in Table 3.10. The table shows the results presented in Section 3.7.3.1 together with those produced in this section, for Base settings. According to the table, the best Precision,

Table 3.10: Comparison of methods using travel locality and not

Method	Precision	Ndcg	HitRate	Coverage
CF-C	**0.114**	**0.242**	**0.621**	0.955
CF-F	0.030	0.064	0.221	0.845
CF-I	0.033	0.067	0.226	0.873
CF-H	0.068	0.132	0.435	0.965
MO-CI	0.102	0.213	0.572	0.993
MO-CFI	0.103	0.213	0.577	0.993
MO-CFIH	0.105	0.218	0.596	**0.999**
MO-CH	0.112	0.227	0.616	0.996
IL-CF-C (5km)	**0.110**	**0.269**	**0.671**	**0.891**
IL-CF-F (5km)	0.037	0.085	0.259	0.514
IL-CF-I (5km)	0.040	0.092	0.273	0.533
IL-MO-CI (5km)	0.102	0.251	0.633	0.870
IL-MO-CFI (5km)	0.104	0.252	0.643	0.876
IL-CF-C (100km)	**0.123**	**0.269**	**0.667**	0.943
IL-CF-F (100km)	0.038	0.080	0.263	0.629
IL-CF-I (100km)	0.043	0.092	0.288	0.687
IL-MO-CI (100km)	0.114	0.251	0.637	0.953
IL-MO-CFI (100km)	0.115	0.251	0.642	**0.957**

Ndcg and HitRate results are obtained by CF-C and IL-CF-C. The best coverage is obtained by multi-objective methods, namely MO-CFIH and IL-MO-CFI. We observed that larger radius value provides better precision results, as it can give many venues as recommendations. When we compare the methods proposed in this section to the methods described in Section 3.7.3.1, we observe that while using travel locality increases the performance in general. Use of no radius or larger radius increases the coverage.

The precision of smaller radius indicates numerically worse performance, however when we compare the results to the upper bounds, we observe that it is not the case: When we compare the results to the the upper bounds, the methods that do not use travel locality idea reach about 25% Precision, 25.0% Ndcg, 63% HitRate and 99% Coverage performance. For the case when travel locality is considered and radius is set to 5 km, the methods reach 38% Precision, 27% Ndcg, 67% Hitrate and 89% Coverage performance. when the radius is set to 100 km, the methods reach 29% Precision, 27% Ndcg, 67% Hitrate and 96% Coverage performance. These ratios

show that when we consider a smaller radius, the system cannot make recommendations to all of the users, but only to some of them (based on Coverage). However, its prediction performance is better (based on Precision). These results indicate that a recommendation system can use smaller radius to predict the venues that will be visited by the target user, and can use larger radius for the users who cannot be given any recommendation with smaller radius.

3.7.3.3 Evaluation Results: Multi-Objective Optimization Based Recommendation Enhanced with Temporal Information

In this section, first, we present the configurations that are necessary for the evaluation and then we present the evaluation results and discussions for methods that use temporal information in non-dynamic and dynamic settings.

Configurations The necessary parameters that need to be pre-defined; namely the number of neighbors, N, and the output list size, k; are assigned to the same values that are decided in Section 3.7.3.1. The N is set to 30 and k is set to 10.

Performance Results for Non-Dynamic Methods We give performance results of the proposed method, which uses temporal information as well as other features, in different settings. The recommendations in this section do not consider the dynamic nature of user preferences. As in the previous experiments, only the locations that are seen in the CheckinsJan data and the users who check in both in training and test periods are taken into account. The upper-bounds of the performance metrics are as follows: for Ndcg, Hitrate and Coverage, the upper-bound is 1.0 The upper-bound for Precision metric is 0.489. In the tables showing the evaluation results, we presented only the methods that are using time information as one of the criterion. The results for other methods are already given in Section 3.7.3.1 and will be presented at the end of this section while comparing the results.

In Table 3.11, the results for recommendation methods that use time similarity as a criterion are given. According to the table, using only temporal similarity leads to very low performance results for Precision, Ndcg and Hitrate. However, for Coverage it is very informative and makes the recommendation system to be able to make

Table 3.11: Results for non-dynamic methods using time similarity

Method	Precision	Ndcg	HitRate	Coverage
CF-T	0.012	0.019	0.096	**1.000**
MO-CT	0.105	0.213	0.576	**1.000**
MO-CHT	0.107	0.220	0.599	**1.000**
MO-CHFT	**0.108**	**0.221**	0.603	**1.000**
MO-CHFIT	0.107	**0.221**	**0.608**	**1.000**
CF-T_W	0.012	0.019	0.096	**1.000**
MO-CT_W	0.103	0.211	0.570	**1.000**
MO-CHT_W	0.108	0.222	0.603	**1.000**
MO-CHFT_W	**0.109**	**0.225**	0.607	**1.000**
MO-CHFIT_W	0.108	0.223	**0.613**	**1.000**
CF-T_R	0.010	0.016	0.082	**1.000**
MO-CT_R	0.103	0.216	0.576	**1.000**
MO-CHT_R	**0.107**	0.224	0.601	**1.000**
MO-CHFT_R	**0.107**	**0.226**	**0.607**	**1.000**
MO-CHFIT_R	0.105	0.223	0.606	**1.000**
CF-T_RW	0.010	0.016	0.082	**1.000**
MO-CT_RW	0.103	0.215	0.573	**1.000**
MO-CHT_RW	0.108	0.225	0.607	**1.000**
MO-CHFT_RW	**0.109**	**0.229**	**0.615**	**1.000**
MO-CHFIT_RW	0.106	0.225	0.610	**1.000**

recommendations to any user. Adding historical check-in information to the time information provides a huge jump in all metrics. The methods that perform best are the ones that uses check-in, home-town, friendship, influence and time information. All of the methods which use temporal information can make recommendation to any user, so the Coverage performance is 1.0. As previously observed, use of weight or rate do not affect the performance results significantly.

In order to observe how time criteria affected the performance of the recommendation system, we presented Table 3.12. The table shows the results of traditional recommendation systems with single criterion, multi-objective optimization based methods presented in Section 3.7.3.1 and the methods presented in this section. For all of them, we present only the results for Base settings. In Table 3.12, the first group gives the results of the traditional collaborative filtering based recommendation systems which use a single criterion. The results of this group shows that using check-in information

Table 3.12: Comparison of methods using temporal similarity or not

Method	Precision	Ndcg	HitRate	Coverage
CF-C	**0.114**	**0.242**	**0.621**	0.955
CF-F	0.030	0.064	0.221	0.845
CF-I	0.033	0.067	0.226	0.873
CF-H	0.068	0.132	0.435	0.965
CF-T	0.012	0.019	0.096	**1.000**
MO-CI	0.102	0.213	0.572	0.993
MO-CFI	0.103	0.213	0.577	0.993
MO-CFIH	0.105	0.218	0.596	**0.999**
MO-CH	**0.112**	**0.227**	**0.616**	0.996
MO-CT	0.105	0.213	0.576	**1.000**
MO-CHT	0.107	0.220	0.599	**1.000**
MO-CHFT	**0.108**	**0.221**	0.603	**1.000**
MO-CHFIT	0.107	**0.221**	**0.608**	**1.000**

provides better performance in terms of Precision, Ndcg and Hitrate. However, use of temporal information has the ability of covering all of the users. These results show that the use of temporal information together with historical check-in information is promising. The second group in the table includes the results of multi-objective optimization based methods presented in Section 3.7.3.1. These methods use combination of historical check-ins, home-town of users, friendship and influence relation among users, but not temporal check-in similarity. These results show that combining multiple criteria together increases the performance, especially for the ones that use friendship only or influence only cases. The last group of the table belongs to the multi-objective optimization based method using temporal information. We observe that use of time information do not always increase the performance. For example, while there is about 1.2% increase in Hitrate performance comparing the methods MO-CFIH and MO-CHFIT, there is about 0.8% decrease in Hitrate performance comparing the methods MO-CH and MO-CHT. However, the use of temporal similarity is always useful to cover, to make recommendation, to all of the users.

Performance Results for Dynamic Methods In this section, we give performance

Table 3.13: Upper bound of the metrics based on the given temporal preference

Temp.	# Users	Precision
WE_M	1526	0.208
WE_A	2803	0.217
WE_E	3914	0.246
WE_N	3297	0.238
WD_M	1994	0.297
WD_A	3902	0.338
WD_E	4756	0.343
WD_N	4117	0.318

results of the proposed method, which uses all the available criteria and gives dynamic recommendations based on the target user's temporal preferences. Even though we used the Base setting for the experiments, it is possible to use other settings as well. Similar to the previous results, we only considered the check-in locations that are seen in the CheckinsJan data. Also, we limit the users to the ones who checked in in the given temporal preference slot during the test interval, February. For example, we did not take into account a user who asked for a recommendation for a weekend afternoon, but never checked in on that time slot during February.

The number of users and the related precision upper bounds are given in Table 3.13, where WE and WD represent weekend and weekday, respectively, and M, A, E and N represent morning, afternoon, evening and night, respectively. From the table we observe that users tend to check in more on the weekdays and the number of users who check in increases from morning to afternoon and then decreases at the night. For the other metrics, namely Ndcg, HitRate and Coverage, the upper bounds are 1.0.

Actually, for hit-rate metric, the maximum value for each time category is lesser when collaborative filtering is used. In the collaborative filtering based methods, the neighbors' past preferences are used to make recommendations, and the evaluation is performed by comparing the recommended venues to the ones visited during the test period. If there is no possible neighbor in the data-set who visited the venues to be predicted during the training period, it is impossible to give the right answers. For example, assume that our target user is looking for recommendations for a weekend morning and he/she visited only $locA$ on a weekend morning in the test period. Also

Table 3.14: Results for WD_M

Method	Precision	Ndcg	HitRate	Coverage
DTP_CF-C	**0.032**	**0.146**	**0.266**	**0.950**
DTP_CF-F	0.008	0.041	0.077	0.618
DTP_CF-I	0.009	0.041	0.081	0.868
DTP_CF-H	0.015	0.068	0.134	0.910
DTP_CF-T	0.004	0.015	0.034	0.943
DTP_MO-CI	0.029	0.128	0.240	0.995
DTP_MO-CFI	0.030	0.130	0.244	0.995
DTP_MO-CFIH	0.030	0.130	0.248	**0.999**
DTP_MO-CH	**0.031**	**0.136**	**0.256**	0.989
DTP_MO-CT	**0.031**	0.134	0.249	0.970
DTP_MO-CHT	**0.031**	**0.137**	**0.255**	0.980
DTP_MO-CHFT	**0.031**	0.136	0.254	0.986
DTP_MO-CHFIT	0.030	0.134	0.251	**0.995**

assume that there is no other user in the training set that has visited that location in a weekend morning. So, for a collaborative filtering based method, it is not possible to recommend $locA$ to this target user, so it is impossible to have a hit in the recommendation. Note that, if we consider this issue, the hit-rate upper bounds are reduced to the following values: <WE_M: 0.541, WE_A: 0.581, WE_E: 0.732, WE_N: 0.705, WD_M: 0.657, WD_A: 0.713, WD_E: 0.777, WD_N: 0.785>.

The evaluation results for different time categories are given in Tables 3.14 - 3.21. All of the tables are divided into three groups, which are traditional methods using single criterion, multi-objective methods without time criterion and multi-objective methods with all the available criteria. The tables have similar patterns. According to the tables, when we use a single criterion the most effective ones are check-in and home-town information for this data-set. This observation follows the one made in Section 3.7.3.1 as well. We observe that including multiple criteria at once helps to increase coverage while preserving the performance for other metrics.

Table 3.22 show how temporal preference of users and dynamic recommendation affects the performance. The table is divided into three sections containing results of traditional methods using single criterion, multi-objective methods without time criterion and multi-objective methods with all the available criteria. The performance

Table 3.15: Results for WD_A

Method	Precision	Ndcg	HitRate	Coverage
DTP_CF-C	**0.034**	**0.143**	**0.285**	**0.960**
DTP_CF-F	0.008	0.035	0.076	0.730
DTP_CF-I	0.009	0.036	0.080	0.872
DTP_CF-H	0.020	0.081	0.176	0.952
DTP_CF-T	0.003	0.009	0.025	0.955
DTP_MO-CI	0.030	0.126	0.255	0.994
DTP_MO-CFI	0.030	0.127	0.260	0.995
DTP_MO-CFIH	0.031	0.128	0.268	**0.999**
DTP_MO-CH	**0.034**	**0.136**	**0.285**	0.993
DTP_MO-CT	0.032	0.129	0.269	0.973
DTP_MO-CHT	**0.033**	0.135	0.278	0.983
DTP_MO-CHFT	**0.033**	**0.136**	**0.280**	0.992
DTP_MO-CHFIT	0.032	0.133	0.275	**0.997**

Table 3.16: Results for WD_E

Method	Precision	Ndcg	HitRate	Coverage
DTP_CF-C	**0.037**	**0.136**	**0.308**	**0.967**
DTP_CF-F	0.009	0.036	0.086	0.815
DTP_CF-I	0.011	0.043	0.099	0.876
DTP_CF-H	0.023	0.079	0.203	0.963
DTP_CF-T	0.003	0.009	0.027	0.963
DTP_MO-CI	0.033	0.125	0.279	0.996
DTP_MO-CFI	0.033	0.126	0.280	0.996
DTP_MO-CFIH	0.034	0.127	0.288	**1.000**
DTP_MO-CH	**0.036**	**0.129**	**0.302**	0.997
DTP_MO-CT	0.035	0.126	0.288	0.979
DTP_MO-CHT	**0.036**	0.131	0.304	0.991
DTP_MO-CHFT	**0.036**	**0.133**	**0.307**	0.996
DTP_MO-CHFIT	0.035	0.131	0.298	**0.998**

Table 3.17: Results for WD_N

Method	Precision	Ndcg	HitRate	Coverage
DTP_CF-C	**0.031**	**0.112**	**0.261**	0.966
DTP_CF-F	0.007	0.024	0.061	0.795
DTP_CF-I	0.008	0.027	0.076	0.865
DTP_CF-H	0.019	0.063	0.169	**0.967**
DTP_CF-T	0.004	0.010	0.035	0.955
DTP_MO-CI	0.027	0.099	0.235	0.994
DTP_MO-CFI	0.028	0.101	0.239	0.995
DTP_MO-CFIH	0.029	0.104	0.247	**0.999**
DTP_MO-CH	**0.031**	**0.110**	**0.263**	0.998
DTP_MO-CT	0.029	0.104	0.250	0.975
DTP_MO-CHT	0.030	0.107	0.260	0.987
DTP_MO-CHFT	**0.031**	0.106	**0.262**	0.994
DTP_MO-CHFIT	0.030	**0.108**	0.259	**0.998**

Table 3.18: Results for WE_M

Method	Precision	Ndcg	HitRate	Coverage
DTP_CF-C	**0.014**	**0.068**	**0.130**	**0.953**
DTP_CF-F	0.003	0.013	0.027	0.569
DTP_CF-I	0.004	0.019	0.033	0.839
DTP_CF-H	0.007	0.036	0.069	0.916
DTP_CF-T	0.002	0.008	0.017	0.947
DTP_MO-CI	0.012	0.062	0.116	0.991
DTP_MO-CFI	0.012	0.062	0.115	0.991
DTP_MO-CFIH	0.012	**0.063**	0.116	**0.996**
DTP_MO-CH	**0.013**	**0.063**	**0.125**	0.988
DTP_MO-CT	**0.013**	0.061	**0.117**	0.966
DTP_MO-CHT	0.012	0.061	0.115	0.976
DTP_MO-CHFT	**0.013**	0.062	0.116	0.982
DTP_MO-CHFIT	**0.013**	**0.065**	**0.117**	**0.994**

Table 3.19: Results for WE_A

Method	Precision	Ndcg	HitRate	Coverage
DTP_CF-C	**0.016**	**0.086**	**0.149**	**0.960**
DTP_CF-F	0.003	0.016	0.030	0.633
DTP_CF-I	0.003	0.020	0.032	0.859
DTP_CF-H	0.009	0.045	0.081	0.942
DTP_CF-T	0.001	0.005	0.011	0.956
DTP_MO-CI	0.013	0.068	0.118	0.993
DTP_MO-CFI	0.013	0.068	0.118	0.993
DTP_MO-CFIH	0.014	0.074	0.126	**0.999**
DTP_MO-CH	**0.017**	**0.084**	**0.151**	0.996
DTP_MO-CT	**0.015**	0.077	**0.137**	0.974
DTP_MO-CHT	**0.015**	**0.080**	0.139	0.985
DTP_MO-CHFT	**0.015**	0.077	0.136	0.989
DTP_MO-CHFIT	**0.015**	0.077	**0.137**	**0.998**

Table 3.20: Results for WE_E

Method	Precision	Ndcg	HitRate	Coverage
DTP_CF-C	**0.018**	**0.075**	**0.165**	**0.965**
DTP_CF-F	0.004	0.017	0.039	0.795
DTP_CF-I	0.005	0.020	0.046	0.873
DTP_CF-H	0.012	0.048	0.111	0.962
DTP_CF-T	0.001	0.004	0.014	0.955
DTP_MO-CI	0.015	0.065	0.137	0.994
DTP_MO-CFI	0.015	0.066	0.140	0.994
DTP_MO-CFIH	0.016	0.069	0.149	**0.999**
DTP_MO-CH	**0.018**	**0.074**	**0.168**	0.997
DTP_MO-CT	0.016	0.069	0.151	0.978
DTP_MO-CHT	**0.018**	0.072	0.163	0.988
DTP_MO-CHFT	**0.018**	**0.074**	**0.164**	0.995
DTP_MO-CHFIT	0.017	0.072	0.158	**0.998**

Table 3.21: Results for WE_N

Method	Precision	Ndcg	HitRate	Coverage
DTP_CF-C	**0.019**	**0.078**	**0.170**	**0.970**
DTP_CF-F	0.004	0.017	0.042	0.706
DTP_CF-I	0.005	0.019	0.047	0.854
DTP_CF-H	0.014	0.057	0.137	0.966
DTP_CF-T	0.001	0.004	0.012	0.964
DTP_MO-CI	0.016	0.069	0.147	0.994
DTP_MO-CFI	0.016	0.070	0.148	0.994
DTP_MO-CFIH	0.017	0.076	0.161	**1.000**
DTP_MO-CH	**0.019**	**0.081**	**0.178**	0.997
DTP_MO-CT	0.018	0.073	0.162	0.981
DTP_MO-CHT	**0.019**	**0.079**	**0.173**	0.991
DTP_MO-CHFT	**0.019**	0.078	0.172	0.994
DTP_MO-CHFIT	**0.019**	0.078	0.171	**0.998**

scores are the averages of the performance results calculated for each time category. According to the results, when we use a single criterion, the most effective methods are the ones that use check-in and hometown information. This observation follows the one made in the previous sections as well. We observe that including multiple criteria at once helps to increase coverage while preserving the performance for other metrics. When we compare these methods to the ones presented in Table 3.12; non-dynamic methods; we observe that considering dynamicity while giving recommendation performs better in terms of Precision (note that for non-dynamic setting the Precision upper bound is 0.489 and for dynamic setting it is 0.276, on the average). However, the Ndcg and Hitrate results show that it is much harder to give at least one true recommendation to the target, when he/she indicates a temporal preference. This can be rooted from the fact that as we include more restriction on the recommendation, the available data becomes much sparser and the process of making recommendation becomes harder.

3.7.3.4 Evaluation Results: Multi-Objective Optimization Based Recommendation Enhanced with Clustering

In this section, first, we present the configurations that are necessary for the evaluation and then we present the evaluation results and discussions.

Table 3.22: Comparison of dynamic results

Method	Precision	Ndcg	HitRate	Coverage
DTP_CF-C	**0.025**	**0.106**	**0.217**	**0.961**
DTP_CF-F	0.006	0.025	0.055	0.708
DTP_CF-I	0.007	0.028	0.062	0.863
DTP_CF-H	0.015	0.060	0.135	0.947
DTP_CF-T	0.002	0.008	0.022	0.955
DTP_MO-CI	0.022	0.093	0.191	0.994
DTP_MO-CFI	0.022	0.094	0.193	0.994
DTP_MO-CFIH	0.023	0.096	0.200	**0.999**
DTP_MO-CH	**0.025**	**0.102**	**0.216**	0.994
DTP_MO-CT	0.024	0.097	0.203	0.975
DTP_MO-CHT	0.024	**0.100**	**0.211**	0.985
DTP_MO-CHFT	**0.025**	**0.100**	**0.211**	0.991
DTP_MO-CHFIT	0.024	**0.100**	0.208	**0.997**
CF-C	**0.114**	**0.242**	**0.621**	0.955
CF-F	0.030	0.064	0.221	0.845
CF-I	0.033	0.067	0.226	0.873
CF-H	0.068	0.132	0.435	0.965
CF-T	0.012	0.019	0.096	**1.000**
MO-CI	0.102	0.213	0.572	0.993
MO-CFI	0.103	0.213	0.577	0.993
MO-CFIH	0.105	0.218	0.596	**0.999**
MO-CH	**0.112**	**0.227**	**0.616**	0.996
MO-CT	0.105	0.213	0.576	**1.000**
MO-CHT	0.107	0.220	0.599	**1.000**
MO-CHFT	**0.108**	**0.221**	0.603	**1.000**
MO-CHFIT	0.107	**0.221**	**0.608**	**1.000**

Table 3.23: Results of Base setting and Clustering based on hometown

Method	Precision	Ndcg	HitRate	Coverage
CLH-CF-C	**0.097**	**0.203**	0.549	0.848
CLH-CF-F	0.025	0.050	0.180	0.530
CLH-CF-I	0.050	0.099	0.329	0.753
CLH-MO-CI	0.095	0.196	0.549	0.922
CLH-MO-CFI	0.095	0.196	**0.551**	**0.923**

Configurations There is a need for setting the number of neighbors, N, and the output list size, k. In Section 3.7.3.1, we performed analysis on the data and decided to set N to 30 and k to 10. We directly used these values in this section.

Evaluation Results The upper bound is calculated for Precision as 0.489, and for the other measures as 1.0. In Section 3.7.3.1, it is shown that more than 30% of the users have 1 or 2 check-ins in the test period. This means that making the correct recommendation to these users is a challenging task.

The results for Base setting while clustering on either home-town or friendship are given, in Tables 3.23 and 3.24, respectively. According to Table 3.23, using historical check-in information provides the best result in terms of Precision and Ndcg. According to Table 3.24, the best methods are CLF-MO-CH and CLF-MO-CIH. We observed that after clustering users based on their friendship, the most informative criteria are check-in and home-town. This result confirms the observation made in Section 3.7.3.1. For both clustering approaches, we observed that the multi-objective based methods give better Hitrate and Coverage results while preserving Precision. High Hitrate and Coverage performance indicate that these methods can make recommendations to more target users than single criterion based methods.

Reported in Tables 3.25 and 3.26 are the results of Weight Based setting when clustering on home-town or friendship is applied. Similar to the previous table, cases which use check-in and home-town information perform better than others and using multi-objective optimization methods help to increase coverage. Compared to the Base setting, the use of weights generally improves the performance of the methods.

In Tables 3.27 and 3.28 the results of Rate Based setting and in Tables 3.29 and 3.30

Table 3.24: Results of Base setting and Clustering based on friendship

Method	Precision	Ndcg	HitRate	Coverage
CLF-CF-C	0.032	0.068	0.216	0.400
CLF-CF-I	0.022	0.047	0.156	0.534
CLF-CF-H	0.025	0.050	0.179	0.530
CLF-MO-CI	0.030	0.063	0.216	0.620
CLF-MO-CH	0.034	**0.072**	**0.238**	0.608
CLF-MO-CIH	**0.043**	0.063	0.213	**0.701**

Table 3.25: Results of Weight Based setting and Clustering based on hometown

Method	Precision	Ndcg	HitRate	Coverage
CLH-CF-C_W	**0.101**	**0.209**	**0.558**	0.848
CLH-CF-F_W	0.025	0.050	0.180	0.530
CLH-CF-I_W	0.049	0.099	0.325	0.753
CLH-MO-CI_W	0.094	0.196	0.543	0.922
CLH-MO-CFI_W	0.087	0.185	0.520	**0.923**

Table 3.26: Results of Weight Based setting and Clustering based on friendship

Method	Precision	Ndcg	HitRate	Coverage
CLF-CF-C_W	0.033	0.071	0.222	0.400
CLF-CF-I_W	0.023	0.050	0.158	0.534
CLF-CF-H_W	0.025	0.050	0.179	0.530
CLF-MO-CI_W	0.034	0.072	0.234	0.620
CLF-MO-CH_W	0.036	**0.075**	**0.244**	0.608
CLF-MO-CIH_W	**0.050**	0.074	0.240	**0.701**

Table 3.27: Results of Rate Based setting and Clustering based on hometown

Method	Precision	Ndcg	HitRate	Coverage
CLH-CF-C_R	**0.097**	**0.203**	0.553	0.848
CLH-CF-F_R	0.027	0.056	0.192	0.530
CLH-CF-I_R	0.049	0.098	0.323	0.753
CLH-MO-CI_R	0.095	0.197	0.554	0.922
CLH-MO-CFI_R	0.095	0.197	**0.555**	**0.923**

Table 3.28: Results of Rate Based setting and Clustering based on friendship

Method	Precision	Ndcg	HitRate	Coverage
CLF-CF-C_R	0.035	0.076	0.231	0.400
CLF-CF-I_R	0.025	0.053	0.170	0.534
CLF-CF-H_R	0.027	0.056	0.192	0.530
CLF-MO-CI_R	0.034	0.072	0.236	0.620
CLF-MO-CH_R	**0.038**	**0.081**	**0.253**	0.608
CLF-MO-CIH_R	0.035	0.075	0.245	**0.701**

the results of Rate and Weight Based setting when clustering on home-town or friendship are given. Similar results to the previous tables are obtained: Use of check-in and home-town information provides the best results. Use of multi-objective optimization increases the coverage. Compared to the Base setting, use of rates slightly increases the performance of the methods.

Table 3.31 includes the results presented in Section 3.7.3.1 and in this section (for Base setting) to show how the clustering step affected the performance of the recommendation system. According to Table 3.31, the best precision results are usually obtained by the methods which use historical check-in information. As already ob-

Table 3.29: Results of Rate and Weight Based setting and Clustering based on hometown

Method	Precision	Ndcg	HitRate	Coverage
CLH-CF-C_RW	**0.102**	**0.214**	**0.563**	0.848
CLH-CF-F_RW	0.027	0.056	0.192	0.530
CLH-CF-I_RW	0.047	0.096	0.316	0.753
CLH-MO-CI_RW	0.094	0.199	0.546	0.922
CLH-MO-CFI_RW	0.088	0.189	0.527	**0.923**

Table 3.30: Results of Rate and Weight Based setting and Clustering based on friendship

Method	Precision	Ndcg	HitRate	Coverage
CLF-CF-C_RW	0.036	0.079	0.235	0.400
CLF-CF-I_RW	0.025	0.055	0.171	0.534
CLF-CF-H_RW	0.027	0.056	0.192	0.530
CLF-MO-CI_RW	**0.038**	**0.082**	0.251	0.620
CLF-MO-CH_RW	**0.038**	0.081	0.256	0.608
CLF-MO-CIH_RW	**0.038**	**0.082**	**0.257**	**0.701**

Table 3.31: Comparison of the results of the clustering methods to traditional and multi-objective optimization methods

Method	Precision	Ndcg	HitRate	Coverage
CF-C	**0.114**	**0.242**	**0.621**	0.955
CF-F	0.030	0.064	0.221	0.845
CF-I	0.033	0.067	0.226	0.873
CF-H	0.068	0.132	0.435	**0.965**
MO-CI	0.102	0.213	0.572	0.993
MO-CFI	0.103	0.213	0.577	0.993
MO-CFIH	0.105	0.218	0.596	**0.999**
MO-CH	**0.112**	**0.227**	**0.616**	0.996
CLH-CF-C	**0.097**	**0.203**	0.549	0.848
CLH-CF-F	0.025	0.050	0.180	0.530
CLH-CF-I	0.050	0.099	0.329	0.753
CLH-MO-CI	0.095	0.196	0.549	0.922
CLH-MO-CFI	0.095	0.196	**0.551**	**0.923**
CLF-CF-C	0.032	0.068	0.216	0.400
CLF-CF-I	0.022	0.047	0.156	0.534
CLF-CF-H	0.025	0.050	0.179	0.530
CLF-MO-CI	0.030	0.063	0.216	0.620
CLF-MO-CH	0.034	**0.072**	**0.238**	0.608
CLF-MO-CIH	**0.043**	0.063	0.213	**0.701**

served, this shows that check-in information is the most valuable information for this data-set. For all measures, the performance reduced when we apply the clustering step. This indicates that considering only a subset of users is not helpful in terms of accuracy (Precision/Ndcg) or coverage (Hitrate/Coverage). Further analysis of the data showed that the number of users from the same home-town and the number of friends a user has is limited (Figures 3.26 and 3.27); this leads to poor recommendation performance.

Figure 3.26 shows the number of home-towns (y-axis) that contain given number of users (x-axis). In the CheckinsJan data-set, the number of home-towns with only one user is 311. The figure reveals that many home-towns contain few users (less than 8). The data-set contains users from 583 different home-towns and 542 of them have less than 30 users. This analysis shows that clustering based on home-town leads to few users in each cluster. This leads to poor recommendations as the number of neighbors to provide recommendation is very small.

Figure 3.27 shows the number of users (y-axis) with given number of friends (x-axis). This figure reveals that in the CheckinsJan data-set most users have limited number of friends (less than 12), while few of them have many friends (more than 32). In this data-set 987 of the users do not have any friend, which is not shown in the figure since it is in log_2 scale. The analysis shows that when we perform clustering on friendship, the output clusters contain limited number of candidate neighbors. This leads to poor performance as the number of neighbors who will recommend items to the target is limited. We conjecture that performing clustering on different data-sets, where number of users in a home-town or number of friends of a user is high, will lead to better accuracy and coverage performance.

In order to observe the effect of clustering on scalability, we measured the time requirements of the methods. Table 3.32 shows the total time used by the methods for all users. We performed all the experiments on an Intel Core2 Duo - 3.17 GHz machine with 8GB RAM and using MySql database. Experimentally, we found that on average the similarity calculations require 1.18 seconds. While similarity is calculated for all users in the reference methods based on Section 3.7.3.1, only elements of the clusters are used in the proposed clustering based methods. We calculated

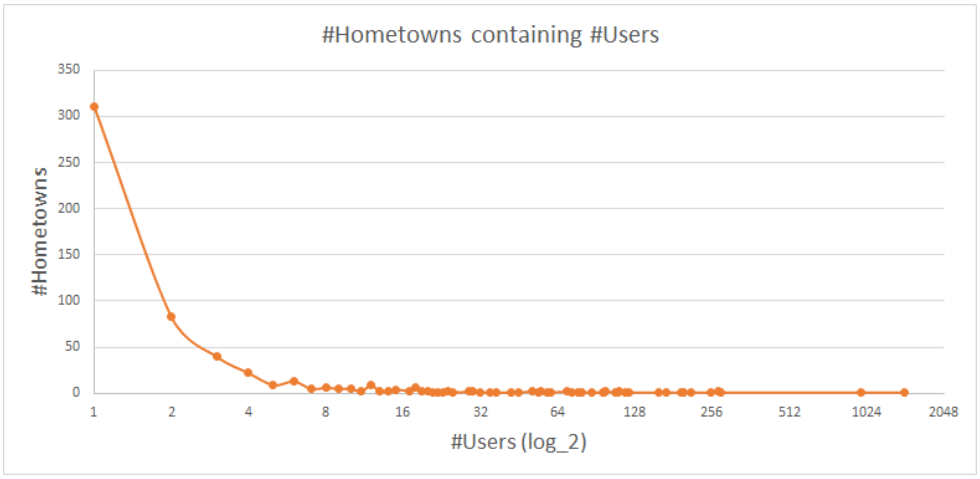


Figure 3.26: Number of home-towns with the given number of users (in log_2 scale)

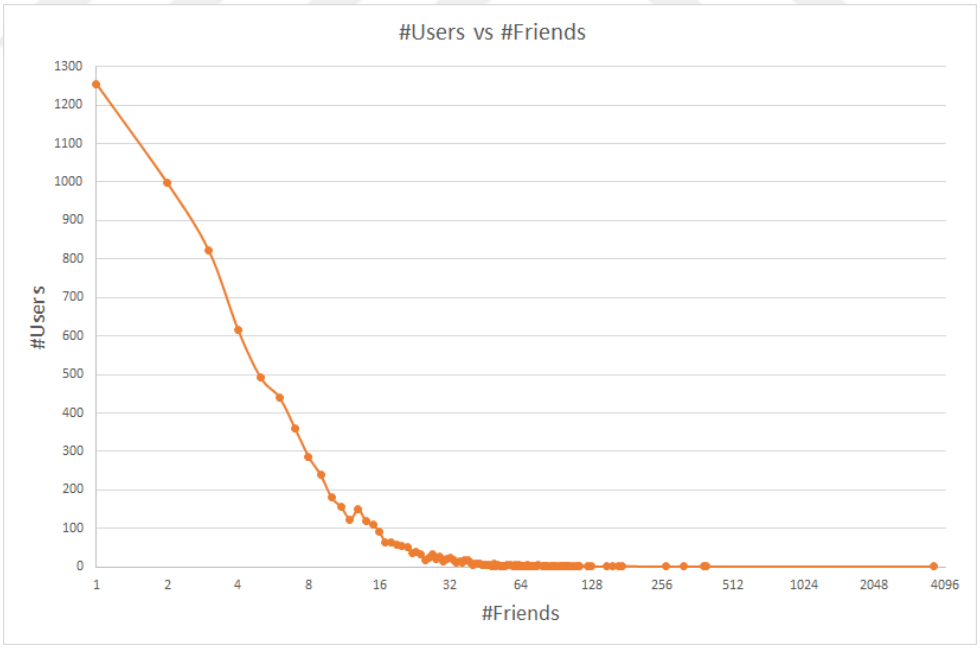


Figure 3.27: Number of users with the given number of friends (in log_2 scale)

Table 3.32: Time requirements for similarity calculations, recommendtion computation and total in seconds

Method	Sim. calc.	Rec. comp.	Total
CF-C	8308*1.18 = 9803.44	6367.71	16171.15
CF-F	9803.44	4966.10	**14769.54**
CF-I	9803.44	10138.00	19941.44
CF-H	9803.44	6831.67	16635.11
MO-CI	9803.44	8663.89	18467.33
MO-CFI	9803.44	7234.28	**17037.72**
MO-CFIH	9803.44	7841.41	17644.85
MO-CH	9803.44	7768.91	17572.35
CLH-CF-C	14.25*1.18 = 16.82	5685.34	5702.16
CLH-CF-F	16.82	2515.83	**2532.65**
CLH-CF-I	16.82	4779.59	4796.41
CLH-MO-CI	16.82	6697.83	6714.65
CLH-MO-CFI	16.82	6814.51	6831.33
CLF-CF-C	7.94*1.18 = 9.37	1769.25	**1778.62**
CLF-CF-I	9.37	3081.61	3090.98
CLF-CF-H	9.37	2456.14	2465.51
CLF-MO-CI	9.37	3016.80	3026.17
CLF-MO-CH	9.37	2940.43	2949.80
CLF-MO-CIH	9.37	4739.75	4749.12

that, on average, clusters based on home-town contain 14.25 users and clusters based on friendship contain 7.94 users. We reported these values for the related methods in Table 3.32 in the similarity calculations column. Similarity calculations are performed for each criterion for methods which use multiple criteria. As the similarity calculation for each criterion is independent from others, we include the similarity calculation time only once for those methods. The overall analysis on time requirements show that clustering improves the time performance of the recommendation system by reducing the number of users who will be considered in the calculations.

3.7.3.5 Evaluation Results: Hybrid Recommendation Enhanced with Clustering

In this section, first, we present the configurations that are necessary for the evaluation and then we present the evaluation results and discussions.

Table 3.33: α_R values

Method	α_0	α_1	α_2	α_3
W-PV-S-HT-CH	0.9	0.1	-	-
W-PV-S-HT-CI	0.9	0.1	-	-
W-PV-S-HT-CFI	0.9	0.06	0.04	-
W-PV-S-HT-CFIH	0.9	0.06	0.02	0.02
W-PW-S-HT-CH	0.9	0.1	-	-
W-PW-S-HT-CI	0.9	0.1	-	-
W-PW-S-HT-CFI	0.9	0.08	0.02	-
W-PW-S-HT-CFIH	0.9	0.08	0.01	0.01

Configurations In this section, we aim to predict the check-ins of users in February using the data in January, namely the CheckinsJan data. In the prediction step, there is a need for setting the number of neighbors, N, and the output list size, k. In Section 3.7.3.1, we performed analysis on the data and decided to set N to 30 and k to 10. We directly used these values in this section. Another configuration we need to decide on is criteria ranking which is needed for the Weighted hybridization technique with Parameter and Rank based settings. We assigned the ranking as $< Check-in, Hometown, Influence, Friendship >$, based on the observations we made on the results in Section 3.7.3.1.

The last configuration is related to values of α_R used by the Weighted hybridization technique with Parameter and Search based settings. The value of α_R of each criterion is decided based on the methods explained in Section 3.6.2. Depending on the type of scoring (Vote based or Weight based), the best α_R values are found to be different (See Table 3.33). In the table the abbreviations of the methods are used, the meaning of the abbreviations are given in Section 3.6.3. In Table 3.33, the index of the values of α_R are in the same order as the features used in the related method. For example, for the method W-PV-S-HT-CFI, the value of α_R for Check-in(C) is 0.9, for Friendship(F) is 0.06 and for Influence(I) is 0.04.

Evaluation Results The upper bound is calculated for Precision as 0.489, and for the other measures as 1.0. In Section 3.7.3.1, it is shown that in the test period more than 30% of the users have 1 or 2 check-ins and less than 38% of the users have more than 5 check-ins. These statistics indicate that it is a challenging data-set for venue

Table 3.34: Results of Mixed(M) technique

Method	Precision	Ndcg	HitRate	Coverage
M-HT-CH	**0.108**	0.226	**0.612**	0.996
M-HT-CI	0.107	**0.229**	0.601	0.993
M-HT-CFI	0.107	**0.229**	0.602	0.993
M-HT-CFIH	0.105	0.223	0.608	**0.999**

recommendation task.

The results for the Mixed (M) technique are given in Table 3.34; these show that each method has its own strength. While using all the features, i.e., M-HT-CFIH, we reach the best Coverage; we obtain the best HitRate and Precision when we use M-HT-CH. In terms of Ndcg, the best methods are M-HT-CI and M-HT-CFI. We observed that the method which uses all the criteria can make recommendation nearly to all users, however its Precision and Ndcg performance are lower than others. This can be related to the fact that this method gives recommendation even to cold-start users to whom giving recommendation is challenging. All other methods perform close to each other in terms of Precision and Ndcg, with less than 0.3% difference.

The results for Weighted-Vote (W-V) and Weighted-Weight (W-W) techniques are given in Table 3.35. We observed that combining the methods that use each available criterion increases Coverage. Using votes instead of weights increases the performance of the system. This may indicate that a less similar neighbor's choice can be important while making a recommendation. Also, giving equal weights to candidate neighbors from different recommendation methods can provide more diverse neighbor lists and can be helpful to make better recommendations.

The results for Weighted-Parameter&Vote (W-PV) techniques with Rank (R) and Search (S) approaches are given in Table 3.36; these results show that rank or search based approaches perform better than equally weighted approaches (W-V). This means that some of the criteria are more important than others. We used results of the previous section, Section 3.7.3.1, to decide on the rank. The rank and search based approaches perform nearly equally well, however searching the space needs an extra time to decide on the value of α. This shows that knowing characteristics of the data

Table 3.35: Results of Weighted(W) and Vote(V)/Weight(W) techniques

Method	Precision	Ndcg	HitRate	Coverage
W-V-HT-CH	**0.105**	0.219	**0.607**	0.996
W-V-HT-CI	0.103	**0.221**	0.591	0.993
W-V-HT-CFI	0.103	0.220	0.594	0.993
W-V-HT-CFIH	0.099	0.209	0.586	**0.999**
W-W-HT-CH	**0.102**	**0.211**	**0.580**	0.996
W-W-HT-CI	0.083	0.179	0.481	0.993
W-W-HT-CFI	0.085	0.182	0.498	0.993
W-W-HT-CFIH	0.099	0.204	0.571	**0.999**

Table 3.36: Results of Weighted(W) and Parameter&Vote(PV) techniques

Method	Precision	Ndcg	HitRate	Coverage
W-PV-R-HT-CH	**0.112**	0.231	**0.615**	0.996
W-PV-R-HT-CI	**0.112**	**0.233**	0.612	0.993
W-PV-R-HT-CFI	**0.112**	**0.233**	0.613	0.993
W-PV-R-HT-CFIH	0.111	0.228	0.610	**0.999**
W-PV-S-HT-CH	**0.112**	0.231	**0.615**	0.996
W-PV-S-HT-CI	**0.112**	0.233	0.612	0.993
W-PV-S-HT-CFI	**0.112**	**0.234**	**0.615**	0.993
W-PV-S-HT-CFIH	**0.112**	0.231	**0.615**	**0.999**

beforehand is very valuable.

The results for Weighted-Parameter&Weight (W-PW) techniques with Rank (R) and Search (S) approaches are given in Table 3.37. Unlike the previous results, we observed that searching the space to assign an importance value to each criterion is essential compared to using preassigned rankings of the criteria. However, search based results reported in Table 3.37 don't beat the results of the cases when we assigned same importance values to neighbors (W-PV techniques). From this observation we can conclude that if we want to use Weighted (W) technique it is better to give equal importance to each neighbor and to use pre-assigned α_R values for each method using different criterion instead of searching the space to decide on the α_R values (since it is time consuming).

In order to observe the difference in performance between traditional recommendation methods, multi-objective optimization based recommendation methods and hy-

Table 3.37: Results of Weighted(W) and Parameter&Weight(W-PW) techniques

Method	Precision	Ndcg	HitRate	Coverage
W-PW-R-HT-CH	0.079	0.162	0.488	0.996
W-PW-R-HT-CI	0.084	0.182	0.483	0.993
W-PW-R-HT-CFI	**0.086**	**0.186**	**0.503**	0.993
W-PW-R-HT-CFIH	0.075	0.152	0.470	**0.999**
W-PW-S-HT-CH	0.104	0.218	0.590	0.996
W-PW-S-HT-CI	0.111	0.231	0.608	0.993
W-PW-S-HT-CFI	**0.113**	**0.234**	**0.615**	0.993
W-PW-S-HT-CFIH	0.112	0.229	0.612	**0.999**

brid methods, we show in Table 3.38 the results presented in Section 3.7.3.1 (for Base settings) and in this section (the best results for Mixed(M) and Weighted(W) techniques). From the results, we observed that in general the most informative criterion of this data-set is historical check-in information. This is the same observation conveyed in Section 3.7.3.1. We observed that combining multiple criteria either using multi-objective optimization or hybridization techniques preserves Precision, Ndcg and Hitrate performance, with 0.7% difference. However, the strength of using multiple criteria is seen in the Coverage performance which is increased about 3% compared to the methods that used a single criterion. Among the hybrid methods, weighted and parameter based (W-PW-S) performed better than mixed based (M). We can conclude that weighted and parameter based (W-PW-S) hybrid methods are effective when the α_R values are decided for the data-set.

3.7.3.6 Evaluation Results: Comparison of Methods

In this section, the best performed methods and the baseline method, [34], are compared. First the configurations and then the comparison results are presented.

Configurations As already explained in the previous sections, we aimed to predict future check-ins of each user using the CheckinsJan data and we needed to set the number of neighbors, N, and the output list size, k as the parameter of the methods. In this section, we set N to 30 and k to 10, as suggested in Section 3.7.3.1. We also compared the methods by setting the k into smaller values, i.e in the range of [1-10].

Table 3.38: Comparing results of the hybridizaton methods to traditional and multi-objective optimization methods

Method	Precision	Ndcg	HitRate	Coverage
CF-C	**0.114**	**0.242**	**0.621**	0.955
CF-F	0.030	0.064	0.221	0.845
CF-I	0.033	0.067	0.226	0.873
CF-H	0.068	0.132	0.435	**0.965**
MO-CI	0.102	0.213	0.572	0.993
MO-CFI	0.103	0.213	0.577	0.993
MO-CFIH	0.105	0.218	0.596	**0.999**
MO-CH	**0.112**	**0.227**	**0.616**	0.996
M-HT-CH	**0.108**	0.226	**0.612**	0.996
M-HT-CI	0.107	**0.229**	0.601	0.993
M-HT-CFI	0.107	**0.229**	0.602	0.993
M-HT-CFIH	0.105	0.223	0.608	**0.999**
W-PW-S-HT-CH	0.104	0.218	0.590	0.996
W-PW-S-HT-CI	0.111	0.231	0.608	0.993
W-PW-S-HT-CFI	**0.113**	**0.234**	**0.615**	0.993
W-PW-S-HT-CFIH	0.112	0.229	0.612	**0.999**

Evaluation Results For the upper bound calculation, the venues are limited to the ones that are seen in the CheckinsJan data and the users are limited to the ones that check in both in January and February. The upper-bounds of the related metrics are as follows: For the Ndcg, Hitrate and Coverage metrics the upper bound is 1.0. For the Precision metric the upper-bounds are presented in Tables 3.39, 3.40 and 3.41. In Table 3.39, the precision upper bounds for non-filtered; e.g. by travel locality or time; approaches are presented. In the table, the precision upper-bound value is 1.0 when $k = 1$ and it decreases as the k increases, since not all of the users have k-many check-ins during the test period for larger k values. In Table 3.40, the precision upper bounds when we consider travel locality are presented. Similar to the previous table, the upper bound for precision increases as k decreases and reaches its maximum value when $k = 1$. In Table 3.41, the precision upper bounds when we consider time categories are presented. Similar to the previous table, the upper bound for precision increases as k decreases and reaches its maximum value when $k = 1$. For the cases of consider travel locality and time categories, we considered all the users in the test set, without limiting the target users to the ones that have at least one check-in in the given travel radius or in the given time slot. Because of this, the upper bounds cannot

Table 3.39: Precision upper-bound value for different k values

k	Upper-bound Precision
10	0.489
9	0.525
8	0.564
7	0.609
6	0.659
5	0.715
4	0.776
3	0.845
2	0.920
1	1.000

Table 3.40: Precision upper-bound value when travel locality is considered for different k values

k	r=5	r=100
10	0.155	0.342
9	0.170	0.370
8	0.187	0.401
7	0.208	0.437
6	0.233	0.477
5	0.264	0.523
4	0.304	0.577
3	0.355	0.639
2	0.423	0.712
1	0.519	0.792

reach to 1.0. For example, there are only 4117 users who have at least one check-in on a weekday night and the hit-rate upper bound is around than 0.5.

For the comparisons, we chose the best performed settings for each of the approaches and the evaluation metrics. For all of the approaches, the Base setting is used and for dynamic temporal preference based methods the averages are presented. In the following Figures, 3.28 - 3.31, the performance results of the methods are presented for different k values. Also, the best results of each method for each evaluation metric is presented in Tables 3.42 - 3.45. In the figures and the tables, the abbreviations that are introduced in the previous sections are used to represent the methods. Newly introduced abbreviations, namely Random, SVD, CCD++, Gao-H, Gao-S and Gao-SH,

Table 3.41: Precision upper-bound value when time categories are considered for different k values

k	WE_M	WE_A	WE_E	WE_N	WD_M	WD_A	WD_E	WD_N
10	0.044	0.085	0.134	0.109	0.082	0.183	0.227	0.182
9	0.049	0.093	0.148	0.121	0.090	0.198	0.246	0.198
8	0.054	0.104	0.164	0.135	0.098	0.215	0.268	0.217
7	0.061	0.118	0.184	0.152	0.108	0.236	0.294	0.239
6	0.070	0.135	0.210	0.174	0.121	0.260	0.326	0.266
5	0.082	0.157	0.242	0.202	0.136	0.290	0.365	0.299
4	0.098	0.187	0.285	0.239	0.155	0.327	0.413	0.342
3	0.121	0.231	0.342	0.288	0.181	0.376	0.474	0.397
2	0.153	0.294	0.425	0.358	0.218	0.442	0.552	0.470
1	0.212	0.390	0.545	0.459	0.277	0.543	0.662	0.573

represent the methods explained in the related work section. The Random method selects the recommended items randomly. It is executed for ten times for each k value, and the average results are presented on the figues and the tables. Both of the SVD method, proposed in [112], and CCD++, proposed in [137], are based on matrix factorization, which decompose the input matrix into sub-matrices. Both of these methods user-latent features relations (U) sub-matrix to decide on the neighbors and then use neighbors' previous preferences to make recommendations. Originally, these methods are not used on LBSNs and they are adopted to location/venue recommendation in this thesis. The Gao-H, Gao-S and Gao-SH methods are proposed in the [34] and they use historical preferences, social ties and their combinations, respectively, to make recommendations.

According to the figures, all of the methods have a similar pattern for different metrics. While for Precision and Ndcg increase in the k value decreases the performance, for Hitrate it increases the performance. This behavior is expected since as k increases the possibility of finding at least one match increases; i.e. increase in Hitrate; but some of the users have less than k-many check-ins in the test period; i.e. decrease in Precision. The Coverage performance of the methods does not change, as it is independent of the k value. According to the tables, the best performing method belongs to Gao-H [34], which uses sequence of historical preferences of the users. The use of travel locality (e.g. IL-CF-C) does not perform good, especially when a small radius

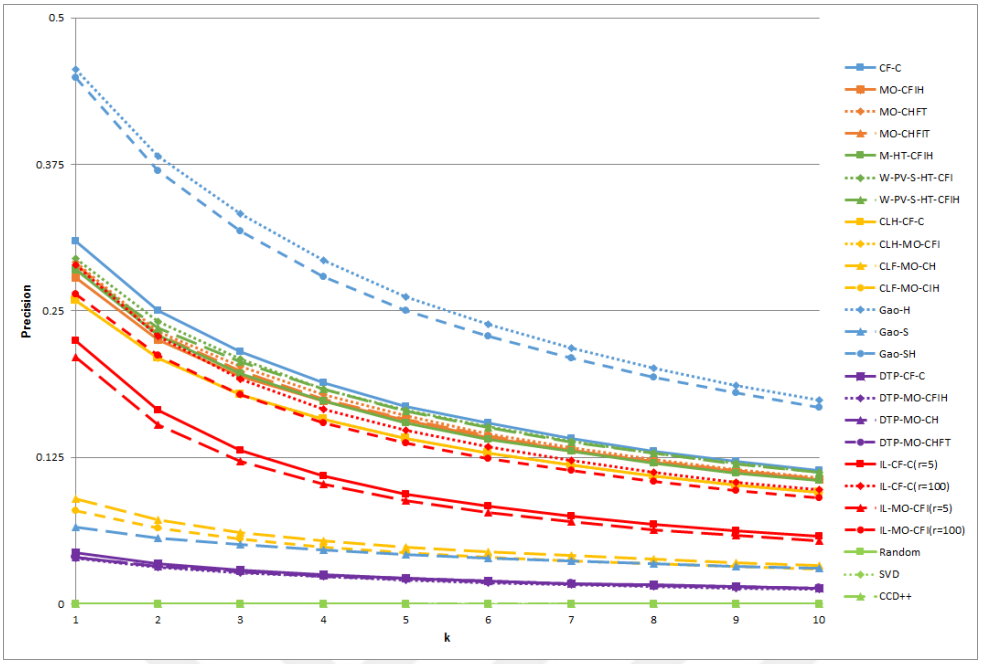


Figure 3.28: Comparison of methods according to their Precision performance

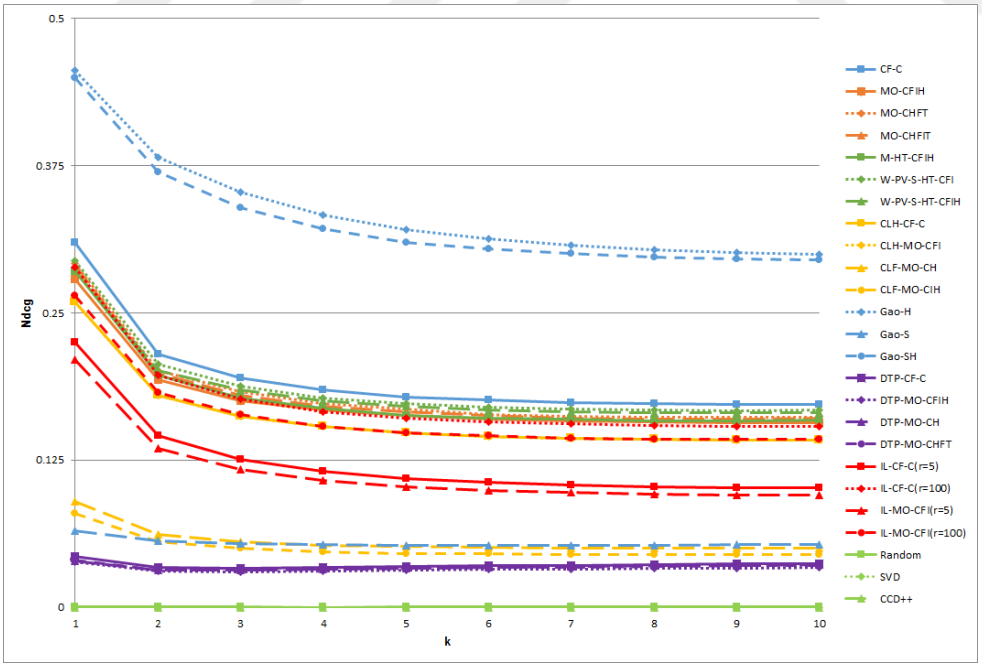


Figure 3.29: Comparison of methods according to their Ndcg performance

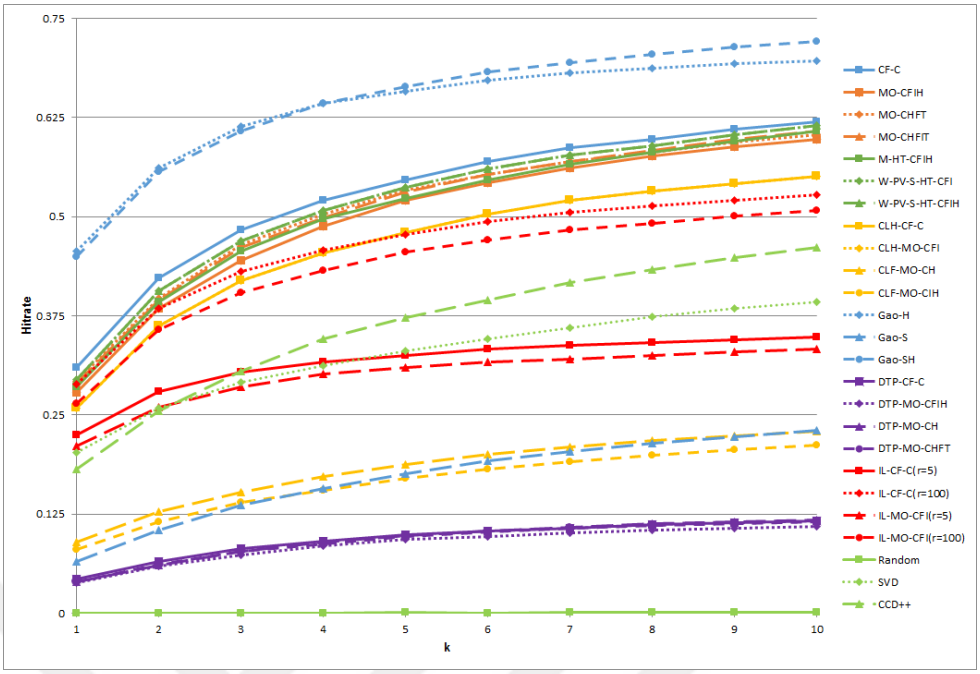


Figure 3.30: Comparison of methods according to their Hitrate performance

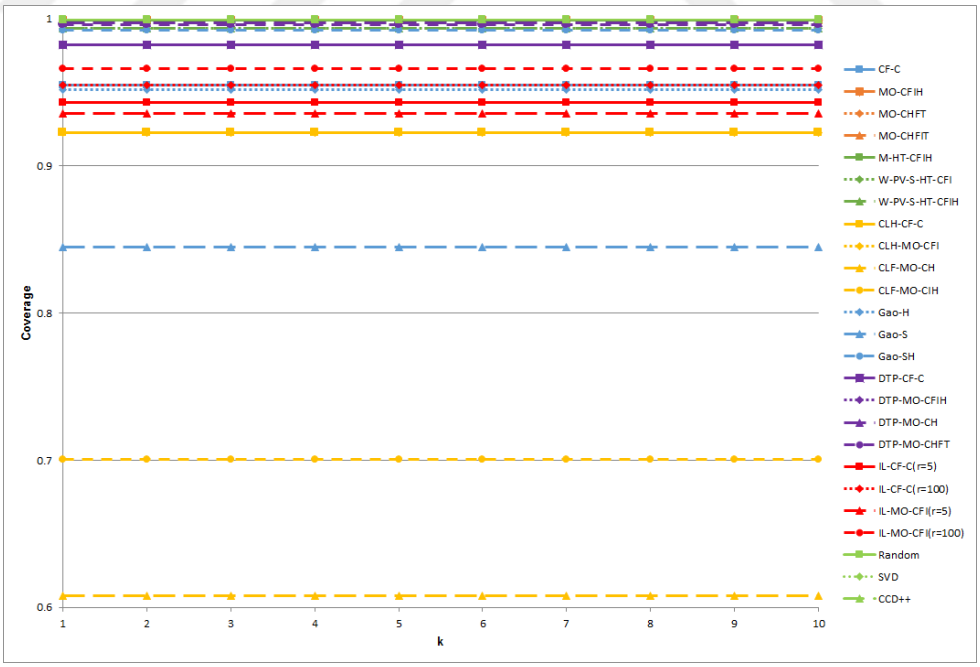


Figure 3.31: Comparison of methods according to their Coverage performance

Table 3.42: Comparison of methods (Precision)

Method	k	Precision
CF-C	1	0.310
MO-CFIH	1	0.278
MO-CHFT	1	0.292
MO-CHFIT	1	0.290
M-HT-CFIH	1	0.286
W-PV-S-HT-CFI	1	0.295
W-PV-S-HT-CFIH	1	0.286
CLH-CF-C	1	0.280
CLH-MO-CFI	1	0.259
CLF-MO-CH	1	0.090
CLF-MO-CIH	1	0.080
Random	1	0.0001
SVD	1	0.202
CCD++	1	0.182
Gao-H	1	**0.456**
Gao-S	1	0.065
Gao-SH	1	0.449
DTP-CF-C	1	0.043
DTP-MO-CFIH	1	0.038
DTP-MO-CH	1	0.039
DTP-MO-CHFT	1	0.040
IL-CF-C(r=5)	1	0.224
IL-CF-C(r=100)	1	0.289
IL-MO-CFI(r=5)	1	0.210
IL-MO-CFI(r=100)	1	0.264

Table 3.43: Comparison of methods (Ndcg)

Method	k	Ndcg
CF-C	1	0.310
MO-CFIH	1	0.278
MO-CHFT	1	0.292
MO-CHFIT	1	0.290
M-HT-CFIH	1	0.286
W-PV-S-HT-CFI	1	0.295
W-PV-S-HT-CFIH	1	0.286
CLH-CF-C	1	0.280
CLH-MO-CFI	1	0.259
CLF-MO-CH	1	0.090
CLF-MO-CIH	1	0.080
Random	1	0.0001
SVD	1	0.202
CCD++	1	0.182
Gao-H	1	**0.456**
Gao-S	1	0.065
Gao-SH	1	0.449
DTP-CF-C	1	0.043
DTP-MO-CFIH	1	0.038
DTP-MO-CH	1	0.039
DTP-MO-CHFT	1	0.040
IL-CF-C(r=5)	1	0.224
IL-CF-C(r=100)	1	0.289
IL-MO-CFI(r=5)	1	0.210
IL-MO-CFI(r=100)	1	0.264

Table 3.44: Comparison of methods (Hitrate)

Method	k	Hitrate
CF-C	10	0.620
MO-CFIH	10	0.598
MO-CHFT	10	0.603
MO-CHFIT	10	0.608
M-HT-CFIH	10	0.608
CLH-CF-C	10	0.550
CLH-MO-CFI	10	0.551
W-PV-S-HT-CFI	10	0.615
W-PV-S-HT-CFIH	10	0.615
CLF-MO-CH	10	0.230
CLF-MO-CIH	10	0.212
Random	10	0.001
SVD	10	0.392
CCD++	10	0.461
Gao-H	10	0.697
Gao-S	10	0.230
Gao-SH	10	**0.721**
DTP-CF-C	10	0.116
DTP-MO-CFIH	10	0.109
DTP-MO-CH	10	0.118
DTP-MO-CHFT	10	0.115
IL-CF-C(r=5)	10	0.348
IL-CF-C(r=100)	10	0.528
IL-MO-CFI(r=5)	10	0.333
IL-MO-CFI(r=100)	10	0.508

Table 3.45: Comparison of methods (Coverage)

Method	k	Coverage
CF-C	1	0.955
MO-CFIH	1	0.999
MO-CHFT	1	**1.000**
MO-CHFIT	1	**1.000**
M-HT-CFIH	1	0.999
W-PV-S-HT-CFI	1	0.993
W-PV-S-HT-CFIH	1	0.993
CLH-CF-C	1	0.848
CLH-MO-CFI	1	0.923
CLF-MO-CH	1	0.608
CLF-MO-CIH	1	0.701
Random	1	**1.000**
SVD	1	**1.000**
CCD++	1	**1.000**
Gao-H	1	0.952
Gao-S	1	0.845
Gao-SH	1	0.992
DTP-CF-C	1	0.982
DTP-MO-CFIH	1	0.999
DTP-MO-CH	1	0.997
DTP-MO-CHFT	1	0.996
IL-CF-C(r=5)	1	0.943
IL-CF-C(r=100)	1	0.954
IL-MO-CFI(r=5)	1	0.935
IL-MO-CFI(r=100)	1	0.966

is chosen. Use of dynamic temporal preference of users (e.g. DTP_MO-CFIH) performs poorly, which roots from the fact that use of time categories as filters makes the data sparser and more challenging to produce good recommendations. We observed that use of clustering does not improve the performance. Other methods, which use traditional collaborative filtering (e.g CF-C), multi-objective optimization based approaches (e.g. MO-CHFT) and hybrid approaches (e.g. W-PV-S-HT-CFI) perform close to each other. For Coverage metric, the multi-objective optimization based approaches (e.g. MO-CHFT) and matrix-factorization based methods perform the best. In general, use of multiple features together increases the coverage performance.

3.8 Conclusion

Recommendation systems suggest items to users by estimating their preference. Most of the recommendation systems are based on single criterion, such that they aim to evaluate the item based on overall rating. In order to give more accurate recommendations, a recommendation system can take advantage of considering multiple criteria. Location based social networks (LBSNs) are one of the resources that provide many features/criteria at once. In this chapter, we used LBSNs as a source to obtain multiple criteria and combined these criteria to make recommendations.

LBSNs provide diverse set of information about their users, such as their check-in venues, check-in time, friends and home-towns. With the help of LBSNs, it is possible to include all this into recommendation systems. In this chapter, we proposed a new multi-objective optimization based recommendation method that combines all these features, namely location, social networks, time information and dynamic preferences of the users. In this method, the criteria are used to decide on the similarity of the users and Pareto dominance idea is used to decide on the most similar users, namely neighbors. Then, the traditional collaborative filtering method is followed to recommend items; such that the neighbors' past preferences are used to make recommendations.

In this chapter, we examined several different combinations of criteria to make recommendations: Firstly, we used past preferences (check-ins), location (home-town),

social network (friendship) of the users and their multiple combinations are used to make recommendation. The evaluation results of these combinations showed that combining multiple criteria with the help of multi-objective optimization approach leads to increase in coverage while preserving precision. Secondly, we proposed a method that infers the home/center location of the users and use that information in recommendation process. Even though LBSNs give personal information of users, such as their gender, age and home-town, not all of the available features provide same amount of detail; e.g. the home-town information can be given in country, state or city level; and their truthfulness cannot be controlled; e.g. the user may indicate he/she is from a fake location. From this observation we concluded that to give better recommendations there is a need to infer fine-tuned home/center location of users and items. In the second method, the center locations of the users and the venues are calculated as longitude-latitude pairs and then this information is used to make recommendations which consider travel locality of the users. The evaluation results of this approach showed that use of smaller radius, such that making recommendation in closer distances, provides better performance in terms of precision, but it cannot provide recommendations to all of the users. In order to make recommendation to any user on the system it is better to use larger radius. Thirdly, we took into account the temporal information available in LBSNs and used it in non-dynamic and dynamic settings. In non-dynamic setting, time category, such as weekend morning, of the check-ins are used to calculate the similarity of the users. Then, the proposed multi-objective optimization based method is applied to make recommendations. In dynamic setting, the time category preferences of the users are taken into account. This way, independent of the current time the users are able to ask for venue recommendation for any time category and the system can make different recommendations for different hours of the day or different days of the week. The evaluation results showed that use of temporal information increases the performance results when it is used together with other criteria. These systems have the ability of giving recommendation to any of the users. Besides, the evaluation results revealed that when dynamicity based on temporal preference is introduced to the system, it becomes more challenging to make recommendation, since data becomes sparser.

Two main dimensions that need to be paid attention while giving recommendation

are accuracy and scalability. The accuracy is related to the fact that right recommendations should be given to the target users, such that users should use the suggested items in the future. The scalability of the system indicates that a recommendation system should give a recommendation in a short time even though it deals with millions of users and items. Motivated by these challenges, we proposed two different sets of methods to make location and social network aware recommendations. In the first set of methods, we aimed to increase accuracy. We applied several different single-criterion-based collaborative filtering methods and then combined them using several different hybridization approaches. In the second set of methods, we aimed to improve the scalability of the system and we extended the proposed multi-objective optimization based recommendation method with a clustering step. The evaluation results showed that once the weights of the single criterion based methods are decided using the data analysis or assigned by using prior knowledge on the data, the hybrid recommendation systems are as effective as the multi-objective optimization based methods. In terms of clustering, we found that clustering of users in the recommendation process leads to decrease in accuracy and coverage performance. However, it helps to deal with the scalability issue by making the methods to use only a subset of users. Clustering helps to perform calculations more efficiently in terms of time.

CHAPTER 4

MAKING RECOMMENDATIONS USING MULTIPLE SOCIAL NETWORKS

Today, many different web-based platforms, such as social networks, review web-sites, and e-commerce web-sites, use recommendation systems to serve their users. For instance, Imdb is a movie review web-site that has a service called "Recommended for you" which gives movie recommendations to its registered users. LinkedIn is a social-networking site for professionals and has a service named "Jobs You May Be Interested In" to suggest jobs to members based on their profiles. Each of these platforms captures and uses proprietary information to model users' preferences [77]. However, considering only the data captured locally will lead to a limited perspective which cannot be used to provide better service. Instead, more guided and informative recommendations are possible by forming a wider perspective by integrating data from multiple sources.

People generally use different web-platforms for different purposes. For example, even though both LinkedIn and Facebook are social networking platforms, people use mostly LinkedIn for professional connections and Facebook for personal connections [92]. Thus, combining information from various platforms can help in modeling users better [140]. To realize this, it is possible to benefit from the reported research on identity resolution which tries to connect identities of a single person across social networking platforms, e.g., [77, 92, 140, 53, 122]. Jain et al.The work described in [53] stated that the solutions to the identity resolution can be adapted by different application domains, such as security, privacy and recommendation systems. Some research efforts in recommendation systems concentrate on recommendations across

domains, e.g., [121, 148, 60]. However, these cross-domain recommendation systems focused solely on matching items and have not considered users' preferences across platforms. There are also recommendation methods that aim to combine multiple features, such as past preferences on items, social relations among users, location and temporal information, e.g., [32], [136], [135], [81]. To the best of our knowledge, even though these works use multiple features at once, none of them employs data from multiple sources. Recently, in a challenge described in [15], usage of linked data in recommendation systems is introduced as a novel strategy. In the challenge description it is stated that combining diverse information about users, items and their relations can improve recommendation performance. However, the data-set introduced in this challenge contains diverse/linked information on items, but not users. Inspired from the above-mentioned research, in this thesis, we combined data collected from multiple social networking platforms and created an integrated repository that reflects users' preferences. This may form better basis for more guided and informative recommendations. To the best of our knowledge, our work is the first to construct such kind of data repository which could be used to model users from a wider global perspective by integrating data from multiple sources [83].

For the evaluation process, we could not find an appropriate data-set because the existing data-sets used in cross-domain recommendations consist information on common items, but not on preferences or behavior of users. Further, the data-sets used in identity resolution have information on users, but not on items that they rate or interact with. To obtain information on users as well as the items they prefer, inspiring from [140], we collected information about users from the BlogCatalog website. In this website bloggers can publicly share information about their accounts in other websites. Using the shared account information, we collected publicly available information from Twitter, Flickr, Facebook, YouTube and LastFm. We anonymized the collected data for privacy concern, and then we used it for the evaluation.

The rest of this chapter is structured as follows: The collected and prepared multi-source data-set is described in Section 4.1. The employed methodology is presented in Section 4.2. Lastly, the evaluation process and the performance results are presented in the Section 4.3. The chapter is concluded in Section 4.4

4.1 Multi-source Dataset

Today, most social networking platforms and e-commerce websites provide application programming interfaces (APIs) to researchers and developers in order to collect data from these platforms. These platforms allow users to make the information about themselves publicly available, or share it only with specific users or user groups. In some of these websites, beside sharing their personal information (e.g., nickname, real name, city, age, etc.), users can also share their account addresses on other platforms.

Inspiring from [140], we referred to BlogCatalog which is a web-site where users can publicly share their accounts on other web-sites and social networks in a section called "My Communities". The shared accounts may exist on social networking platforms like Digg, Facebook, Flickr, Twitter, etc. The number of users from each social network are reported in Figure 4.1.

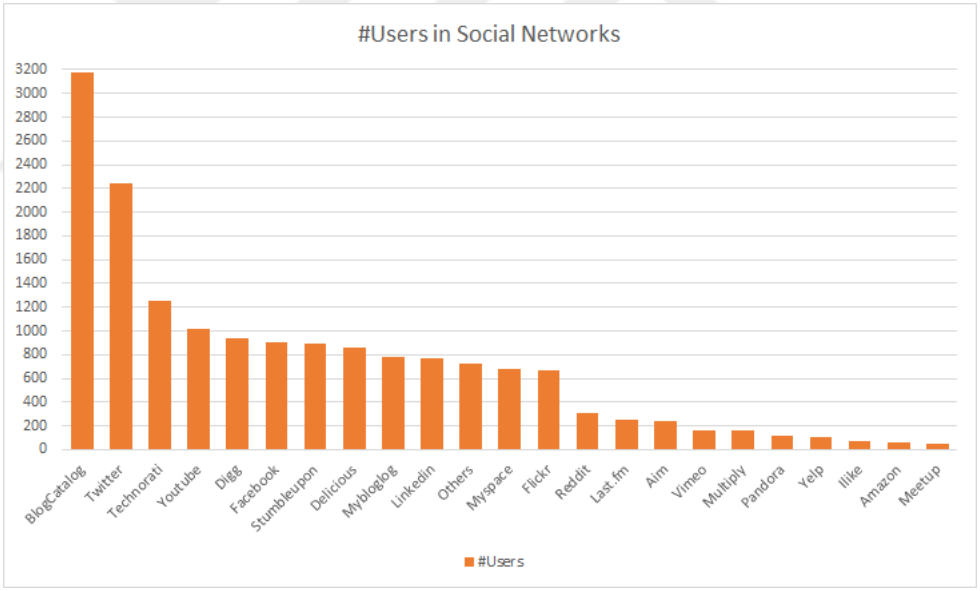


Figure 4.1: Number of users on each social network

According to Figure 4.1, there are three clusters of social networks based on the number of users: The first cluster contains social networks with the highest amount of users. From the first cluster, we collected data from BlogCatalog and Twitter. We attempted to collect data from Technorati. However, the platform stopped its previous services on May 2014 and the related data is not reachable anymore. The

second cluster contains social networks with average number of users. We collected data from Flickr, YouTube and Facebook, because of the publicly available data and it is easy to use their APIs. We attempted to collect data from other platforms too. However they either closed their services or they have very restrictive rules on data collection. For example, LinkedIn platform does not allow developers to collect most of the data without users' permission and does not let them to save any collected for future use. The last cluster contains limited number of users (i.e. less than 300). We collected data from LastFm since it is easy to use its API and most of the data is publicly available.

We mostly selected active social networking platforms, namely BlogCatalog, Twitter, Flickr, YouTube, Facebook and LastFm, where the state of the network may change over time. For example, a user may start to follow new users or may become member to new groups. The data we collected represents only the state for the date it was accessed. We collected publicly available data from BlogCatalog, Twitter and Flickr on 19-20 February 2015 and from YouTube, Facebook and LastFm on 3-8 June 2015.

To help readers to better understand the environment used in this section, we briefly explain the theme of each of the five platforms used in this study:

- BlogCatalog is a platform which allows bloggers to share information about themselves and their blogs. Also users can search for blogs of their interest and interact with other bloggers on the forums. Each blogger owns his/her own page where he/she may post a brief self-description, share own hometown, communities and accounts on other platforms. Also the platform provides information on recent visitors of bloggers' personal pages, owned blogs, followers, followees and reading list.

- Twitter is an online social networking platform which provides services to its users to share short messages (tweets) with the public, send or receive directed messages to other users and connect with other users. On their personal pages, users may post own brief information and city information. This page contains also information about written tweets, followers and followees, favorites and lists.

- Flickr allows its users to share their photographs, label them with tags, titles and descriptions. Photos shared, albums created, and photos favored can be seen on personal pages. Also, users may form groups, share their photos with others, and discuss subjects of their choice.

- Facebook is an online social networking platform which provides services to its users to create user profiles, connect to other users, exchange messages, post status updates and photos, and form groups. The publicly available information is chosen by the users themselves.

- YouTube is a video-sharing platform which provides services to users to upload, view and share videos. On the video pages, the number of views, the number of likes and dislikes and the comments are available. Registered users can create their own channels, also they can subscribe to other channels to follow new videos. Description of the channel, the uploaded videos and the play-lists created by users are available on the channels.

- LastFm is a music platform in which users have their own profile pages. The tracks users listened to, top artists and top albums they like are listed on profile pages. LastFm uses this information, and many others, to make recommendations to registered users regarding new tracks to listen to or other users to connect to. It also provides services to users to form groups and add/search events.

Recall that BlogCatalog has been used as our base platform based on the work described in [140]. On BlogCatalog users may publicly share their accounts on other social networking platforms. Using this information, we first found the mapping of user-ids, and then we collected data about these users from other platforms by using their APIs. During this process, we realized that it is not possible to collect information about all users of each platform, since some users close their accounts or do not publicly share their information. For example, on BlogCatalog 671 of 3179 users indicate that they have a Flickr account. However, we could not collect any information about 318 of them because their accounts are unreachable. We collected only publicly available information and anonymized the collected data to avoid privacy issues. Another challenge we faced is that each social networking platform structures its data

in a different way. However, since we used their APIs, we were able to easily collect the required information related to target users. Afterwards, based on users' mapping across social networking platform as obtained from BlogCatalog, we created our own structure to save the data. The last challenge is related to privacy of the collected data. We anonymized all the data and we assigned our own ids, which are unrelated to the ids assigned by the accessed websites/social networks.

As mentioned earlier, the data-set contains data from six different social networking platforms, namely BlogCatalog, Twitter, Flickr, Facebook, YouTube and LastFm. Figure 4.2 presents all features collected from these social networking platforms.

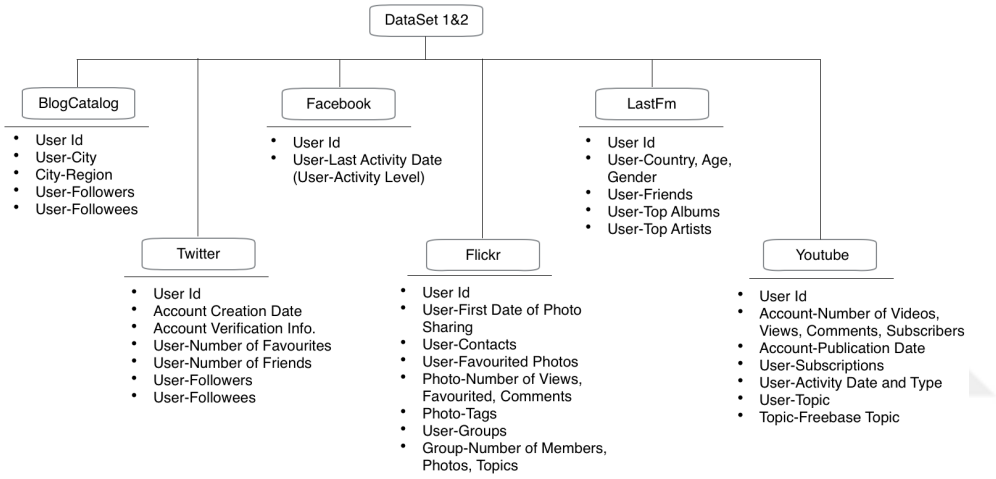


Figure 4.2: The collected data

The collected information on these platforms is as follows: From BlogCatalog, we collected (1) user-ids, (2) cities of users, (3) regions of cities, e.g., North America, Europe, etc., (4) followers of users, and (5) followees of users. From Twitter, we collected (1) user-ids, (2) account creation date, (3) account verification information, (4) number of favorites of the users, (5) number of friends, (6) followers of the users, and (7) followees of the users. From Flickr, we collected (1) user-ids, (2) first date of photo sharing, (3) contacts, (4) favored photos, (5) number of views, favorites and comments of photos, (6) tags of photos, (7) membership in groups, and (8) number of members, photos and topics of groups. From Facebook, we collected (1) user-ids and (2) account last update date. We did not collect other information since either they may cause privacy issues (e.g., user name) or the information is not publicly available.

We used last update date information as a measure to indicate user's activity level. Table 4.1 presents number of users in the given update date ranges, in terms of year or year-month, their ratio to all users and the label assigned to them.

Table 4.1: Last update dates on Facebook in the given date ranges

Date	Count	Ratio	Label
$<$ 2014 or Unknown	103	0.137	Inactive (0)
2014	161	0.214	Not very active (1)
$<$ 2015 May	242	0.323	Active (2)
2015 May or June	245	0.326	Very active (3)

From YouTube we collected: (1) user-ids, (2) video, view, comment and subscriber count of accounts, (3) account publication date, (4) user-ids of subscribed accounts (i.e., following) and subscription date to those accounts, (5) user activity date and type (i.e., Bulletin, PlayListItem, Like, Subscription or Upload) and (6) users topics. Topics on YouTube are related to Freebase topics. Consequently, we collected from Freebase the following information on topics: (1) topic ids, (2) topic names, (3) topic notable ids, and (4) topic notable names. Notable topics can be thought of as upper level representation of a topic. For example, both "Game show" and "Animated cartoon" have the same notable topic "TV genre". We further created the relationship from user topics to Freebase notable topics to make the analysis easier. We used this relationship during the conducted experiments. From LastFm, we collected (1) user-ids, (2) country, age and gender, (3) friends, (4) top albums with their ranks, and (5) top artists with their ranks.

For this study, we prepared two different data-sets that incorporate the above-mentioned data collected from six different social networking platforms. In the first data-set, firstly we limited the users to the ones which have accounts in the first three social networks; namely BlogCatalog, Twitter and Flickr. Then we expanded it with the last three social networks; namely Facebook, YouTube and LastFm. In the expansion, we considered those users and expanded the data by covering new social networking platforms. In the second data-set, we used all available users from all the six social networking platforms regardless of the level of overlap. Detailed information related to the data-sets is given next:

Dataset1: From BlogCatalog we collected information of 22291 users whose city information is known. However, only 3179 users explicitly indicate their accounts on other social networks. Even though users share their accounts publicly on BlogCatalog, we are not able to collect all of them; since some accounts are closed or private. Among BlogCatalog users, only 2187 publicly shared their Twitter accounts and only 671 publicly shared their Flickr accounts. There are 241 users who have accounts in all of the above mentioned three platforms. First, these 241 users are used to get a subset of users. Then, we added to these selected 241 users when exists their related available information collected from Facebook, YouTube and LastFm. Some information related to the selected 241 users could be expressed as follows:

- There are 241 users available in BlogCatalog data-set; they are from 66 different cities, which are located in 6 different regions. Among them 133 users have followees and 156 have followers.

- In Twitter data-set, there are 241 users available; 237 of the them have followers and 234 have followees.

- In Flickr, there are 241 users available, and 160 of them have at least one contact. Of the 241 Flickr users, 161 are members of at least one group. Total number of Flickr groups in the produced data-set is 4802. Of the 241 users, 105 have at least one favorite photo, and the total number of distinct photos favored by a member is 5067. These photos have 17611 different tags in total.

- Of the selected 241 users, 89 are available in Facebook data-set. Among these 89 Facebook users, only 15 have activity level inactive, 17 are not very active, 27 are active, and 30 are very active.

- In YouTube data-set, 118 of the selected 241 users are available. Among these users 115 of them have at least one kind of activity, only 1 of them has a subscription to other channels and only 1 of them are subscribed by other users. Of the 118 YouTube users 103 of them indicated at least one topic as their interest. Number of distinct topics is 118 and number of Freebase notable topics is 57. This indicates that even though most users have different topic tags as their interest, they mostly share a common taste.

- In LastFm data-set, 53 of the selected 241 users are available. These users indicated that they are from 17 different countries; 10 of them are females, 30 are males, and 13 did not indicate any gender. Among the selected 53 LastFm users, 37 have at least one friend and only 2 of them are friends of others. Of the 53 LastFm users, 19 indicated their top albums and top artists. There are 14818 different albums and 7846 different artists listed in the top albums and top artists lists; 13807 albums and 6297 artists are listed in top by only a single user. This indicates that the collected LastFm users have mostly unique music taste.

Dataset2: In the second data-set, instead of limiting the users to a subgroup, we used all the available users.

- There are 22291 users in BlogCatalog who are used as base in this project. The total number of names listed as users, followers or followees is 84467. The 22291 base BlogCatalog users are from 94 different cities from 6 different regions. Among these users, 6990 of them follow some other users and 8912 of them are followed by other users. The ones who have following information follow 61621 distinct users. The ones who have follower information are followed by 31476 users.

- Even though there are 2187 BlogCatalog users who indicated having Twitter accounts, we were able to collect information about only 1802 of them because some users have closed their accounts or have private accounts. 1760 of these users have at least one follower and 1738 of them have at least one followee. None of the followers (but 884 of the followees) are among the base 1802 Twitter users.

- There are 349 Flickr users who have shared their accounts on BlogCatalog and whose information is publicly accessible. Of the 349 Flickr users, 240 have at least one contact and 13 of the contacts are among the selected 349 Flickr users. 189 of these users are members of at least one group. Total number of Flickr groups which have at least one member is 7725. Of the 349 Flickr users, 150 have at least one favorite photo, the total number of distinct photos favored by a member is 7451 and these photos have 24719 distinct tags in total.

- On Facebook, we were able to collect information of 751 users. Among these users, there are 103 inactive, 161 not very active, 242 active and 245 very active users.

- There are 822 YouTube users who have shared their account on BlogCatalog and whose information is publicly accessible. 792 of these users have an activity type and only 20 of them are subscribed to a channel. Of the 822 YouTube users, 702 declared interest in at least one topic. Number of distinct topics and number of Freebase notable topics indicated of interest to these users are 523 and 193, respectively.

- On LastFm, there are 234 users who have shared their accounts on BlogCatalog and whose information is publicly accessible. These users are from 29 countries; 60 of these users are female, 115 are male and 53 haven't indicate any gender information. Of the 234 LastFm users, 168 have at least one friend; 81 of these users have at least one top artist and 78 have at least one top album; 19736 artists and 43066 albums are listed in top-artist or top-albums lists of these LastFm users.

Finally, we conjecture that the constructed data-sets can be used for several different purposes; such as tag prediction, item recommendation, link prediction, identity prediction and location prediction. For instance, the behavior of a user in a single network or multiple social networks can be used to predict his/her hometown. Further, this information can be used by researchers and practitioners working on recommendation systems, privacy and security, among other domains.

4.2 Recommendations Using Multiple Data Sources

Acquiring diverse and rich sets of features from a wide range of data-sources helps in developing alternative recommendation systems capable of serving various purposes. In this thesis, we exemplify two alternatives. Our first objective is to make recommendations to Flickr users about groups they may join in future. For this purpose, we use the first data-set introduced in Section 4.1. As a second objective, we decide

to make recommendations to BlogCatalog users regarding whom they can follow in future. For this purpose, we use the second data-set described in Section 4.1.

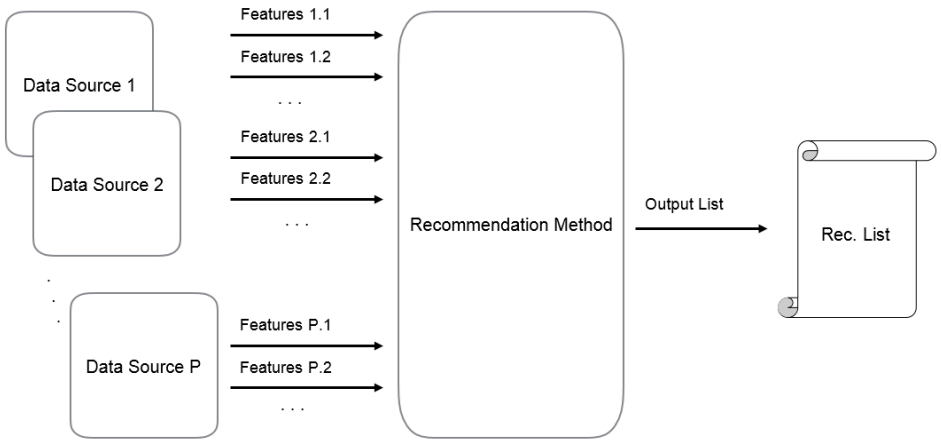


Figure 4.3: The general structure of the system

Shown in Figure4.3 is the general structure of the system which shows how recommendation methods function based on data from multiple data sources with multiple features. Data sources are labeled from 1 to P, and each source is characterized by a non-empty set of features which are used as an input to the recommendation method to produce a list of alternative recommendations. For the study described in this section, we decided to continue to use collaborative filtering, multi-objective optimization based, hybrid and social-historical model based recommendation methods to observe the effect of using data from multiple data sources.

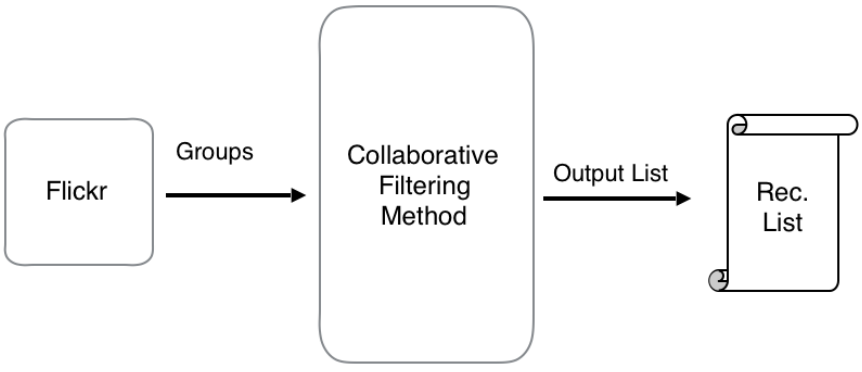


Figure 4.4: Collaborative filtering recommendation method

Collaborative filtering based recommendation: We used user-based collaborative

filtering. In user-based collaborative filtering, neighbors are decided based on similarities of users to the target user. Users who are most similar to the target user are assigned as neighbors. Then recommended items are decided by using neighbors' past preferences. In this step, different approaches to combine neighbors' preferences can be used: In voting based approach, all neighbors are considered equal and items are chosen based on number of neighbors who have chosen them. In weighted approach, neighbors can be assigned different weights, e.g., based on their similarities, and items are chosen based on number of the neighbors who have chosen items and weights of the neighbors. Other approaches are also possible. In this study, we used voting based approach. The user-based collaborative filtering method uses only a single feature from a single data-source. The structure of the methodology is illustrated in Figure 4.4, where a single data source, namely Flickr, and a single feature, namely Flickr groups, is used to make recommendations.

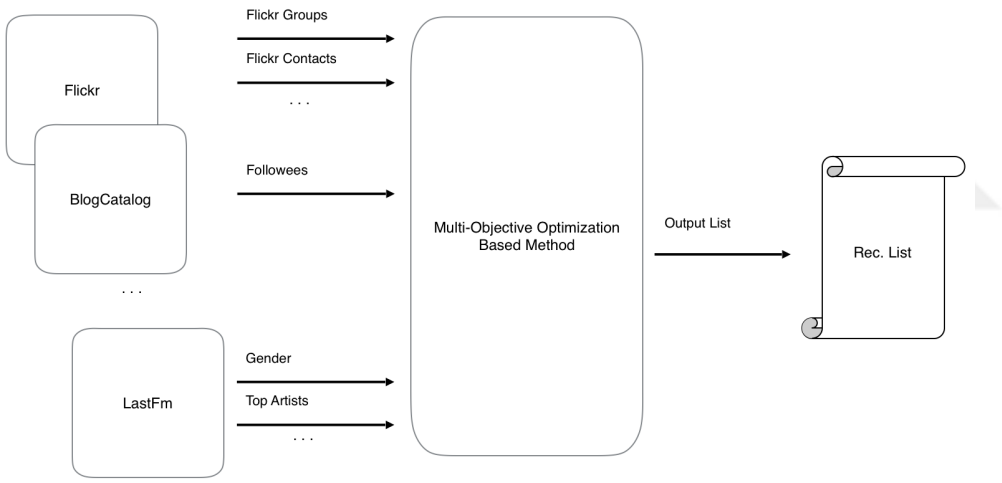


Figure 4.5: The multi-objective optimization based recommendation method

Multi-objective optimization based recommendation: We used the multi-objective optimization based method proposed in this thesis. This method determines neighbors by employing Pareto dominance method. For this purpose, similarities of users are calculated for each feature. Then based on similarities, non-dominated users are decided based on Pareto dominance results. Users who are non-dominated are assigned as neighbors. If a predefined number of neighbors should be chosen, an iterative process can be employed by removing the already selected ones from the set and

re-applying the approach on the new set. After selecting neighbors, past preferences of neighbors are used to make recommendations as it is done in collaborative filtering. In this thesis, we applied iterative Pareto dominance based process of neighbor selection and voting based item selection. The multi-objective optimization based method proposed in this thesis is capable of combining multiple features from a single data-source or from multiple data-sources at once. The overall flow of the method is depicted in Figure 4.5, where multiple data sources and multiple features are used to make recommendations. For instance, to make Flickr groups recommendations to Flickr users, we used all data from the six social networking platforms mentioned in Section 4.1. Also, any features can be used from each platform without any limitation on the number of features to be used or on the combination of features.

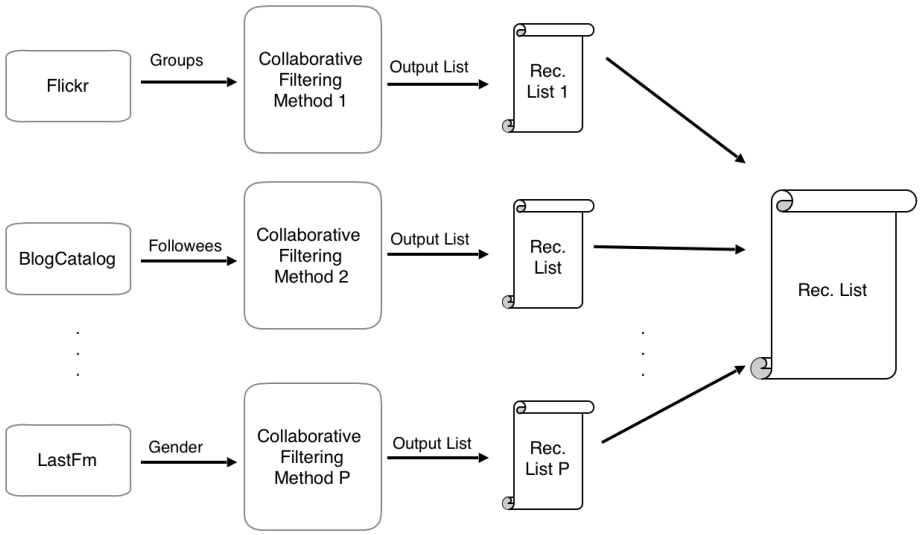


Figure 4.6: The hybrid recommendation method

Hybrid recommendation: We used an item based hybrid approach which combines the output of different recommendation methods. Even though several techniques for hybridization are explained in [11], in this study we combined only collaborative filtering methods using different features. The combination of the results and the ranking of items are decided based on the number of votes each item received. This method is capable of combining multiple features once they are decided by a single feature based collaborative filtering method. Since each recommendation method is executed separately, there is no limit on the data-source, i.e., features from single

data-source or from multiple data-sources can be used. The overall structure of the methodology is represented in Figure 4.6, where first multiple collaborative filtering based methods are used, then their output predictions are combined. Each collaborative filtering based method, labeled from 1 to P, uses only one feature from a single social network. The item based hybrid method uses predictions made by each collaborative filtering method to combine and create a prediction list.

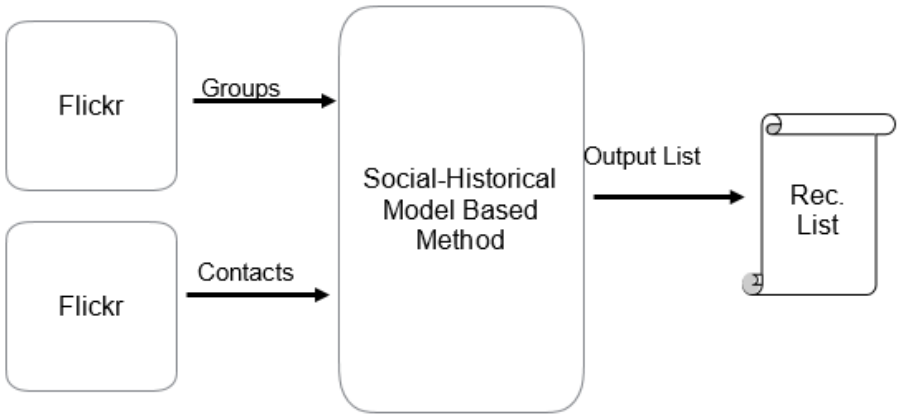


Figure 4.7: The social-historical model based recommendation method

Social-historical model based recommendation: We used the method proposed in [34], which is one of the state-of-the-art methods described in the literature. It originally aims to predict next check-in venue by combining users' past check-ins and friendship information in location based social networks. It models check-ins by a language model. The combination of friendship information is performed by a linear model. Also, it is possible to make recommendations using only past check-ins or only friendship information. The social-historical model based recommendation method [34] proposed to combine two features at once. These features do not necessarily need to be from a single data-source, i.e. features from multiple data-source can be used. The structure of the methodology is illustrated in Figure 4.7, where data from two sources are used to make recommendations. The method is able to combine up to two features. These features may be collected from different social networking platforms as well as a single one.

Even though the data-set contains features from different social networking platforms, in this section we used a subset of the features to demonstrate the effectiveness of integrating information from multiple sources. It is possible to add new features in future experiments. The features used in this thesis and their source social networking platform are explained next:

Flickr groups: In Flickr users can join different groups depending on their interests. A recommendation system can use knowledge about previously joined groups to make predictions on the groups that these users may join in the future.

Flickr contacts: In Flickr users can connect with other members based on their real-world interactions or on similar interests on the web-platform. A recommendation system can use contacts' past preferences to make predictions, as contacts most probably share similar interests.

Flickr common contacts: Similar to Flickr contacts, this feature uses past preferences of other similar users. However, for this feature similarities among users are calculated based on common contacts rather than having direct connection.

Twitter followees: In Twitter, users mostly follow other users who they already know (e.g., friends, family members, etc.) or who they like, admire or support (e.g., political leaders, singers, etc.). Having common followees may indicate that two users are similar to each other, and this can be used to make recommendations.

BlogCatalog followees: Similar to Twitter followees, users in BlogCatalog follow other users based on similar interests. Information from other users who have similar interests can be used to make recommendations to target users.

Facebook activity level: The source social networking platform of this feature is Facebook. Some users in our data-set are actively using their Facebook account, and some others are not. As explained in Section 4.1, we divided users into four groups based on their activity level. Knowledge on this feature may help to group users together by their activity on social networks. For example, more active users may join more recent groups while others tend to join well-established ones.

YouTube-Freebase topics: In YouTube users indicate their interested topics and Freebase notable topic, i.e., the upper level representation of the topic, is available. We used the combination of this information and found similarity of users on their Freebase notable topics. We used this similarity to make recommendations to target users.

LastFm gender: The source social networking platform of this feature is LastFm. Users of different genders may prefer different items to use/follow. This feature can be used for making recommendations.
LastFm top-artists: Based on mostly listened tracks, LastFm provides a list of top-artists for each user. Having common top-artists may indicate that users have similar music taste, which can be used for making recommendations.

For all the selected features, except Flickr contacts, Facebook activity level and LastFm gender, we calculated user-user similarity using Cosine Similarity measure. For Flickr contacts, similarity between the target user and his/her contacts is assigned the value 1.0, and for others the value assigned is 0.0. Similarly, for Facebook activity level is handled as follows: users at the same activity level are assigned similarity value of 1.0, and others receive 0.0. Lastly, for LastFm gender, users of same gender are assigned similarity value of 1.0, and others receive 0.0.

4.3 Evaluation of Recommendation using Multi-Source Social Networking

In this section we presented the evaluation process and results of making recommendations using multiple social networks. The data-set and the evaluation results are presented in Sections 4.3.1 and 4.3.2.

4.3.1 Multi-Source Social Networking Dataset

Information on the multi-source social networking data-set is already presented in Section 4.1. Here, we only present a summary of the data-sets description: In the data-sets publicly available information of users that have account in the BlogCatalog, Twitter, Flickr, Facebook, YouTube and LastFm websites are collected. All the collected data are anonymized to avoid privacy issues. The collected data from each website is as follows:

- BlogCatalog: (1) user-ids, (2) cities of the users, (3) regions of these cities, e.g., North America, Europe, etc., (4) followers of users, and (5) followees of users.

- Twitter: (1) user-ids, (2) account creation date, (3) account verification information, (4) number of favorites of the users, (5) number of friends, (6) followers of the users, and (7) followees of the users.

- Flickr: (1) user-ids, (2) first date of photo sharing, (3) contacts, (4) favored photos, (5) number of views, favorites and comments of photos, (6) tags of photos, (7) membership in groups, and (8) number of members, photos and topics of groups.

- Facebook: (1) user-ids (2) account last update date. The account last update date information is further used to decide on the (3) activity levels of users as explained in Section 4.1.

- YouTube: (1) user-ids, (2) video, view, comment and subscriber count of accounts, (3) account publication date, (4) user-ids of subscribed accounts (i.e., following) and subscription date to those accounts, (5) user activity date and type (i.e., Bulletin, PlayListItem, Like, Subscription or Upload) and (6) users Freebase topics: topic ids and names, topic notable ids and names; i.e. higher representation of topics. Further users' topic preferences and Freebase topics are mapped to each other to make computations easier.

- LastFm: (1) user-ids, (2) country, age and gender, (3) friends, (4) top albums with their ranks, and (5) top artists with their ranks.

To create the training and test set we applied the same approach on the two data-sets. For each data-set, we created two disjoint sets on the related feature, one for training and the other for testing. The sets are created by selecting users with at least 5 items, and we randomly selected 20% of their items from the data-set. The randomly selected items are collected as the test set and the rest are used as the training set.

For the first data-set, we set our objective as making Flickr group recommendations. In the original data-set there are 126 Flickr users where each is a member of at least one group. After creating the training and test sets, we had 126 users in the training set and 86 users in the test set. On average, these users are members of 56.008 groups for the training set and 12.628 groups in the test set.

For the second data-set, we set our objective as making BlogCatalog followee recommendations, i.e., whom to follow. In the original data-set there are 6990 BlogCatalog users who follow at least one other BlogCatalog user. After creating the training and test set, the number of users were found as 6990 and 3670, respectively. The average number of followees in the training and test sets are 39.073 and 17.304, respectively.

4.3.2 Evaluation Results

We implemented several methods with a variety of features from the collected data-sets. In Table 4.2, we present the list of methods together with the used features and their abbreviations. These abbreviations will be used in the rest of this section. Whenever multiple features from a single social networking platform are used, e.g., Flickr groups and Flickr Contacts, the method uses data from a single source, e.g., Flickr, and whenever the features are from different social networking platforms, e.g., Flickr groups and BlogCatalog followees, the method uses data from multiple sources, e.g., from Flickr and BlogCatalog.

Table 4.2: The abbreviations used in this section

Methods	Abbreviation
Collaborative filtering	CF
Multi-objective optimization	MO
Hybrid	HI
Social-historical	SH
Features	**Abbreviation**
Flickr Groups	FG
Flickr Contacts	FC
Flickr Common Contacts	FCC
Twitter followees	TF
BlogCatalog followees	BCF
Facebook activity level	FbA
YouTube-Freebase topics	YT
LastFm gender	LG
LastFm top-artists	LAr

In order to evaluate the performance of the methods we used Precision@k, Recall@k

and F1-measure. We explained precision previously in Section 3.7. In this section, we only explain recall and F1-measure.

- **The Recall@k** metric is presented in Equation 4.1. In the equation tp_k represent true positives and fn_k represents false negatives in the given output list with size k. False negatives are the ones that are not listed in the output list but are actually should be on the list, such that the venue visited in the future by the target user is not presented on the output list or the actually connected genes are not predicted. While giving the evaluation results, we presented the average of the $Recall_k$ values.

$$Recall_k = \frac{tp_k}{tp_k + fn_k} \tag{4.1}$$

- **The F1-Measure** is presented in Equation 4.2. It is the harmonic mean of $Prec_k$ and $Recall_k$ values.

$$F1 - measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.2}$$

Details of the evaluation results on our two objectives, namely recommendations of Flickr groups and recommendations of BlogCatalog followees, are given in Sections 4.3.2.1 and 4.3.2.2, respectively.

4.3.2.1 Recommendation of Flickr groups

Our first example recommendation system aims to recommend Flickr groups to users. For the experiments, we need to assign neighbors count (N) and the output list size (k), which can affect the performance of the methods. We first started with some arbitrary values of these parameters and decided on the method that performs the best. Afterwards, we decided on the best values of N and k using only the selected method. To be fair to the other methods, lastly, we performed the analysis on all the methods using the determined N and k values.

First, we assigned N and k to 5. We did not want to assign a larger value to N because we only have 126 users. We assigned to k the value 5 based on experi-

ence because we observed in our daily life that most recommendation systems prefer to present a small number of items as recommendations to their users. Evaluation results of the methods with the assigned values of N and k are shown in Figure 4.8, where the method and features combination are reflected in the form $Method - Feature1\_Feature2\_Feature3$. For example, the combination $HI - FG\_TF\_BCF\_LG$ refers to the Hybrid method combined with the features Flickr groups, Twitter followees, BlogCatalog followees and LastFm gender. Also, hit-rate has been scaled to its 10% in order to have a better representation together with the other metrics.

According to Figure 4.8, using data from the same social networking platform and same features do not necessarily lead to better performance. For example, CF_FCC (collaborative filtering using Flickr common contacts) performs better than CF_FG (collaborative filtering using Flickr groups). Methods that use information from a different social networking platform only, such as CF_BCF or CF_LG, don't perform better than methods that use information from the target social networking platform. This may be related to the fact that people use different web-platforms for different purposes [92] and may behave differently in different social networks. Using the Social-Historical model [34] did not show good performance when the output list size is set to 5. Using multiple features at once (HI and MO methods) performs better than using a single feature (CF methods). Hybridization of item recommendations (HI methods) showed best performance when Flickr groups, Flickr common contacts and BlogCatalog followees are used altogether. Our multi-objective optimization based recommendation method performed the best when Flickr groups, Flickr common contacts or when additionally Twitter followees are used together. Other metrics, namely recall, F1-measure and hit-rate, follow similar patterns to the precision results. According to these results, the best performing method is $HI - FG\_FCC\_BCF$; it is actually used to decide on the best values of the two parameters N and k.

We decided on the best value of N by considering the range [1,62] with 1 increment. We stopped at 62 neighbors at most since this is half of the candidate neighbors. We used the method $HI - FG\_FCC\_BCF$ only because it is the best performing method we found in the previous experiment. We kept the value of k set to 5 as in the previous experiment. The evaluation results for different N values are given

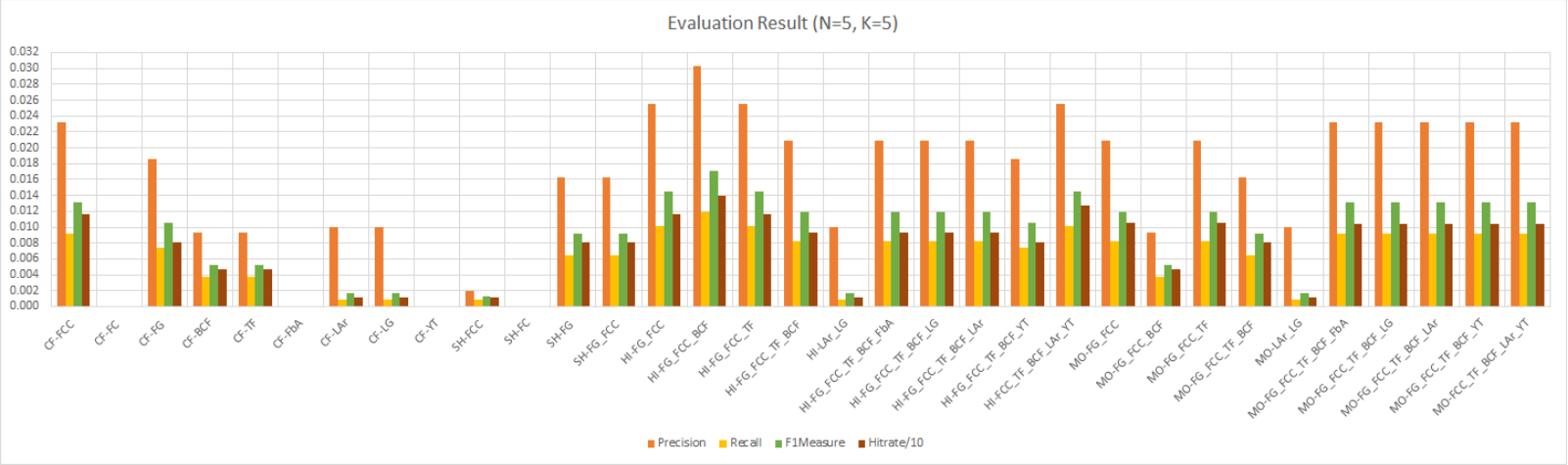


Figure 4.8: Evaluation results for N=5 and k=5 (Flickr groups)

in Figure 4.9 which shows that the performance of the algorithm deviates based on neighbors count. If the method uses only few neighbors, it cannot make good recommendations. Similarly, assigning many users as neighbors produces poor performance because they may introduce noise to the methods. Besides depending on the method, choosing many neighbors may cause time and memory problems. For example, in collaborative filtering based algorithms the candidate items are selected by looking at the historical preferences of each neighbor. If there are many neighbors, the number of candidate items will be high, and this will lead to longer computations. Interestingly, the hit-rate performance remains nearly the same for most of the neighbors count, except few spikes. This may indicate that independent of neighbors count, predicting at least one true recommendation is possible, however a tuned number of neighbors can help to increase the number of true predictions for each user. Based on Figure 4.9, 16 has been determined as the best performing N value for this data-set; this value actually provides the best precision, recall and F1-measure performance.

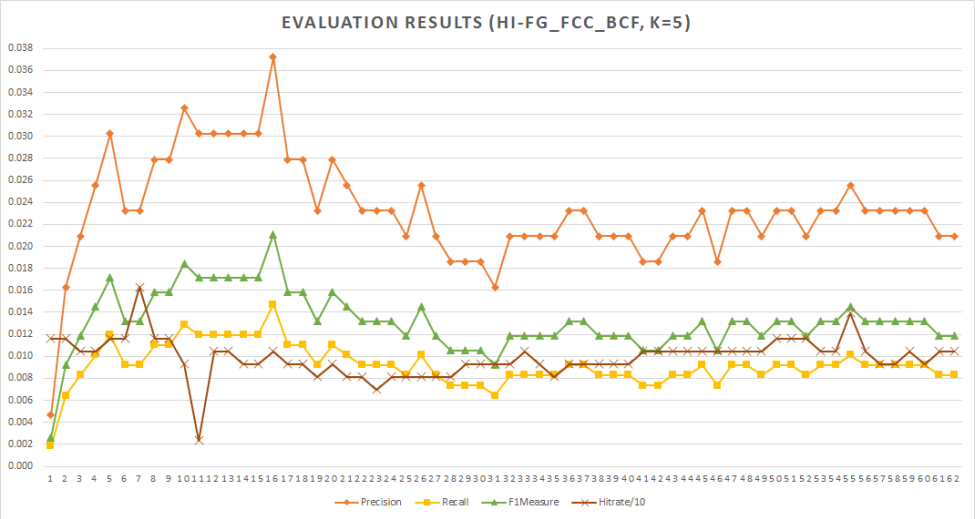


Figure 4.9: Evaluation results for different values of N

After deciding on the value of N, the next step is to decide on the value of k, the output list size. Using the method $HI - FG\_FCC\_BCF$ and setting N to 16, we searched for the best value of k in the range [1,30] with increments of 1. We set the upper bound as 30 items because it is usual for recommendation platforms to present smaller number of recommendations. Also, it is known that in search engines usually one outcome is composed of 15 results and users tend to select the results in the first few reported pages. The evaluation results for the best k value are presented in Figure

4.10. The results show that for different measures, different values of k perform better. The best value of k for precision is 4, it is 27 or 28 for recall, and it is 12 for F1-measure. As expected, when the value of k increases, precision decreases but recall increases, since larger number of recommendations are presented. The hit-rate performance follows a pattern similar to recall, since it is expected that both of these measures perform better as k increases.

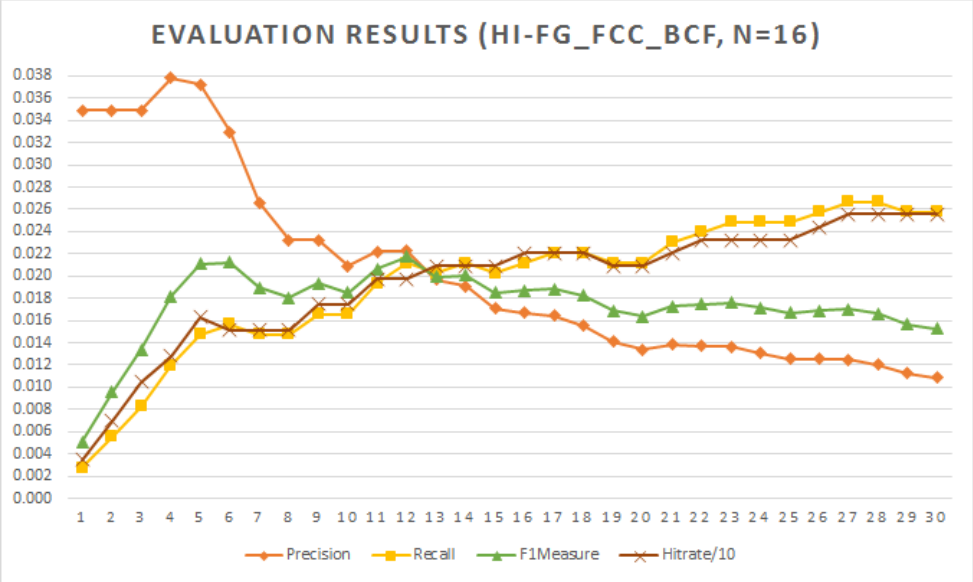


Figure 4.10: Evaluation results for different values of k

Lastly, we performed the analysis on all methods with the selected values of parameters N and k. We discarded Flickr contacts feature since it does not provide any successful recommendations as reflected in Figure 4.8. The evaluation results of the methods when N is set to 16 and k is set to either 4, 27 or 12 are presented in Figures 4.11, 4.12 and 4.13. When $k = 4$, i.e., the value that produces the highest precision in the previous experiment, collaborative filtering that uses the target feature, namely Flickr groups, and hybridization of item based methods (HI methods) perform best. As observed in the previous work, we observe that change of neighbor and output list size affect the performance of the methods, and hence they should be tuned carefully. Mostly, recall, F1-measure and hit-rate results follow patterns similar to precision results while using different methods and features. These results confirm that using data from multiple sources (e.g., social networks) improves recommendation performance. When $k = 27$, i.e., the value that produces the highest recall in the previous experiment, generally precision performance of the methods decreases

while recall performance increases. This is the expected behavior, since the increased output list size makes it easier to list true recommendations on the output, however it also leads to include more false recommendations in the output list. According to Figure 4.12, the best performing methods belong to the social-historical model. However unlike other methods and the results of previous experiments, there is a larger gap between hit-rate and recall performance. This may indicate that these methods are able to give better recommendations for certain users as k increases. Besides, the social-historical model based method and methods that use multiple features at once (HI and MO methods) perform equally good to or better than methods that use single feature (CF methods). When $k = 12$, i.e., the value that produces the highest F1-measure in the previous experiment, results similar to the previous experiment ($k = 27$) are observed. There is a decrease in precision and increase in recall and F1-measure compared to the case $k = 4$. As explained previously, longer output list leads to higher recall and F1-measure.

Even though we performed experiments that favor each of the metrics, i.e., precision, recall, and F1-measure, it is more important to make true recommendations to as many users as possible, i.e., higher hit-rate; and to present in the limited length output list one or more items that the user is expected to use in the future, i.e., higher precision. Based on our analysis, for shorter output lists and when precision and hit-rate are considered more important, the best performing method is the hybridization method which combines information of multiple features from multiple social networking platforms. This way, the method can model its users with other aspects which are not obvious for a single social networking platform.

4.3.2.2 Recommendation of BlogCatalog followees

Our second example recommendation system suggests new links, i.e., new followees, to BlogCatalog users. Similar to the experiments performed in the previous section, we started with a guess on values of parameters N and k and found the best performing method, then we decided on best N and k values using the selected method, and lastly we conducted experiments on all methods with the decided N and k values.

First, we assigned N and k to 5, as we did in the previous experiment. Evaluation

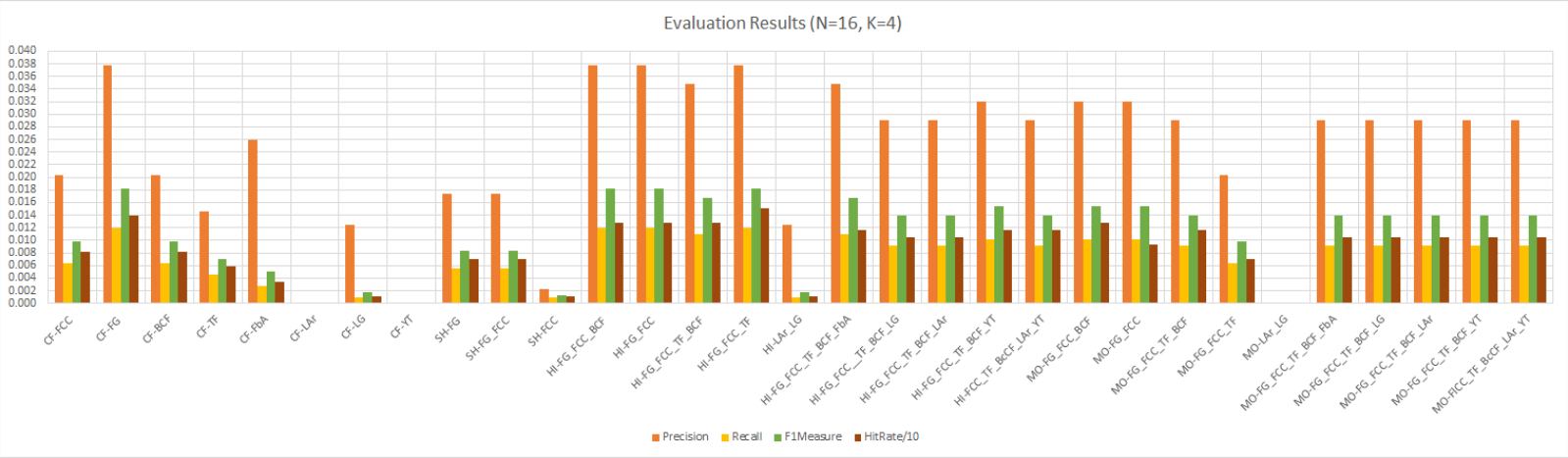


Figure 4.11: Evaluation results for N=16 and k=4 (Flickr groups)

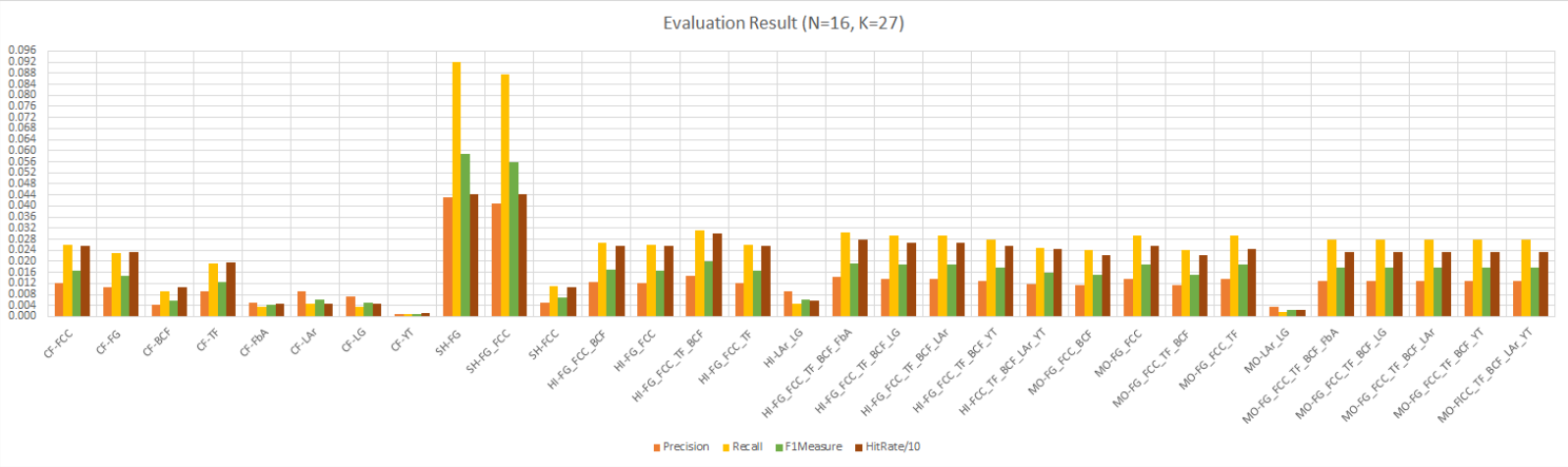


Figure 4.12: Evaluation results for N=16 and k=27 (Flickr groups)

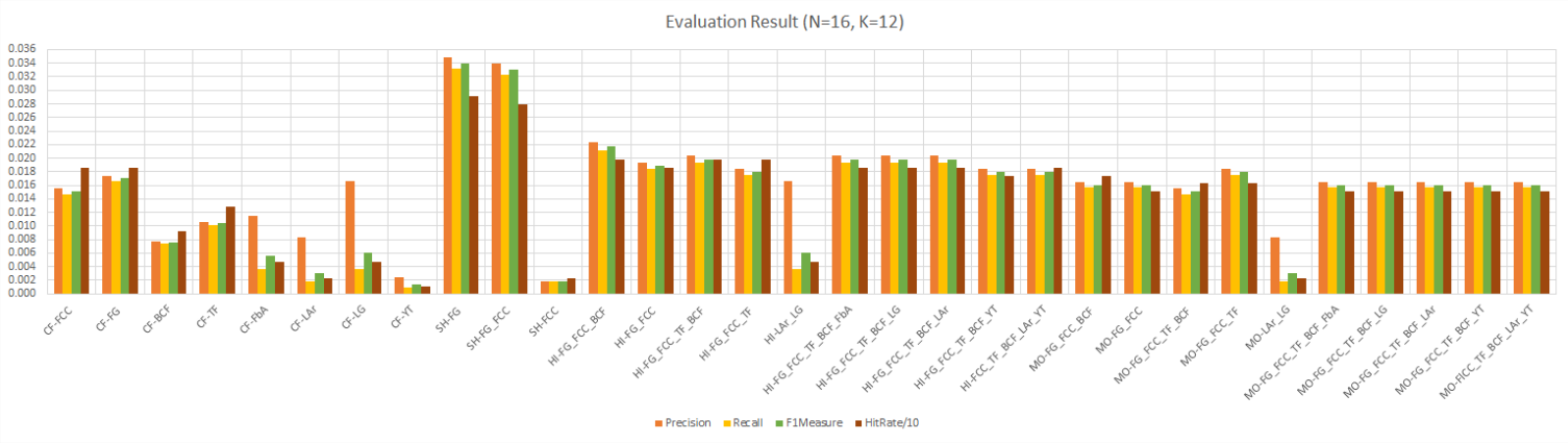


Figure 4.13: Evaluation results for N=16 and k=12 (Flickr groups)

results of the methods with our initial assigned values of N and k are shown in Figure 4.14, with the same naming pattern, i.e., $Method{-}Feature1\_Feature2\_Feature3$. Similar to the previous figures, hit-rate has been scaled to its 10% in order to have a better representation of the other metrics.

According to Figure 4.14, the best performing methods are collaborative filtering method that use the target feature, i.e., CF_BCF, and methods that use combination of features from multiple social networking platforms. We observed that HI-FG_FCC_TF_BCF_LG, which is based on hybrid itemization method using Flickr groups, Flickr common contacts, Twitter followings, BlogCatalog followings and LastFm gender features performs slightly better than other methods. For instance, it performs 0.1% better than others in terms of recall, F1-measure and hit-rate. We chose this method to decide on the best N and k values.

Using the HI-FG_FCC_TF_BCF_LG method, first we decided on parameter N by setting its value in the range [1,50] with 1 increment. Even though there are 6990 users, we stopped neighbor count at 50. During the experiments we didn't use any parallelization technique, and higher N value required more resources. In the future we will move our implementation to a parallelized implementation to observe the performance with higher N values. Results from the experiment are shown in Figure 4.15. According to the figure, after increasing N from 1 to 3-4 there is a balance on performance up to $N = 36$. After $N = 36$, there is a small increase in performance of the evaluation metrics. However, it is not obviously seen, that hit-rate ratio increases as N increases. For example, when $N = 1$ hit-rate is 0.0000012339, it is 0.0000022346 when $N = 36$, and it is 0.0000025742 when $N = 50$. Even though these values are really small, we preferred to set N to 50.

After deciding on $N = 50$, the next step is to decide on k value. We conducted experiments using HI-FG_FCC_TF_BCF_LG method and by setting k values in the range [1,30] with 1 increment. We limited maximum output list size to 30, since in real life most recommendation systems present shorter lists, such as 10 or 15 elements in a page at most. Results of the experiment are shown on Figure 4.16. According to the figure, precision performance increases up to $k = 11$, then it stays in balance, and finally starts to decrease after k is around 16. Performance of recall, F1-measure and

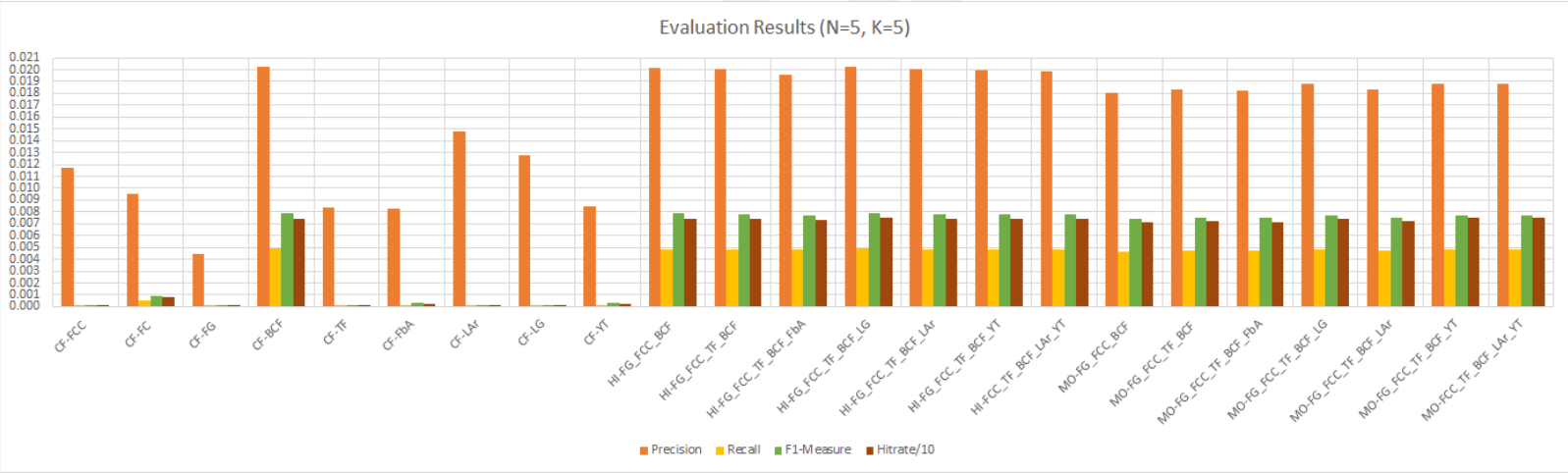


Figure 4.14: Evaluation results for N=5 and k=5 (BlogCatalog followees)

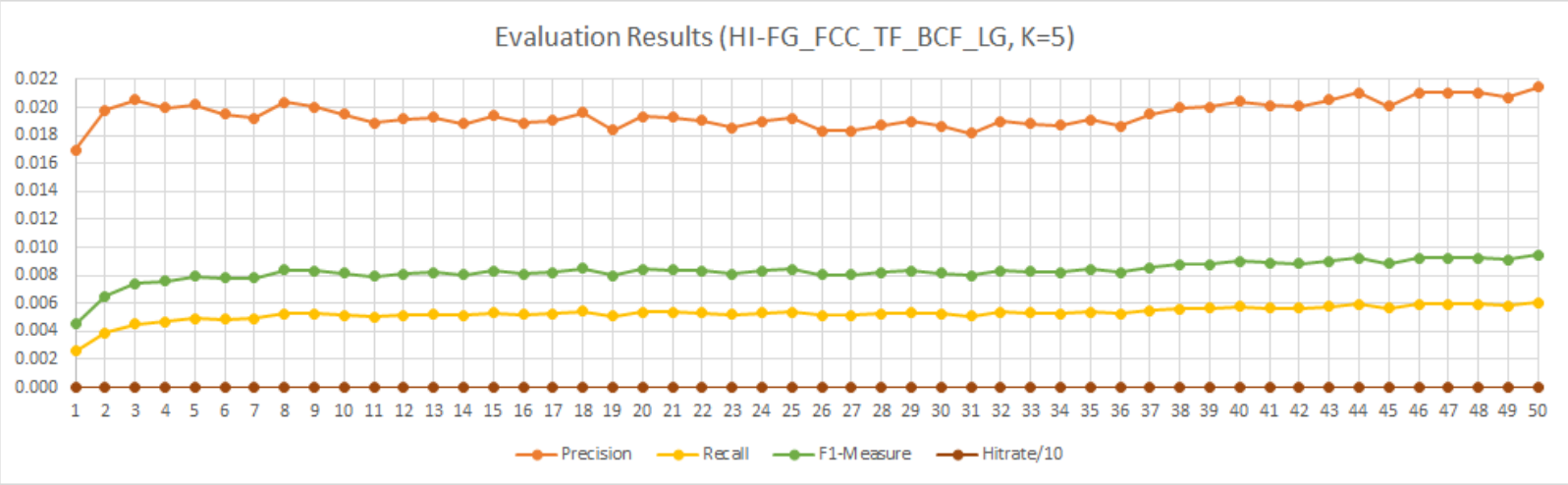


Figure 4.15: Evaluation results for HI-FG_FCC_TF_BCF_LG method and k=5 (BlogCatalog followees)

hit-rate increase as k increases. As a result of these observations, we decided to set k to 15, which is one of the values that provide best precision performance, and to 30 which provides best recall, F1-measure and hit-rate performance.

We conducted experiments on all methods using the parameters $N = 50$, $k = 15$ or $k = 30$. The results are presented in Figures 4.17 and 4.18, respectively. The figures show that the best performing methods are collaborative filtering which uses BlogCatalog followee feature, i.e., CF_BCF, and methods that use combination of features from multiple social networking platforms. Multi-objective optimization based methods perform slightly better than others in terms of hit-rate when $k = 30$.

From the previous experiments, we observed that collaborative filtering method that use the BlogCatalog followee feature, i.e., CF_BCF, and hybridization of items methods (HI methods) and multi-objective optimization based methods perform equally well. We further analyzed their performance on different types of users. For this purpose, we selected CF_BCF, HI-FG_FCC_BCF and MO-FG_FCC_BCF methods. We compared these methods for $N = 50$ and $k = 15$; similar analysis can be easily performed for other settings too. All these methods have nearly the same overall precision performance and the latter two of them use features from multiple social networking platforms. Analysis results are reported in Tables 4.3 and 4.4.

Table 4.3: Comparison of methods (CF_BCF vs. HI-FG_FCC_BCF)

Analysis	Avg. upper bound (test set)	Avg. no. of followees on the train test
Perform equally well	0.447	73.771
Perform equally well (At least one true rec.)	0.666	178.875
Perform equally well (No true rec.)	0.359	31.343
CF_BCF performs better	0.850	66.000
HI-FG_FCC_BCF performs better	0.400	21.000

According to the results, collaborative filtering and hybrid method perform equally well on users whose precision is around 0.445; this means they can perform equally good for users who have on average 6.7 followees in the test set (Found by multi-

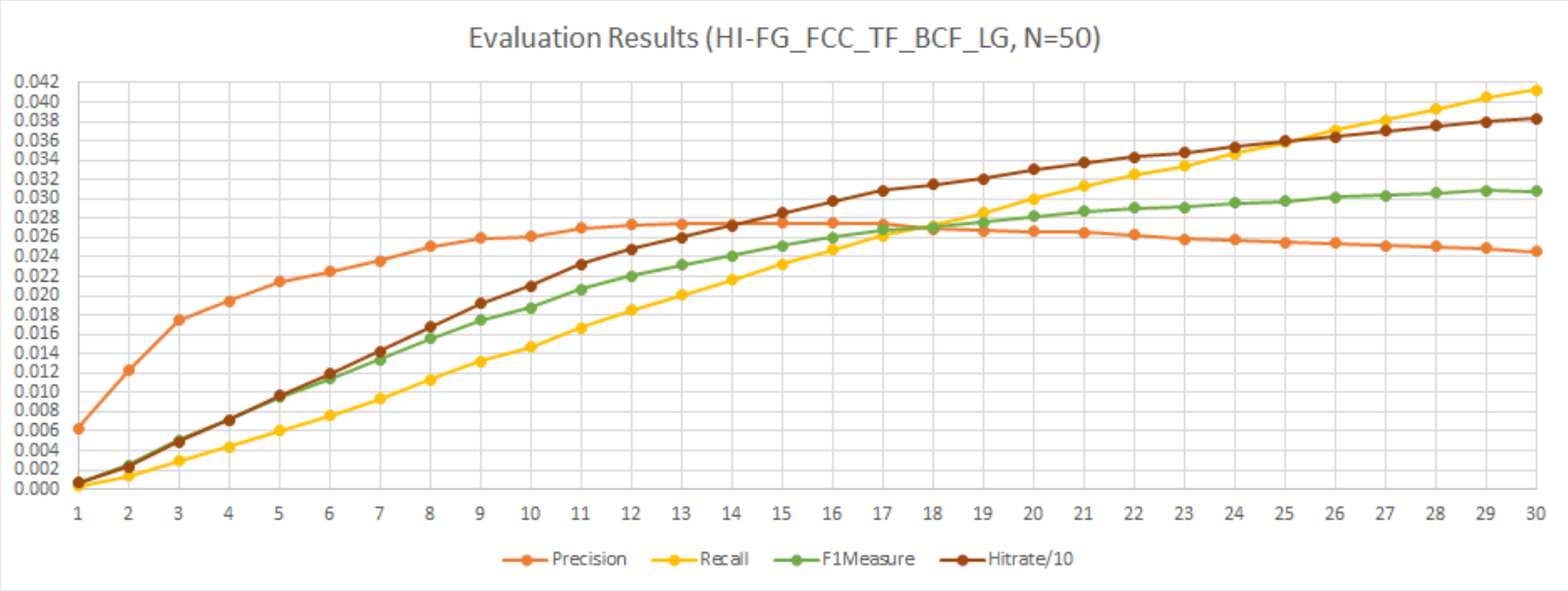


Figure 4.16: Evaluation results for HI-FG_FCC_TF_BCF_LG method and N=50 (BlogCatalog followees)

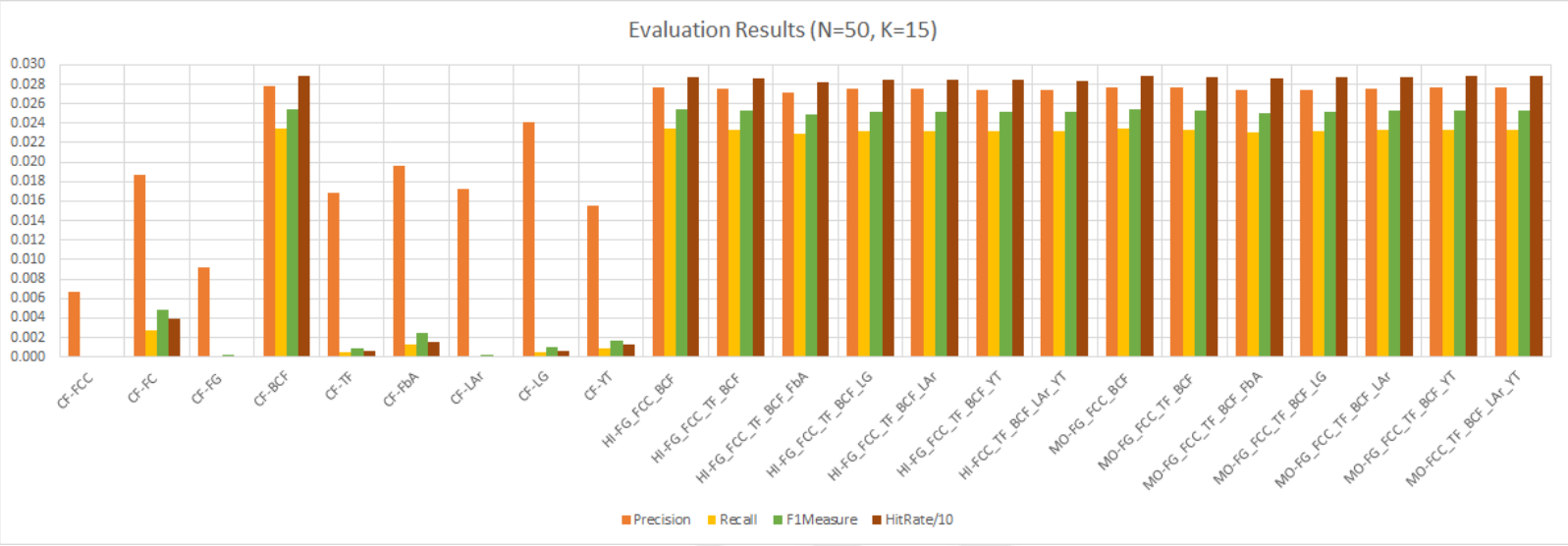


Figure 4.17: Evaluation results for N=50 and k=15 (BlogCatalog followees)

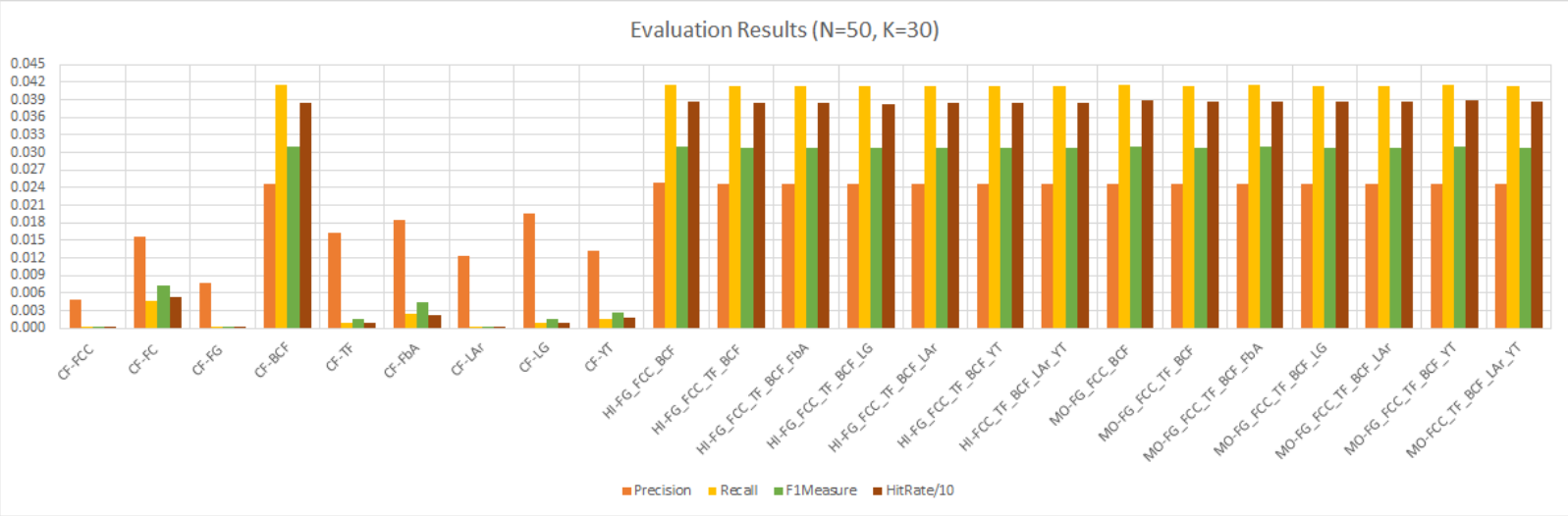


Figure 4.18: Evaluation results for N=50 and k=30 (BlogCatalog followees)

Table 4.4: Comparison of methods (CF_BCF vs. MO-FG_FCC_BCF)

Analysis	Avg. upper bound (test set)	Avg. no. of followees on the train test
Perform equally well	0.439	39.813
Perform equally well (At least one true rec.)	0.663	105.865
Perform equally well (No true rec.)	0.355	32.319
CF_BCF performs better	0.717	205.292
MO-FG_FCC_BCF performs better	0.562	33.119

plying k by average precision, i.e., 15 * 0.445 = 6.7). Also, these methods perform equally well when there are about 74 followees information in the train set. Similarly, comparison of collaborative filtering and multi-objective optimization based method reveals equal performance when there are about 6.6 followees in the test set and around 40 followees in the train set. When we further analyzed the case of equal performance, we observed that all three methods are unable to make true recommendations for users who have on average 5.3 followees in the test set and about 32 followees on average in the train set. This indicates that these methods perform better for users with more followees in the training and test sets. When we looked at the cases where CF_BCF performs better, we observed that this method performs better for users with 10-12 followees in the test set. On the other hand, both hybrid (HI) and multi-objective optimization based (MO) methods are better than collaborative filtering based method for users with less number of followees in the test set. Also they are able to model the target user better than collaborative filtering based method when there are less number of followees information in the train set. For example, MO-FG_FCC_BCF performs better than CF_BCF when there are on average 33 followees in the train set. The analysis shows that using multiple features from multiple social networking platforms helps the system to model users more effectively, especially for users with less information.

4.4 Conclusion

Beside combining multiple criteria from a single data source, we aimed to combine multiple criteria from multiple data sources. Restricting the analysis only to a single source may miss some vital information about the users and it is more beneficial and rewarding to consider integrated information from multiple sources to have more complete information about each user.

In this chapter, we integrated information collected from multiple different social networking platforms to create an integrated model of individuals and to make recommendations to them. For this purpose, we collected data that contains information collected from BlogCatalog, Twitter, Flickr, Facebook, YouTube and LastFm web-sites and created two different data-sets. We used the created data-sets to make recommendations to target users on different platforms, i.e., recommending to Flickr users new groups to follow, and recommending to BlogCatalog users other users to follow. We implemented several different types of recommendation methodologies to observe their performance. These methods include collaborative filtering, multi-objective optimization based recommendation, hybrid and social-historical model based recommendation methods.

We compared the performance of these recommendation methodologies while using single versus multiple features from a single versus multiple sources. The conducted experiments showed that using multiple features from multiple sources improved the recommendation performance.

CHAPTER 5

INFERRING GENE REGULATORY NETWORKS VIA MULTIPLE DATA SOURCES AND MULTI-OBJECTIVE OPTIMIZATION

Network modeling has numerous applications in different branches of science. For example, in sociology researchers study on friendship networks, in information technology researchers work on world wide web, which is a network of web pages, or in biology scientists study on interactions among genes and proteins [128]. The structure of the network can reveal various information which are unknown or hard to experiment in a laboratory. For example, in biology it is very costly to directly observe the gene relationships by experiments [128]. However, it is easier to measure gene expression levels, which can be used to computationally decide on the connections among genes.

In biology, there are numerous available data on DNA, RNA, proteins and metabolites interactions, which can be used to infer interactions among these biological components [108]. Gene regulatory networks (GRNs) are composed of these components and their interactions. The most common approaches to infer (reverse engineer) GRNs are based on Boolean networks, Bayesian networks, relevance networks, differential and difference equations [108]. Recently, integration of prior knowledge to this process is presented to the literature ([108, 91, 120, 46]).

The methods that aim to infer GRN should consider the properties of GRNs, which are sparseness, scale-free topology, modularity and structurality of inferred networks [46]. The GRNs are sparse, such that the number of connection among genes are

limited. Some genes regulate many other genes while some others regulate only few or no other genes, which is related to the scale-free topology feature. It is shown that GRNs follow the power distribution function of the connectivity ([94, 147]). In [46], it is shown that the GRNs are structurally decomposable into network motifs. The modularity feature of GRNs indicates that there are clusters of genes which are highly co-expressed and/or have similar functions [108]. We observed that the GRNs and the recommendation systems have similar features. For example, both of them are sparse and have a topology that usually follows power distribution function ([136, 34, 7]). Also, recommendation systems usually contain clusters of users and/or items which is mostly used to predict the future preferences of the users by the recommendation methods.

In the following sections, the employed methodology is presented in Section 5.1 and the evaluation process and the results are presented in Section 5.2. The chapter is concluded in Section 5.3.

5.1 Pareto dominance and collaborative filtering based prediction

Observing the similarities between GRNs and recommendation systems and with the purpose of constructing GRNs using information from multiple data-sets, we applied the proposed multi-objective optimization based method explained in Section 3.1 to infer the GRNs. In the original approach, target users are recommended with the items, which are predicted to be preferred by the user in the future. In this section, we mapped the target users into genes and the output is mapped to the predicted genes that the target user interacts (regulates). We decided to use this method since it is able to combine information from multiple features from multiple data-sets. The method is composed of 3 main steps: Similarity calculation, neighbor selection and regulated genes (item) selection.

- Similarity calculation: Similarity among genes are calculated using the features available in data-set(s). Features don't have to exist in a single data-set and features from multiple data-sets can be used. In the literature there are various similarity or correlation calculation methodologies, such as Euclidean distance,

Pearson correlation and Cosine similarity. In this section we preferred to use Cosine similarity, the Equation 5.1.

In the Equation 5.1, the genes are shown as *A* and *B*. Genes can have multiple features and each feature is indicated by the subscript *i*. The subscript *j* indicates the values of each feature. For example, assume that A and B represent the genes ACE2 and ASH1, respectively. In the Spellman's data-set [119], these genes are represented by 77 different measures. These measures are further divided into three phases by the [9]. Each of these phases contains different number and kind of measures. So, ACE2 and ASH1 are represented by three phases, which can be thought to be features (*i* in the equation) and these phases contains a list of (vector of) values (*j* in the equation).

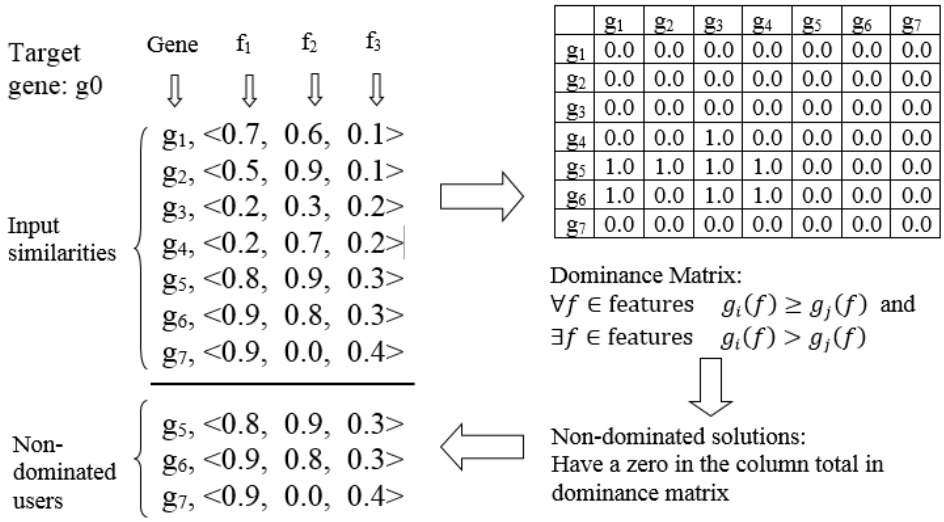
$$sim(A_i, B_i) = \frac{\sum_{j=1}^{n} A_{ij} \times B_{ij}}{\sum_{j=1}^{n} A_{ij}^2 \times \sum_{j=1}^{n} B_{ij}^2} \tag{5.1}$$

- Neighbor selection: The neighbors are the ones that behaves most similar to the target gene. Knowing these genes and their connections in the graph can be used to predict the connections of the target gene. In order to decide the most representative neighbors, the similarity values calculated in the previous step and Pareto dominance relation are used (Equation 5.2). In the equation g_i and g_j represent genes and f indicates the different features. According to the equation, if the gene g_i has at least one higher similarity value and no lower similarity values than the gene g_j, then the gene g_i dominates the gene g_j. At the end, the non-dominated genes are assigned as the neighbors of the target gene. Note that this equation and the Equation 3.5 are same, except the meaning of the parameters.

$$dom(g_i, g_j) = \begin{cases} 1.0 & \forall f\ g_i(f) \geq g_j(f) \text{ and} \\ & \exists f\ g_i(f) > g_j(f) \\ 0.0 & \text{otherwise} \end{cases} \tag{5.2}$$

For example, in Figure 5.1 a multi-dimensional data is given. This example figure is similar to Figure 3.1, except in this figure we use genes instead of users.

In the example given in Figure 5.1, similarities of seven genes to the target gene are given. The f_i values are the calculated similarities in the previous step for three different features. After constructing the dominance matrix using the Equation 5.2, the non-dominated genes are decided. The genes whose column sum equals to 0.0 are the non-dominated genes.



	g1	g2	g3	g4	g5	g6	g7
g1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
g2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
g3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
g4	0.0	0.0	1.0	0.0	0.0	0.0	0.0
g5	1.0	1.0	1.0	1.0	0.0	0.0	0.0
g6	1.0	0.0	1.0	1.0	0.0	0.0	0.0
g7	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 5.1: Example input and non-dominated solutions for prediction of GRN

In order to collect as many neighbors as predefined, an iterative process of neighbor collection is applied. First, we apply the method of finding non-dominated genes as explained previously. If the number of non-dominated genes is less than the given neighbor count, we remove the selected genes from the data representation and re-apply the method of the finding non-dominated genes. We continue this process until the given number of neighbor count is reached.

- Regulated genes (Item) selection: The genes that the target gene has a connection are decided by using collaborative filtering. In this process, the known connections of neighbor genes are used to decide on best matching gene to be predicted as the regulated gene by the target gene. The genes which are already known to be regulated by the neighbor genes are assigned as the candidate genes. For each candidate gene a connection score is calculated by the Equation 5.3. The higher connection score indicates that the candidate gene is more promising to be regulated by the target gene. In the Equation 5.3, the

score represents the connection score, *t* represents the target gene, *n* represents the neighbor gene and *c* represents the candidate gene. In the calculation the similarity among the target and neighbor genes ($sim(t, n)$) and binding probabilities of the neighbor and candidate genes ($b(n, c)$) are used.

$$score(c) = \sum sim(t, n) \times b(n, c) \tag{5.3}$$

In each step, different settings can be used. The explanations and the abbreviations of these settings are given as follows:

- Multi-Objective Optimization Type (MOT): This setting is related to the step of neighbor selection. As explained in Section 3.1.2, predefined number of neighbors can be selected in multiple iterations. Also it is possible to run only a single iteration and collect the non-dominated genes on a single iteration or after running multiple iterations the method may or may not prune the number of neighbor genes to match with the given neighbor count.
  - Only_Dominates (OD): Find non-dominated neighbors in a single iteration. The number of non-dominateds genes is not set and it depends directly on the similarity values.
  - N_Dominates (ND): Find exactly N neighbors by running multiple iterations and pruning when necessary.
  - At_Least_N_Dominates (AND): Find at least N neighbors by running multiple iterations. Unlike N_Dominates setting, no pruning is applied in this setting.
- Outlist Type (OT): This setting is related to the regulated genes (item) selection step. The output of the method may contain a fixed sized list or may depend on threshold on the similarities, binding probabilities or connection scores of the genes, which are used in the Equation 5.3.
  - Fixed_Length (F): Recommend k items.
  - Threshold_Based (T): Recommend items whose connection score are larger than the threshold T. In this setting, also the candidate genes whose bind-

ing probability to the neighbors are less than T are removed from the candidate list.

- Regulated genes (Item) Selection Method Type (IST): This setting is related to the regulated genes (item) selection step. Even though we presented a general equation to calculate the connection scores of genes to the target gene, different values can be used for similarities and binding probabilities used in the Equation 5.3.

  - Sum (SUM): Without considering similarities between the target and the neighbor genes, the binding scores for each candidate gene is summed, such that $sim(t, n) = 1$ for all neighbors.
  - Average (AVG): After summing up values -as done in SUM method-, the result is divided into the number of neighbors that suggest the candidate gene.
  - Maximum (MAX): For each candidate gene, the maximum binding probability is used, without considering similarity between the target gene and the neighbor genes.
  - Weighted Average (WAVG): AVG method is performed by additionally using the similarities among the target and the neighbor genes. So instead of dividing the summation into the number of neighbor genes, it is divided into the summation of the similarities between the target and the neighbor genes. In the application we used binding probabilities between target and neighbor gene, instead of calculating similarities, such that $sim(t, n) = b(t, n)$.

  In Figure 5.2, an example is given to show how different regulated genes (item) selection method affects the predictions.

  In the figure the aim is to predict a single gene that the target gene ($g0$) regulates. The neighbors are given as $g1$, $g2$ and $g3$. the binding probabilities of the neighbors and the candidate genes ($g4$ and $g5$) are also given. Based on different regulated genes (item) selection method, different genes are predicted to be regulated by the target gene. For example, when summation is used, the gene $g4$ is predicted since its connection score is 1.6 and the score of the gene $g5$ is

Target: g0, Neighbors: g1, g2, g3 Candidates: g4, g5 Binding prob: g1→ g4: 0.8 g2→ g4: 0.8 g3→ g5: 0.9	Predicted gene to be regulated is: • Sum: g4 • Avg: g5 • Max: g5 • Weighted avg: g4 or g5 (Depending on the similarities between the target and neighbor genes)

Figure 5.2: Example for regulated genes (item) selection method

0.9. However, when averages are used the average score of $g4$ is 0.8 and the average score of $g5$ is 0.9, so $g5$ is predicted to be regulated. For weighted average method, the similarities between the target gene, $g0$, and the neighbors, $g1$, $g2$ and $g3$, affects the prediction results and depending on these values either $g4$ or $g5$ can be predicted to be regulated by the target gene.

5.2 Evaluation of Gene Regularity Network Structure Prediction

In this section we presented the evaluation process and results of GRN structure prediction. The data-set and the evaluation results are presented in Sections 5.2.1 and 5.2.2.

5.2.1 Bio-informatics Datasets

For the evaluation of GRN predcition, we run two different experiments. In the first one, we combined two different data-sets, namely microarray data from Spellman et al.[119] and transcription factor (TF) binding data from Lee et al. [66], to observe the performance of our method. In the second experiment, we used the Dream4 In Silico Network Challenge data-set [14] to see the performance of our method on different sizes of data-sets. We present the details of these experiments in the subsections 5.2.1.1 and 5.2.1.2.

5.2.1.1 Microarray and Transcription Factor (TF) Binding Datasets

The first data-set [119] contains time series gene expression data, in which there are 6178 genes and 77 time steps. In [9] these time steps are divided into three phases. In this section we used each phase as a different feature, rather than using each time step as a feature. The second data-set [66] contains binding location data of 6270 genes and 106 TFs. Even though the data-sets contain many more genes, in the [120] 25 of them are chosen based on the studies in [9]. Following that work, we also worked on the same 25 genes.

In [120] a commercial tool is used to collect the golden data. Unlike them, we preferred to use a public tool named as GeneMANIA [37]. GeneMANIA provides various information based on interaction types: Genetic interactions, Co-localization, Co-expression, Physical interactions, Shared protein domains, Other. We collected information for the selected 25 genes for all the interaction types from GeneMANIA on March 10, 2015. If not explicitly stated otherwise, we presented the average of all interaction types as the evaluation results.

5.2.1.2 Dream4 In Silico Network Challenge Dataset

Dream4 In Silico Network Challenge [14] is prepared by researchers from the Laboratory of Intelligent Systems of the Swiss Federal Institute of Technology in Lausanne and the IBM T.J. Watson Research Center in New York [87, 86, 105]. Its goal is defined as reverse engineering the GRNs from simulated steady-state and time-series data, such that inferring the directed gene network. In the challenge three sub-challenges are defined, which are InSilico_Size10, InSilico_Size100 and InSilico_Size100_Multifactorial. In this thesis, we attacked the InSilico_Size10 and the InSilico_Size100 sub-challenges, with the aim of observing the performance on data with different sizes. For all of the sub-challenges, information for five different networks are provided and rankings of the teams are decided based on the predictions made for all five networks. Their rational to provide multiple networks instead of one is that they wanted to measure the consistency of the methods on independent networks with different topologies. They provide the golden data and the evaluation

scripts for the researchers after the end of the challenge.

The InSilico_Size10 sub-challenge five networks containing 10 nodes are provided. The data provided with this challenge are wild-type, knockouts, knockdowns, multifactorial perturbations, and time series data. Similarly, The InSilico_Size100 sub-challenge contains five networks with 100 nodes. The data provided are same, except for this data-set the multifactorial perturbations data is not included. All the data provided corresponds to noisy measurements of mRNA levels, in which the maximum normalized gene expression value is 1. We chose to use knockdowns, knockouts, multifactorial perturbations;if available; and time series data to performs the similarity calculations. In the original data-set time series data contains 5 different time series for the network sized 10 and 10 time series for the network sized 100 and each time series contains 21 time points. We combined the time series information by getting average of the similarities, which are calculated independently for each time series.

5.2.2 Evaluation Results

For the evaluation on microarray and transcription factor (TF) binding data-sets [119], [66], we used Precision@k, Recall@k and F1-measure, which are commonly used metrics in the literature. In order to evaluate our method on the Dream4 In Silico Network Challenge [14] data-set, we used the evaluation scripts and the golden data provided in the challenge website [14]. For the scoring the area under the precision versus recall curve, precision at 1%, 10%, 50%, and 80% recall, and the area under the ROC curve are calculated. The ranks of the teams in the challenge are decided based on the overall performance. In this report, we used the overall score to present our results as well.

We present the evaluation results for the two data-sets, namely microarray and transcription factor (TF) binding data-sets [119], [66]and Dream4 In Silico Network Challenge [14] data-set, in the following sections.

5.2.2.1 Evaluation Results for Microarray and Transcription Factor (TF) Binding Datasets

Before evaluation, we executed the same pre-processing steps performed in the [120] on the data-set:

- Filling the missing values in the microarray data-set [119]: The missing values existing in the data-set for the selected 25 genes are filled by applying the k-nearest neighbors algorithm with k = 10, same as it is done in [120].
- Converting p-values in the binding data [66] into probability values: The binding data [66] gives information on the p-values, which indicates the confidence of TF bindings to the genes [120]. Smaller p-values indicates the higher confidence. As suggested in [120], we converted the p-values into probabilities of existence of connections. For this purpose Equation 5.4 are used, which is described in [9]. In the equation, E_i represents one of the edge/connection in the graph G. P_{ij} is the p-value of the edge E_{ij} which connects the genes i and j. $The\lambda_H$ and λ_L are the highest and lowest bounds of the λ, which is a parameter of exponential distribution. The ϑ_{ij} is the short form for $P(E_{ij} \in G)$, such that $\vartheta_{ij} = P(E_{ij} \in G)$. We assigned the values of λ_H, λ_L and ϑ_{ij} to 10000, 0.1 and 0.5 respectively, as suggested in [9] and [120].

$$P(E_i \in G \mid P_i = p) = \frac{1}{\lambda_H - \lambda_L} \int_{\lambda_L}^{\lambda_H} \frac{\lambda e^{-\lambda p_i} \vartheta_{ij}}{\lambda e^{-\lambda p_i} \vartheta_{ij} + (1 - e^{-\lambda})(1 - \vartheta_{ij})} d\lambda \tag{5.4}$$

- Filling missing probabilities in the binding data [66]: Among the selected 25 genes, 10 of them exist as TFs in the the binding data. In the previous step, only for those 10 TFs/genes the probabilities are calculated. For the rest of them, we set their binding probabilities to 0.50, as done in [120].

We combined the microarray data[119] and the binding data [66] in two different ways. In the first of them, we used the three phases extracted from the microarray data for the similarity calculations step, and the binding data for the regulated genes (item)

Table 5.1: The best results for the F1-measure and with the 3F_Exp

N	k	T	MOT	OLT	IST	Precision	Recall	F1
1	23	-	ND	F	SUM	0.183	**0.953**	0.300
1	23	-	ND	F	AVG	0. 183	**0.953**	0.300
22	21	-	AND	F	MAX	**0.199**	0.943	**0.321**
1	23	-	ND	F	WAVG	0. 183	**0.953**	0.300

selection step. In the second of them, we added the binding data to the similarity calculations step. As a result, for the first experimental setting, we used three features and for the second one we used four features. In the following paragraphs we will refer these setting as 3F_Experiment and 4F_Experiment, respectively.

In the experiments, depending on the settings explained in the previous section, we need three variables to be assigned, which are neighbors count (N), the output list size (k) and threshold (T). The performance of the methods may differ based on these parameters. We performed tests by assigning different values to these parameters: For N, we assigned the range to [1,25], where 25 is the total number of the genes. Similarly, for k we assigned the range to [1,25], where 25 is the total number of the genes. For T, we used the range of [0.51,1.00], with 0.03 increments.

We present the best results for each regulated genes (item) selection method type (IST), using 3 or 4 features (3F_Experiment or 4F_Experiment). Depending on the evaluation metric; precision, recall and F1-measure; different parameters provides the best results. We present the best results for different metrics in the following parts.

In Table 5.1, we present the best results for the F1-measure and with the 3F_Exp. According to the table the best performing method is the one that chooses at least N many neighbors (AND) when using the MAX as the item selection approach. However, this approach uses nearly all of the genes as the neighbors; such that chooses 22 out of the 25 genes as the neighbors; and presents most of the genes in the output list; such that 21 out of the 25 genes.

In Table 5.2, we present the best results for the precision and with the 3F_Exp. According to the table the best performing method is the one that chooses N many neighbors (ND) when using different item selection approaches. In the best setting the number of neighbor genes to be selected is 12 and the output list size is set as 2.

Table 5.2: The best results for the precision and with the 3F_Exp

N	k	T	MOT	OLT	IST	Precision	Recall	F1
12	2	-	ND	F	SUM	0.301	**0.157**	**0.198**
12	2	-	ND	F	AVG	0.301	**0.157**	**0.198**
3	1	-	AND	F	MAX	**0.402**	0.092	0.145
12	2	-	ND	F	WAVG	0.301	**0.157**	**0.198**

Table 5.3: The best results for the recall and with the 3F_Exp

N	k	T	MOT	OLT	IST	Precision	Recall	F1
6	24	-	AND	F	SUM	**0.181**	0.981	0.299
6	24	-	AND	F	AVG	**0.181**	0.981	0.299
22	24	-	AND	F	MAX	**0.181**	**0.983**	**0.300**
6	24	-	ND	F	WAVG	**0.181**	0.978	0.298

In Table 5.3, we present the best results for the recall and with the 3F_Exp. According to the table the best performing method is the one that chooses at least N many neighbors (AND) when using MAX as the item selection approach. In the best setting the number of neighbors to be selected is 22 and the output list size is set as 24, such that all the genes are presented.

According to the results for precision, recall and F1-measure for the 3F_Exp, we observe that F1-measure and recall favor selection of many neighbor genes and predicting too many, nearly all, genes as being regulated by the target gene. Since it is known that the GRNs are sparse, this tendency does not seem to be correct. So, we decided to use precision as our main objective in the rest of this section.

In Table 5.4, we present the best results for the precision and with the 4F_Exp. According to the table the best performing method in terms of precision is the one that chooses N many neighbors (ND) when using MAX as the item selection approach. In this setting the number of neighbor genes to be selected is 3 and the output list size is set as 1. For recall and F1-measure, the best performing method is the one that chooses at least N many neighbors (AND) when using WAVG as the item selection approach. In this setting the chosen N and k values are 5 and 2, respectively. Comparing the performance of 3F_Exp and 4F_Exp, we observe that adding the binding data for the similarity calculations increases the performance slightly.

Table 5.4: The best results for the precision and with the 4F_Exp

N	k	T	MOT	OLT	IST	Precision	Recall	F1
6	2	-	AND	F	SUM	0.301	0.157	0.198
6	2	-	AND	F	AVG	0.301	0.157	0.198
3	1	-	ND	F	MAX	**0.404**	0.097	0.151
5	2	-	AND	F	WAVG	0.310	**0.161**	**0.203**

Table 5.5: The results for the undirected graph

N	k	T	MOT	OLT	IST	Precision	Recall	F1
1	1	-	ND	F	SUM	0.275	0.124	0.163
1	1	-	ND	F	AVG	0.275	0.124	0.163
5	1	-	AND	F	MAX	0.333	0.098	0.146
1	1	-	ND	F	WAVG	0.275	0.124	0.163
6	4	-	AND	F	SUM	0.248	**0.470**	0.299
6	4	-	AND	F	AVG	0.248	**0.470**	0.299
10	1	-	ND	F	MAX	**0.342**	0.085	0.132
6	4	-	ND	F	WAVG	0.250	0.464	**0.300**
-	-	-	-	-	-	0.213	0.193	0.203

Our method provides the directed graph, such that it predicts which gene regulates the others. However, in [120] the only graph provided in the paper is undirected. Since we want to compare our result to theirs, we also converted our directed graph into undirected by adding the reverse directions of the edges to the graph. In Table 5.5, we present the results for 3F_Exp, 4F_Exp and [120], in the order of the sections seen in the graph. According to the table, the best method in terms of precision is the one that uses N many neighbors with MAX as the item selection method. For recall the best method is the one that chooses N neighbors by using SUM or AVG as the item selection method. For F1-measure, the best method uses ND with weighted average approach. For all of the measures, the best results belong to the 4F_Exp setting. We can conclude that adding the binding data to the similarity calculations step increases the performance.

We observed from the tables that for directed graphs weighted average (WAVG) method for choosing items works better when we consider all the measures; i.e. precision, recall, F1-measure. For undirected graph, there is no single winner for the item selection method. For all of the experiments, using a fixed length output list (F)

performed better than using a threshold (T). Also we observed that using exactly N many neighbors (ND) mostly performed better than other approaches. The next thing we want to decide is the best values for N and k. In Figures 5.3 and 5.4, we present the plot of precision values for different N and k values for the experiments using three or four features. For both of the experiments, we observe that increase in k decreases the precision. For the 3F_Exp, the best precision is obtained when N is set to 12 and k to 2. For the 4F_Exp, the best precision is obtained when N is set to 6 and k to 2. Even though the performance results of both experiments are similar, adding the binding data to the similarity calculations (i.e. 4F_Exp) help the system to reduce the calculations by decreasing the necessary number of neighbors to choose. Note that for both of the experiments the number of genes to be regulated by the target gene (i.e. k) is found to be 2, which is a small value. This observation matches with the sparsity feature of GRNs.

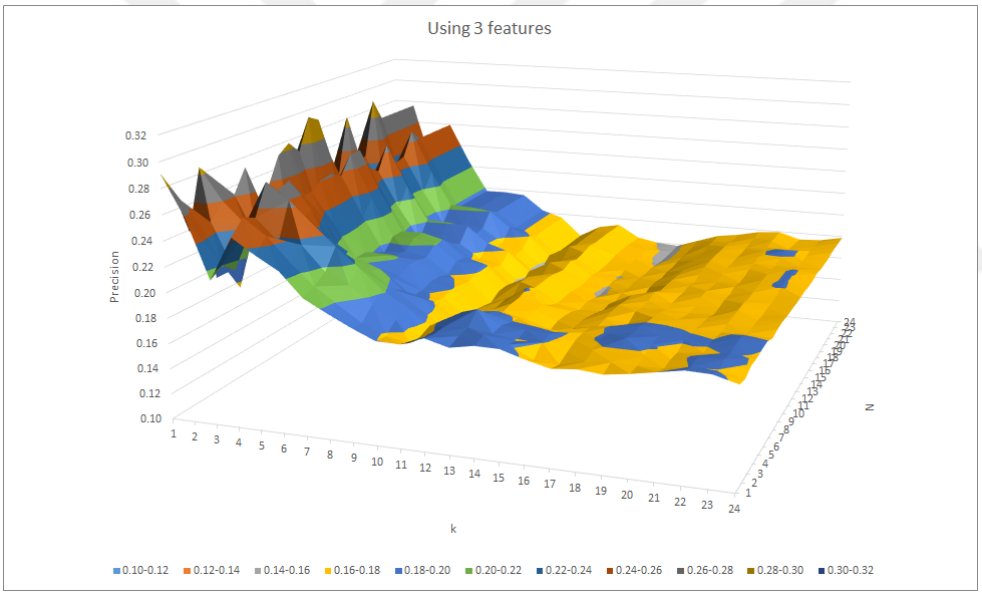


Figure 5.3: The precision results for ND and WAVG with different N and k (3F_Exp)

To be fair to all of the multi-objective optimization types, we also get the results using weighted average (WAVG) method for choosing items and fixed length output list, where N is set to 6 and 12 and k is set 2, based on the results observed from the previous figure. In Tables 5.6 and 5.7 the precision results are given. The results for the 3F_Exp are shown on the upper parts of the tables, while for the 4F_Exp are shown on the lower parts. According to the tables, when N is set to 6, the best performance is obtained by ND with four features (4F_Exp) and when it is set to 12

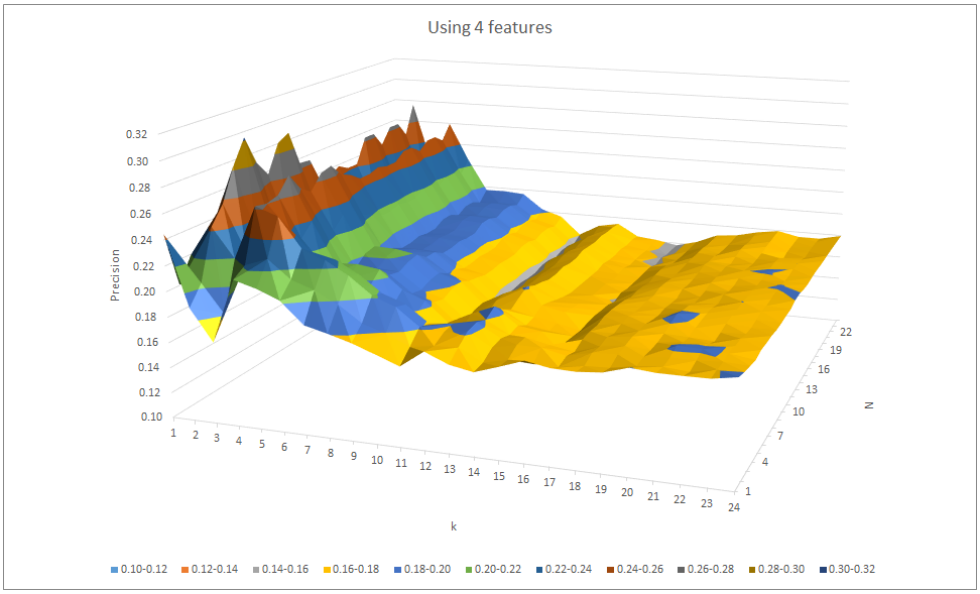


Figure 5.4: The precision results for ND and WAVG with different N and k (4F_Exp)

Table 5.6: The results for N=6 and k=2 with WAVG item selection method

N	k	T	MOT	OLT	IST	Precision	Recall	F1
-	2	-	OD	F	WAVG	0.263	0.126	0.164
6	2	-	AND	F	WAVG	0.284	0.148	0.186
6	2	-	ND	F	WAVG	0.279	0.146	0.183
-	2	-	OD	F	WAVG	0.255	0.128	0.164
6	2	-	AND	F	WAVG	0.288	0.149	0.188
6	2	-	ND	F	WAVG	**0.302**	**0.155**	**0.196**

the performance is obtained by ND with three features (3F_Exp). These results are consistent with the previous results and shows that fixed number of neighbors and using multiple features from multiple sources are useful approaches to predict the genes to be regulated.

Lastly, we present the results for each interaction type provided by GeneMANIA, which are genetic interactions, co-localization, co-expression, physical interactions, shared protein domains, other. Based on the results of the previous experiments we chose to present the results of 4F_Exp using ND, WAVG, N=6 and k=2. Based on the results presented in Table 5.8, our method performs well for Genetic interactions, Physical interactions, Shared protein domains and Other, but relatively less well for Co-localization and Co-expression.

Table 5.7: The results for N=12 and k=2 with WAVG item selection method

N	k	T	MOT	OLT	IST	Precision	Recall	F1
-	2	-	OD	F	WAVG	0.263	0.126	0.164
12	2	-	AND	F	WAVG	0.267	0.142	0.177
12	2	-	ND	F	WAVG	**0.301**	**0.157**	**0.198**
-	2	-	OD	F	WAVG	0.255	0.128	0.164
12	2	-	AND	F	WAVG	0.242	0.133	0.165
12	2	-	ND	F	WAVG	0.263	0.141	0.176

Table 5.8: The results for ND, WAVG, N=6 and k=2

Int. Type	Precision	Recall	F1
Genetic interactions	0.469	0.192	0.273
Co-localization	0.125	0.060	0.081
Co-expression	0.318	0.084	0.133
Physical interactions	0.227	0.179	0.200
Shared protein domains	0.300	0.273	0.286
Other	0.375	0.140	0.204

5.2.2.2 Evaluation Results for Dream4 In Silico Network Challenge Dataset

As in the previous experiment, for Dream4 In Silico Network Challenge [14] data-set we need to set neighbors count (N), the output list size (k) and threshold (T) variables. We performed tests by assigning different values to these parameters: For N and k, we assigned the range to [1,M], where M is the total number of the nodes provided in the data, such that 10 or 100. However, for the InSilico_Size100 data-set we limited the M as 15, as it would take long time to run the experiments up to 100 and we have observed in the previous experiments that less number of neighbors can capture the necessary information. For T, we used the range of [0.51,1.00], with 0.03 increments.

In Table 5.9, based on the overall score, the best five settings for InSilico_Size10 challenge are presented. For this result, we used the average time series similarities as the parameter ($b(n, c)$) to be used in the regulated genes (item) selection step. Based on the table, we observed that the best performing setting N is set to 9, k is set to 6, at least N many neighbors are selected (AND) and as MAX is used as the regulated genes (item) selection type. Using these settings, we performed the same calculations while using different data as the regulated genes (item) selection parameter. From

Table 5.9: The top 5 results for InSilico_Size10 challenge (Avg. Time series)

N	k	T	MOT	OLT	IST	SCORE
9	6	-	AND	F	MAX	**1.119**
9	6	-	ND	F	MAX	**1.119**
8	6	-	AND	F	MAX	1.001
9	7	-	AND	F	MAX	0.902
9	7	-	ND	F	MAX	0.902

Table 5.10: The results for InSilico_Size10 challenge

Type	SCORE
Avg. Time	**1.119**
Knockdowns	1.078
Knockouts	**1.119**
Multifactorial pert.	**1.119**

Table 5.10, we observe that except knockdowns information, all the data performs equally well.

We performed the similar calculations for the InSilico_Size100 challenge. According to Table 5.11 the best performance is obtained when N is set to 11, threshold based calculations are done with the threshold value of 0.99, the best performing method is ND, which choose exact N many neighbors, and the best item selection type is weighted average (WAVG). In Table 5.12 we present the results while using 11 neighbors with ND and WAVG settings. Instead of directly using the threshold obtained while using average time series similarity as the binding probability, we searched for the best threshold value. According to this table, best feature to be used for binding probability is the similarity on knockouts.

Table 5.11: The top 5 results for InSilico_Size100 challenge (Avg. Time series)

N	k	T	MOT	OLT	IST	SCORE
11	-	0.99	ND	T	WAVG	**3.108**
8	-	0.99	AND	T	WAVG	3.043
9	-	0.99	AND	T	WAVG	3.033
9	-	0.99	ND	T	WAVG	3.023
10	-	0.99	AND	T	WAVG	2.967

Table 5.12: The results for InSilico_Size100 challenge

Type	N	k	T	MOT	OLT	IST	SCORE
Avg. Time	11	-	0.99	ND	T	WAVG	3.108
Knockdowns	11	-	0.54	ND	T	WAVG	2.039
Knockouts	11	-	0.99	ND	T	WAVG	**3.211**

The scores and the rankings of the teams attended to the challenge are given in the challenge web-page [107]. The number of teams attended to the InSilico_Size10 challenge is 29 and to the InSilico_Size100 challenge is 19. Based on our best performing settings, if we were attended the challenge, our rank would be 25 for InSilico_Size10 challenge and 17 for InSilico_Size100 challenge. Even though the results show that our method performs better than some other methods in the literature, we observed that not having the information on which genes are regulated by the neighbor genes diminishes the performance of our method. Since the data-sets of this challenge is produced only for the challenge and are not correspond to real world/experimentally known genes, it not possible to map the genes known genes and collect information on those genes via some other resources, such as GeneMANIA.

5.3 Conclusion

Recommendation methods can also be used by different applications related to different branches of science. Gene regulatory network (GRN) inference, from biology, is one of these applications. In this chapter, observing the common features of recommendation systems and GRNs, we used the proposed multi-objective optimization based recommendation method to predict the gene relationships; such that which genes regulates the others. For the purpose of GRN re-construction, instead of target users we used target genes and instead of making item recommendations to be used in the future we predicted the genes that are regulated by the target gene. For the evaluation we used several different data-sets. The results showed that using information from multiple sources improves the performance. Also, we observed that use of an approach from recommendation systems performs well. We anticipate that other recommendation approaches can be used in GRN inference problem in the future.

CHAPTER 6

DRUG REPOSITIONING USING PARETO DOMINANCE AND COLLABORATIVE FILTERING

The aim of this work is to predict new uses of known drugs by analyzing multiple features and multiple data sources. For this purpose, we adapted a recommendation system based method which has been successfully applied in other domains. Fortunately, the results reported from this study clearly demonstrate the effectiveness and applicability of recommendation methods for drug repositioning because the process could be easily mapped to recommending an existing drug for handling a new disease by studying characteristics of news diseases in link to already known diseases and their associated drugs. Zhang el al. [145] stated that similar drugs are indicators for similar diseases. Accordingly, in their work they used similar drugs' indications to re-position the target drugs. Realizing the fact that this approach is similar to collaborative filtering in the recommendation systems domain, we adapted our proposed method in this thesis to drug re-positioning problem. We present the proposed method in general and we detail steps of the method in Sections 6.1 and 6.2. We present the evaluation process and the results in Section 6.3. The chapter is concluded in Section 6.4.

6.1 Pareto dominance and collaborative filtering based prediction

The method that we proposed for recommendation uses Pareto dominance and collaborative filtering approaches to predict future venue preferences (i.e, check-in locations) of target users. Its idea is based on the observation that similar users tend

to visit similar venues, and recommending to the target user venues that have been visited by similar users performs well. We also applied to the bioinformatics domain for predicting the structure of gene regulatory networks as described in the previous section. In that section, instead of target users, target genes are used and regulated genes are predicted. The achieved results confirmed promising aspects of mapping a recommendation system to discover gene regulation.

Figure 6.1: Design of the proposed method

The study of gene regulatory networks motivated us to investigate the applicability of recommendation systems to drug re-positioning. The overall design of the proposed method for drug re-positioning is shown in Figure 6.1, where the modules and their interactions, are presented. The proposed method is composed of three main steps, namely similarity calculation, neighbor selection and item (disease) selection. In the similarity calculation step, each feature is used to determine similarity between drugs. Then, similarities are used to find most similar drugs, namely neighbors, by a Pareto dominance based method. Then known connections among neighbor drugs and indicated diseases are used for prediction. At the end, we present a prediction list of target drugs and predicted diseases which could be treated by target drugs.

6.2 Details of the proposed method

For the calculations, we used three main features: namely chemical properties of drugs, protein targets, and side-effect profiles. In this section, we explain details of the various steps of the proposed method and how the above-mentioned features are used.

Similarity calculation: In this step, similarity between drugs is calculated for each type of features. We used several similarity measures in the calculation, namely Cosine similarity, Jaccard similarity and a similarity score based on Smith-Waterman sequence alignment. In this section, we present how these similarity measures are calculated. In the evaluation section, we present how we used these similarity measures, how we combined them, and their performance results.

Equation 6.1 presents how Cosine similarity is calculated, where drugs are denoted by *A* and *B*. Drugs may be represented as vectors, such that a vector contains one value per feature to reflect how a drug is related to the specific feature. In the equation the subscript *j* refers to individual values of a feature vector. For instance, for the "chemical properties" feature, a drug may be represented as a binary vector where values represent the existence/non-existence of a chemical structure. Similarity between two drugs can be calculated based on common chemical structures and the length of the feature vector.

$$sim(A,B) = \frac{\sum_{j=1}^{n} A_j \times B_j}{\sum_{j=1}^{n} A_j^2 \times \sum_{j=1}^{n} B_j^2} \tag{6.1}$$

Equation 6.2 presents how Jaccard similarity is calculated. In the equation, drugs are indicated by *A* and *B*. Here, $|A|$ represents length of the drug feature vector and $|AB|$ represents size of common elements in the feature vector. This similarity measure is also called Tanimoto index/similarity when the feature vector is binary.

$$sim(A,B) = \frac{|AB|}{|A| + |B| - |AB|} \tag{6.2}$$

In the work of Zhang et al. [145], a similarity score based on Smith-Waterman sequence alignment is used. In this study we also applied the same similarity measure when possible. As explained previously, drugs may be represented as a feature vector. Entries/elements of a vector themselves can be represented as sequences. For instance, a drug can be represented as a vector of proteins. Proteins themselves may be represented as a sequence of smaller biological elements. Similarity of these se-

quences, e.g. protein sequences, can be calculated by Smith-Waterman sequence alignment method. After having Smith-Waterman sequence alignment score, similarity among drugs can be calculated by the formula given in Equation 6.3.[1]

In Equation 6.3, drugs are indicated as *A* and *B*. $V(A)$ represents the feature vector for drug A, and $|V(A)|$ represents the length of that vector. In the equation, each vector element is composed of a sequence of smaller elements, where these elements are represented as $V_i(A)$. Smith-Waterman sequence alignment score computed in Equation 6.3 is denoted $sim_SW(V_i(A), V_j(B))$.

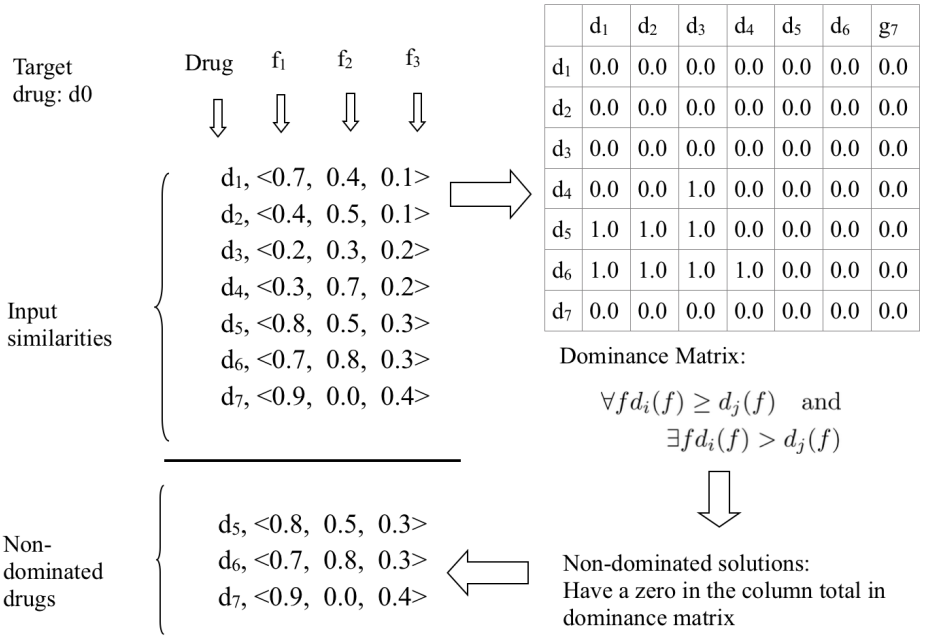
$$sim(A,B) = \frac{\sum_{i=1}^{|V(A)|} \sum_{j=1}^{|V(B)|} sim_SW(V_i(A), V_j(B))}{|V(A)| \times |V(B)|} \tag{6.3}$$

Neighbor selection: In this step, most similar drugs to the target drug (i.e., its neighbors) are selected. Neighbors are decided by using the similarities calculated in the previous step and by applying a Pareto dominance based method. In this method drugs that are not dominated by other drugs are selected as neighbors. Dominance relation among drugs is decided by Equation 6.4, where d_i and d_j represent drugs and f indicates features. According to the equation, if drug d_i has at least one higher similarity value and no lower similarity values than drug d_j, then drug d_i dominates drug d_j.

$$dom(d_i, d_j) = \begin{cases} 1.0 & \forall f \; d_i(f) \geq d_j(f) \text{ and} \\ & \exists f \; d_i(f) > d_j(f) \\ 0.0 & \text{otherwise} \end{cases} \tag{6.4}$$

An example input and non-dominated solutions are given in Figure 6.2, where the data-set is composed of eight drugs and the target drug is identified as drug d_0. The similarities between drugs for each feature f_i is also listed. First, based on these similarities dominance matrix is created using Equation 6.4. Then non-dominated drugs (i.e., drugs with zero column total in the dominance matrix) are selected as neighbors. In the example, d_5, d_6 and d_7 are selected as most similar drug to the target drug.

[1] We used Uniprot to collect protein sequence information and ClustalX2 for protein sequence alignment.



	d1	d2	d3	d4	d5	d6	g7
d1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
d2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
d3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
d4	0.0	0.0	1.0	0.0	0.0	0.0	0.0
d5	1.0	1.0	1.0	0.0	0.0	0.0	0.0
d6	1.0	1.0	1.0	1.0	0.0	0.0	0.0
d7	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 6.2: Example input and non-dominated solutions

As explained previously, the application of Pareto dominance based approach on a single iteration may provide less than the predefined number of neighbors. In order to collect as many neighbors as predefined, an iterative process can be applied. In each iteration, first, non-dominated neighbors are found and are removed from the first set of candidates. Then iterations are executed until the predefined number of neighbors are collected. At the end, if the collected number of neighbors is more than the predefined number; i.e., more than expected non-dominated drugs are found in the last iteration, neighbors can be pruned into exact number of neighbors or neighbor list may remain as it is.

- Only_Dominates (OD): Execute single iteration to find non-dominated neighbors. The number of non-dominated drugs is not set, and it depends directly on similarity values.
- N_Dominates (ND): Execute multiple iterations to find non-dominated neighbors. The number of non-dominated drugs is set exactly to N, i.e. pruning is applied when necessary.
- At_Least_N_Dominates (AND): Execute multiple iterations to find non-dominated

neighbors. The number of non-dominated drugs is set at least to N, i.e. pruning is not applied.

Item selection: In this step, items to be recommended are selected. In the problem investigated in this study, items selected in this step are diseases for which the target drug could be re-positioned. In this step, first candidate items are identified by collecting items with which neighbors are related, i.e. diseases are listed as indicators for neighbor drugs. For each candidate item (disease) a score is calculated by Equation 6.5, where the score is denoted *score(c,t)*, the candidate item (disease) is denoted c, the target is denoted t, and the neighbor is denoted n. Similarity between the target and neighbor drugs is given as $sim(t, n)$. The function $f(n, c)$ represents neighbor drug-candidate disease relationship score given in the input data. This score can be different than zero and one, but our data-set is represented as binary vectors, i.e. a drug has a relation with a disease or not and the values of f(n,c) is either one or zero. Higher item selection score means the target drug has a more promising relation with the candidate disease.

$$score(c, t) = \sum_{n \in \text{Nghb}} sim(t, n) \times f(n, c) \tag{6.5}$$

For computing the score, two different settings can be used. We called them Item Selection Type (IST) settings which could be described as follows:

- Sum (SUM): Without considering similarities between the target and neighbor drugs, votes (summation of $f(n, c)$ values) are calculated for each candidate. Items (disease) which have highest number of votes are presented in the output list. This setting is already presented in the previous sections.

- Weighted Sum (WSUM): For the summation, $sim(t, n)$ value is also included, such that more similar drugs have more weight on the prediction. Items (disease) which have highest scores are presented in the output list.

6.3 Evaluation of Drug Repositioning

In this section we presented the evaluation process and results of identification of new indications for known drugs, drug re-positioning. The data-set and the evaluation results are presented in Sections 6.3.1 and 6.3.2.

6.3.1 Drug Repositioning Datasets

As the golden data-set, we used the drug-disease data provided by Zhan et al. [145]; the same data-set was also used by Li et al. [69]. The data-set integrates three data sources, namely chemical data source, protein data source and side-effect data source.

- The chemical data source contains information about drug and PubChem [127] chemical substructures relationships. It contains relationship between 1007 drugs and 881 PubChem chemical substructures; number of associations (i.e. edges) is 122022. Sparsity of the data-set is about %86.25.
- The protein data source contains information about drug and UniProt target proteins relationships. It contains relationship between 1007 drugs and 775 target proteins. Number of associations in this data-source is 3152, which means that the data-source is %99.60 sparse. For this data-source, target drugs are generated by using DrugBank [132].
- The side-effect data-source contains information about drugs and side-effects. In the data-source there are 888 drugs and 1385 side-effects. Number of associations in this data is 61102 and sparsity ratio is %95.03. Information on this data-source is generated from SIDER [59].

Each data source contains information about a single feature and these features are represented as a binary vector. The drugs listed in each data source are not necessarily the same. Based on this, the overall data-set (combination of all the three data-sources) contains more than 1007 drugs. Since drugs in each data source may be different, drugs may have missing information about one or more features.

In this section, after obtaining the data-set of Zhang et al. [145], we applied a preprocessing step to collect a list of drug names and the mapping to drug names in chemical, protein and side-effect data sources. During this process, we noticed that some drugs may have different names (synonyms). For example, we found that one drug is referred to as *Ursodiol* in chemical data source, while it is referred to as *Ursodeoxycholic acid* in protein and side-effect data-sources. We looked up synonyms from DrugBank website [26]. As a result of the preprocessing step, we obtained 1224 different drugs with the mappings of their names.[2]

The golden data-set contains relationship information among 799 drugs and 719 diseases, with 3250 treatment relations (edges). However, not all drugs listed in this data-set are listed in the input data sources (chemical, protein and side-effect data). Since it is nearly impossible to predict indications of a drug without any prior information, we did not consider those drugs as targets. The resulting golden data-set contains 781 drugs, 719 diseases and 3179 relations[3].

6.3.2 Evaluation Results

In order to evaluate the performance of the methods we used Precision@k, Recall@k and F1-measure. Figure 6.3 shows the upper bounds of precision, recall and F1-measure for different k values. As expected, precision is at its best when k is smaller and it decreases as k increases. Recall has reverse behavior compared to precision, i.e., it increases as k increases. F1-measure, which is the harmonic mean of precision and recall, reaches its best value when k is equal to 4. We stopped the evaluation when $k = 20$, since recall has already reached 0.9966.

Setting the output list size to exactly k has one drawback. Not all drugs in the golden data-set have association with k-many diseases. If we set output list size exactly to k, then some predictions will always be wrong. For example, assume that k is set to 10, and for target drug d_1, disease associations in the golden set is 5. Then precision will be at most 0.5. However, if k is set to 10 in a loosely way to allow the methods to predict *at most* 10 items, precision may become 1.0. Our proposed method has this

[2] We plan to share the mappings of names on our website.
[3] We will share the golden set on our website.

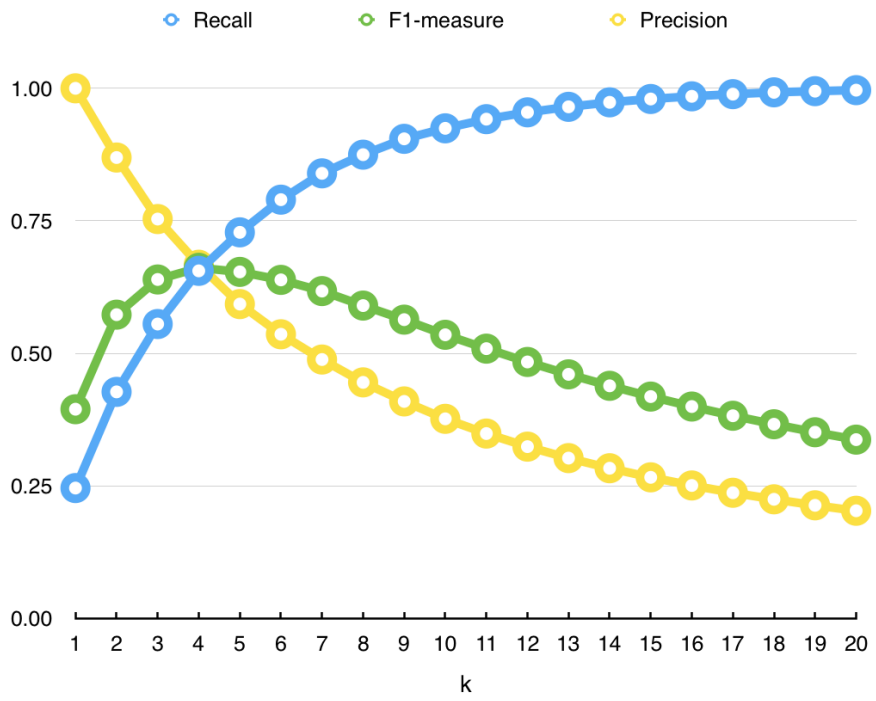


Figure 6.3: Upper bounds recall, precision and F1 measures

ability, it can predict at most k associations and does not make any random guess. We argue that making random guesses for drug re-positioning is not an appropriate idea. It will reduce the benefits of computational drug re-positioning compared to traditional methods. Figure 6.4 shows upper bounds of precision, recall and F1-measure when random guess is not allowed. In this figure, precision is always 1.0, as expected. The recall increases as k increases and this leads to increase in F1-measure. Here it is worth noting that the process of making at most k predictions (without guess) is more challenging, since the method should decide on the best output list size for each target, in addition to making the best prediction.

We conducted experiments using several settings. We used different similarity metrics, Multi-Objective Optimization Type (MOT), and Item Selection Type (IST). For similarity type settings, we concentrated on four different settings that use Cosine similarity, Jaccard similarity or Smith-Waterman sequence alignment based similarity scores for various features, namely chemical, protein and side-effect features. In the first setting (CCC), Cosine similarity is used for all features. In the second setting (JJJ), Jaccard similarity is used for all the features. In the third setting (JJC), Jaccard similarity is used for chemical and side-effect features and Cosine similarity is used

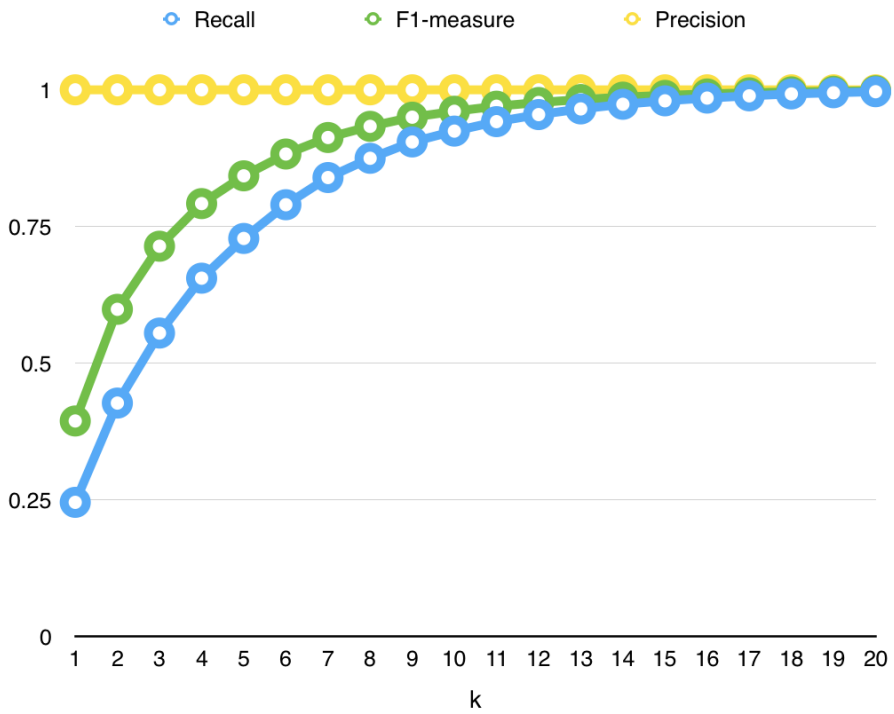


Figure 6.4: Upper bounds of recall, precision and F1-measure when random guess is not allowed

for protein feature. For the last setting (JJS), Jaccard similarity is used for chemical and side-effect features and Smith-Waterman sequence alignment based similarity is used for protein feature.

In the experiments, we need to set two variables, namely neighbors count (N) and output list size (k). We set maximum neighbor count and output list size to 20. Instead of testing with a single value, during the experiments we set N and k to 1, 4, 8, 12, 16 or 20 and conducted experiments using the combination of N and k values. Figures 6.5, 6.6 and 6.7 present the best performance of the proposed method with different settings. The presented results are calculated for each $N \times k$ combinations, but only results of best performing values for the related setting are used. The settings are presented on the x-axis and each line reflects a similarity type (e.g. CCC), MOT (e.g., ND) and IST (e.g., SUM), respectively.

Figures 6.5 and 6.6 reveal that using weighted summation for item selection (WSUM) performs equally well or better than summation (SUM). ND and AND settings as MOT type perform equally well; they perform better than OD which has the limita-

tion of choosing non-dominated neighbors on a single iteration and lead to selection of few neighbors. ND and AND have ability to choose more neighbors and performance results show that choosing more neighbors is more informative. Using different similarity measures during the calculations don't effect the performance much. Using Smith-Waterman sequence alignment based similarity score for protein feature similarity (JJS) performs slightly better than others in terms of precision. Figure 6.7 shows that the performance of all settings are nearly equal. Considering all figures, the observation of performance on F1-measure indicates that methods which perform good on precision do not perform good on recall or methods which perform good on recall do not perform good on precision.

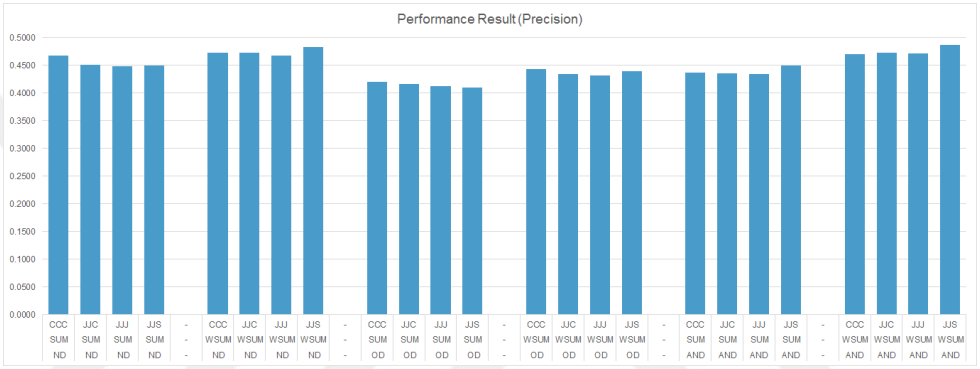


Figure 6.5: Performance results (Precision)

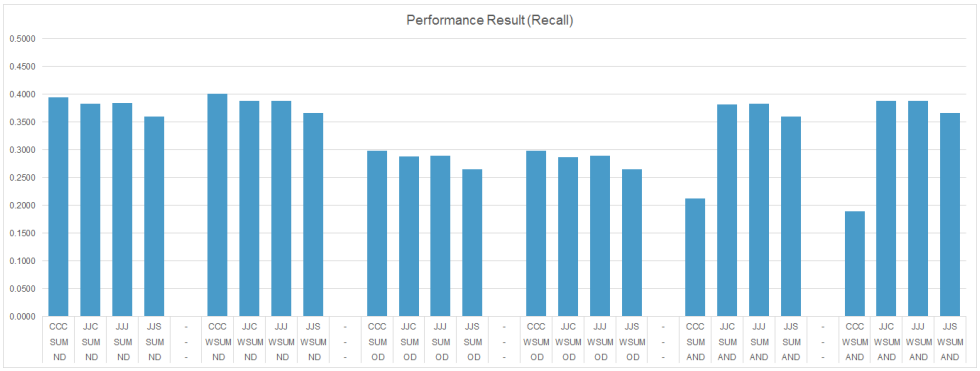


Figure 6.6: Performance results (Recall)

Table 6.1 reports best performance of the settings which use different similarity metrics in more detail. The performance result of each setting is grouped together and in each group we present the approach that produced best precision, best recall and best F1-measure scores. As expected, precision performed better when there are fewer predictions and recall performed better when there are many predictions. While list-

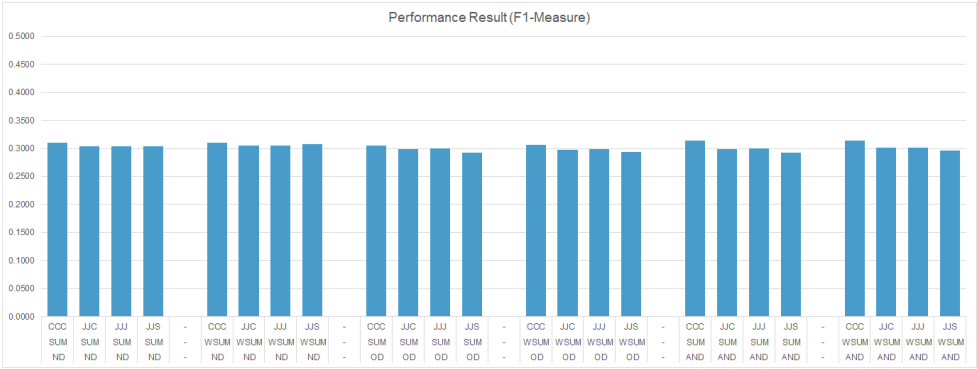


Figure 6.7: Performance results (F1-Measure)

ing only one disease for a target drug produced better precision, listing many (20) diseases as the prediction produced the best recall. We observed that using ND or AND method as Multi-Objective Optimization Type (MOT) performed better compared to OD. During the experiments, we observed that OD (Only dominates) type usually finds a few neighbors. We further observed that having more neighbors is more useful for making better prediction. When we look at the Item Selection Type (IST) we observe that using weighted sum (WSUM) performs better than using sum (SUM). This indicates that it is more informative to integrate similarity between the target drug and its neighbors.

We observed that many of the studies described in the drug re-positioning literature prefer to present AUC-ROC (Area Under Curve - Receiver Operator Characteristic) results. However, it is stated in [23] that for highly skewed data using precision-recall is more informative than using ROC curves. Prediction based on data which has fewer positive relations while having many negative relations is commonly referred to ' in the information retrieval literature as "searching for a needle in haystack'. The golden data we used has similar characteristics, where there are only 3179 positive relations and 558360 negative relations.

Table 6.2 reports the calculated AUC-PR scores of the proposed method and settings. The results show that using Jaccard and Smith-Waterman sequence alignment based similarity scores perform better than other methods, especially when the output list size is limited to few predictions (e.g. $k = 1$).

We also compared our proposed method to the methods described in the literature; the

Table 6.1: The best results when different similarity metrics are used

SimType	N	k	MOT	IST	Prec.	Recall	F1
CCC-Prec.	20	1	ND	WSUM	0.4723	0.0884	0.1489
CCC-Recall	20	20	AND	WSUM	0.1894	**0.4017**	0.2575
CCC-F1	4	1	ND	WSUM	0.2636	0.3762	**0.3100**
JJJ-Prec.	12	1	AND	WSUM	0.4716	0.0862	0.1457
JJJ-Recall	20	20	ND	WSUM	0.1891	0.3888	0.2544
JJJ-F1	4	20	ND	WSUM	0.2621	0.3649	0.3051
JJC-Prec.	12	1	AND	WSUM	0.4723	0.0859	0.1453
JJC-Recall	20	20	ND	WSUM	0.1889	0.3885	0.2542
JJC-F1	4	20	ND	WSUM	0.2629	0.3652	0.3057
JJS-Prec.	12	1	AND	WSUM	**0.4864**	0.0846	0.1442
JJS-Recall	20	20	ND	WSUM	0.2036	0.3671	0.2619
JJS-F1	4	20	ND	WSUM	0.2753	0.3473	0.3071

results are reported in Table 6.3. Actually, we compared our method to state of the art methods which were evaluated using the same data-set we used in this study, namely Li and Lu [69], Chiang and Butte [20] and Zhang et al. [145]. For the proposed method, we presented two of them which produce the best precision and best recall. Since we observed that the methods in the literature usually use ROC and AUC-ROC, we decided to include sensitivity (recall), specificity and AUC-ROC measure as well as precision, recall and F1-measure. Sensitivity (recall) and specificity are used to create ROC. Equation 6.6 shows how specificity (SPC) is calculated. In the equation tn is true negative, i.e, not predicted and actually not indicated diseases, and fp is false positive, i.e., predicted but actually not indicated diseases. Specificity (SPC) measures performance of the methods on negative links (i.e., no indication for a disease). Finally, to calculate AUC-ROC values of the proposed methods we used ROCR library in R.

$$SPC = \frac{tn}{tn + fp} \tag{6.6}$$

Table 6.3 shows that the proposed method with JJS setting performs better than other

Table 6.2: AUC-PR results when different similarity metrics are used

SimType	N	k	MOT	IST	AUC-PR
CCC-Prec.	20	1	ND	WSUM	0.2178
CCC-Recall	20	20	AND	WSUM	0.0584
CCC-F1	4	1	ND	WSUM	0.2123
JJJ-Prec.	12	1	AND	WSUM	0.2181
JJJ-Recall	20	20	ND	WSUM	0.0595
JJJ-F1	4	20	ND	WSUM	0.0850
JJC-Prec.	12	1	AND	WSUM	0.2184
JJC-Recall	20	20	ND	WSUM	0.0595
JJC-F1	4	20	ND	WSUM	0.0852
JJS-Prec.	12	1	AND	WSUM	**0.2252**
JJS-Recall	20	20	ND	WSUM	0.0662
JJS-F1	4	20	ND	WSUM	0.0917

methods in terms of precision and specificity. This indicates that this method is able to make true predictions for positive and negative relations; i.e. its tp and tn values are high. However, it has low recall, indicating that it cannot predict all true drug-disease relations. Other methods have higher recall and AUC-ROC values. This reflects that those methods were able to predict many drug-disease relations, but they also listed many false relations (since their precision is lower).

Table 6.3: Comparison of the proposed method to other state of the art methods from the literature

Type	Prec.	Recall	F1	SPC	AUC
Li and Lu [69]	-	**0.7700**	-	0.9200	0.8880
Chiang and Butte [20]	-	0.7400	-	0.8500	-
Zhang et al. [145]	0.3452	0.6505	**0.4510**	-	**0.8949**
Proposed Method - JJS	**0.4864**	0.0846	0.1442	**0.9995**	0.5421
Proposed Method - CCC	0.1894	0.4017	0.2575	0.9902	0.6960

The golden data we use is very skewed and very sparse with %99.44 sparse; i.e. there are many diseases that are irrelevant to the target drug. We would argue that precision is more important than recall for this data-set and for the drug re-positioning problem in general; i.e. making the right prediction for drug-disease relations is more important than finding all the relations. Comparing our method to other state of the art methods from the literature shows that the method proposed in this study can achieve higher precision, i.e. when it predicts a drug-disease relation, nearly half of the predictions are true.

6.4 Conclusion

Another application area that recommendation methods can be applied is drug repositioning. Drug repositioning can be defined as identifying new indications for known drugs [69]. It is a good alternative to traditional drug discovery approaches, since drug re-positioning can reduce the risks,cost and required time to identify new drugs.

In this chapter, we adapted our proposed multi-objective optimization based recommendation method to drug re-positioning. For this purpose, the most similar drugs to the target drug are identified and these neighbor drugs are then used to predict new indication of the target drug. Also, we applied several different settings that effect the calculations and compared their performance to each other. The experimental results indicated that the proposed method is able to achieve high precision performance and nearly half of its predictions are true. Comparison to the methods in the literature showed that the proposed method is better at making the right predictions. In general, the results demonstrated that use of recommendation method for drug re-positioning problem is promising.

CHAPTER 7

CONCLUSION AND FUTURE WORK

We summarize the ideas and findings explained in thesis in Section 7.1 and give ideas about future work in Section 7.2

7.1 Summary and Conclusion

In this thesis, four subjects are studied in general: 1) Use of multiple criteria from a single source to make recommendations, 2) use of multiple criteria from multiple sources to make recommendations, 3) use of recommendation methods to predict gene regularity networks and 4) use of recommendation methods to identify new indications for known drugs.

Firstly, we proposed a new multi-objective optimization based recommendation method that makes point of interest recommendations to target users. In this method, several different criteria, such as past preferences, location, social network and time, are combined together. We also expanded the proposed method by inferring the home/center location of the users and including travel locality idea, by making dynamic recommendations based on users' temporal preferences and by grouping users in terms of their home-town or their friendship relation. The evaluation results showed that combining multiple criteria with the help of multi-objective optimization approach leads to increase in coverage while preserving precision. We also observed that using smaller radius while using the travel locality idea, such that making recommendation in closer distances, provides better performance in terms of precision, but it cannot provide recommendations to all of the users. When we make dynamic recommen-

dations based on users' temporal preferences, we observed that the recommendation process becomes more challenging, since data becomes sparser. In terms of clustering, we found that clustering of users in the recommendation process leads to decrease in accuracy and coverage performance. However, it helps to deal with the scalability issue by making the methods to use only a subset of users.

Secondly, we aimed to combine multiple criteria from multiple data sources. In this thesis, we integrated information collected from multiple different social networking platforms to create an integrated model of individuals and to make recommendations to them. For this purpose, we collected data from BlogCatalog, Twitter, Flickr, Facebook, YouTube and LastFm web-sites and created two different data-sets. We used the created data-sets to make recommendations to target users on different platforms, i.e. recommending to Flickr users new groups to follow and recommending to BlogCatalog users other users to follow. We implemented several different types of recommendation methodologies to observe their performance. We compared the performance of these recommendation methodologies while using single versus multiple features from a single versus multiple sources. The conducted experiments showed that using multiple features from multiple sources improved the recommendation performance.

Thirdly, observing the common features of recommendation systems and GRNs, we applied the proposed multi-objective optimization based recommendation method to predict the gene relationships; such that which gene regulates the others. For the purpose of GRN re-construction, we used target genes and we predicted the genes that are regulated by the target gene. For the evaluation we used several different data-sets. The results showed that using information from multiple sources improves the performance. Also, we observed that use of an approach from recommendation systems performs well.

Lastly, we adapted our proposed multi-objective optimization based recommendation method to drug re-positioning problem which is defined as identification of new indications for known drugs. For this purpose, the most similar drugs to the target drug are identified and these neighbor drugs are then used to predict new indication of the target drug. The experimental results indicated that the proposed method is able to

achieve high precision performance and nearly half of its predictions are true. The results also demonstrated that using recommendation method on drug re-positioning problem is promising.

7.2 Future Work

The four subjects that are studied in this thesis have several different dimensions that can be studied further.

For the first subject, use of multiple criteria from a single source to make recommendations, several different ideas can be further studied: In the future we want to work on several other features available in LBSNs and add them to the recommendation process. For example, age group, gender, marital status and affiliation of the users may affect their check-in preferences. The information may also be used to give dynamic and group recommendations. Also, we want to study more on the noisy or unreliable data; such that we want to improve the performance of home/center location inference by introducing personal distance thresholds. Another idea that we want to work more on is to use different criteria for different users. We observed that there is no single combination of criteria that performs well for all the users. We conjecture that based on the user-profiles; such as having many friends/not, having many check-ins/not; different set of criteria can be used for each user and better recommendations can be made.

For the second subject, use of multiple criteria from multiple sources to make recommendations, we want to integrate identity resolution methods into our work and produce an end-to-end recommendation system. We also want to use more features to be captured from other social networking platforms not covered in this thesis. Finally, we want to try some other recommendation methods to observe their effectiveness while using a multi-source data-set.

For the last two subjects, use of recommendation methods to predict gene regularity network and to identify new indications for known drugs, we want to use other known recommendation methods. Also we want to apply our proposed method on other data-sets to observe its performance in depth.

REFERENCES

[1] G. Adomavicius and Y. Kwon. New recommendation techniques for multicriteria rating systems. *IEEE Intelligent Systems*, 22(3):48–55, May 2007.

[2] G. Adomavicius, N. Manouselis, and Y. Kwon. Multi-criteria recommender systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 769–803. Springer US, 2011.

[3] G. Adomavicius and A. Tuzhilin. Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749, 2005.

[4] I. S. Altingovde, O. N. Subakan, and O. Ulusoy. Cluster searching strategies for collaborative recommendation systems. *Inf. Process. Manage.*, 49(3):688–697, May 2013.

[5] L. Backstrom, E. Sun, and C. Marlow. Find me if you can: Improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 61–70, New York, NY, USA, 2010. ACM.

[6] J. Bao, D. Lian, F. Zhang, and N. J. Yuan. Geo-social media data analytic for user modeling and location-based services. *SIGSPATIAL Special*, 7(3):11–18, 2015.

[7] J. Bao, Y. Zheng, and M. F. Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '12, pages 199–208, New York, NY, USA, 2012. ACM.

[8] A. Bellogín, I. Cantador, F. Díez, P. Castells, and E. Chavarriaga. An empirical comparison of social, collaborative filtering, and hybrid recommenders. *ACM Trans. Intell. Syst. Technol.*, 4(1):14:1–14:29, Feb. 2013.

[9] A. Bernard, A. J. Hartemink, et al. Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data. In *Pacific symposium on biocomputing*, volume 10, pages 459–470. World Scientific, 2005.

[10] P. Bhargava, T. Phan, J. Zhou, and J. Lee. Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization

on sparse user-generated data. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 130–140, 2015.

[11] R. D. Burke. Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact.*, 12(4):331–370, 2002.

[12] M. Campillos, M. Kuhn, A.-C. Gavin, L. J. Jensen, and P. Bork. Drug target identification using side-effect similarity. *Science*, 321(5886):263–266, 2008.

[13] Á. Castellanos, E. W. De Luca, A. García-Serrano, and J. Cigarrán. Time, place and environment: Can conceptual modelling improve context-aware recommendation? In *Workshop on Context-awareness in Retrieval and Recommendation(CaRR) (To be published)*, 2015.

[14] D. I. S. N. Challenge. Dream4 in silico network challenge, 2015 (accessed 2015-04-30). `https://www.synapse.org/#!Synapse:syn3049712/wiki/`.

[15] E.-. Challenge. Eswc-14 challenge: Linked open data-enabled recommender systems, March 2014.

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

[17] D. Cheng, C. Knox, N. Young, P. Stothard, S. Damaraju, and D. S. Wishart. Polysearch: a web-based text mining system for extracting relationships between human diseases, genes, mutations, drugs and metabolites. *Nucleic acids research*, 36(suppl 2):W399–W405, 2008.

[18] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: A content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 759–768, New York, NY, USA, 2010. ACM.

[19] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui. Exploring millions of footprints in location sharing services. In L. A. Adamic, R. A. Baeza-Yates, and S. Counts, editors, *ICWSM*. The AAAI Press, 2011.

[20] A. P. Chiang and A. J. Butte. Systematic evaluation of drug-disease relationships to identify leads for novel drug uses. *Clinical pharmacology and therapeutics*, 86(5):507, 2009.

[21] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM*

SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11, pages 1082–1090, New York, NY, USA, 2011. ACM.

[22] P. Csermely, T. Korcsmáros, H. J. Kiss, G. London, and R. Nussinov. Structure and dynamics of molecular networks: a novel paradigm of drug discovery: a comprehensive review. *Pharmacology & therapeutics*, 138(3):333–408, 2013.

[23] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[24] C. A. Davis Jr, G. L. Pappa, D. R. R. de Oliveira, and F. de L Arcanjo. Inferring the location of twitter messages based on user relationships. *Transactions in GIS*, 15(6):735–751, 2011.

[25] T. Di Noia, I. Cantador, and V. Ostuni. Linked open data-enabled recommender systems: Eswc 2014 challenge on book recommendation. In V. Presutti, M. Stankovic, E. Cambria, I. Cantador, A. Di Iorio, T. Di Noia, C. Lange, D. Reforgiato Recupero, and A. Tordai, editors, *Semantic Web Evaluation Challenge*, volume 475 of *Communications in Computer and Information Science*, pages 129–143. Springer International Publishing, 2014.

[26] DrugBank. Drugbank, 2016.

[27] J. Dudley, T. Deshpande, and A. J. Butte. Exploiting drug-disease relationships for computational drug repositioning. *Briefings in Bioinformatics*, 12(4):303–311, 2011.

[28] Facebook. Facebook:about, 2016 (accessed 2015-05-02). `https://newsroom.fb.com/company-info/`.

[29] Foursquare. Foursquare:about, 2016 (accessed 2016-05-02). `https://foursquare.com/about`.

[30] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.

[31] H. Gao and H. Liu. Location-based social network data repository, 2014. `http://www.public.asu.edu/~hgao16/dataset.html`.

[32] H. Gao, J. Tang, X. Hu, and H. Liu. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM conference on Recommender systems*, RecSys '13, pages 93–100. ACM, 2013.

[33] H. Gao, J. Tang, X. Hu, and H. Liu. Modeling temporal effects of human mobile behavior on location-based social networks. In *Proceedings of the 22Nd*

ACM International Conference on Conference on Information & Knowledge Management, CIKM '13, pages 1673–1678, New York, NY, USA, 2013. ACM.

[34] H. Gao, J. Tang, and H. Liu. Exploring social-historical ties on location-based social networks. In J. G. Breslin, N. B. Ellison, J. G. Shanahan, and Z. Tufekci, editors, *ICWSM*. The AAAI Press, 2012.

[35] H. Gao, J. Tang, and H. Liu. Addressing the cold-start problem in location recommendation using geo-social correlations. *Data Mining and Knowledge Discovery*, pages 1–25, 2014.

[36] J. Gebert, N. Radde, and G.-W. Weber. Modeling gene regulatory networks with piecewise linear differential equations. *European Journal of Operational Research*, 181(3):1148 – 1165, 2007.

[37] GeneMANIA. Genemania, 2015.

[38] V. Gligorijević and N. Pržulj. Methods for biological data integration: perspectives and challenges. *Journal of The Royal Society Interface*, 12(112), 2015.

[39] M. Göksedef and c. Gündüz-Öğüdücü. Combination of web page recommender systems. *Expert Syst. Appl.*, 37(4):2911–2922, Apr. 2010.

[40] A. Gottlieb, G. Y. Stein, E. Ruppin, and R. Sharan. Predict: a method for inferring novel drug indications with application to personalized medicine. *Molecular systems biology*, 7(1):496, 2011.

[41] M. Grzegorczyk and D. Husmeier. Improving the structure mcmc sampler for bayesian networks by introducing a new edge reversal move. *Machine Learning*, 71(2-3):265–305, 2008.

[42] Haversine_Distance, 2015 (accessed 2014-12-08). `http://andrew.hedges.name/experiments/haversine/`.

[43] J. He, W. W. Chu, and Z. V. Liu. Inferring privacy information from social networks. In *Proceedings of the 4th IEEE International Conference on Intelligence and Security Informatics*, ISI'06, pages 154–165, Berlin, Heidelberg, 2006. Springer-Verlag.

[44] J. He, X. Li, L. Liao, D. Song, and W. K. Cheung. Inferring a personalized next point-of-interest recommendation model with latent behavior patterns. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 137–143, 2016.

[45] B. Hecht, L. Hong, B. Suh, and E. H. Chi. Tweets from justin bieber's heart: The dynamics of the location field in user profiles. In *Proceedings of the*

SIGCHI Conference on Human Factors in Computing Systems, CHI '11, pages 237–246, New York, NY, USA, 2011. ACM.

[46] M. Hecker, S. Lambeck, S. Toepfer, E. van Someren, and R. Guthke. Gene regulatory network inference: Data integration in dynamic models—a review. *Biosystems*, 96(1):86 – 103, 2009.

[47] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, Jan. 2004.

[48] B. Hu, M. Jamali, and M. Ester. Spatio-temporal topic modeling in mobile social media for location recommendation. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, ICDM '13, pages 1073–1078. IEEE, 2013.

[49] G. Hu and P. Agarwal. Human disease-drug network based on genomic expression profiles. *PloS one*, 4(8):e6536, 2009.

[50] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and C. Zhu. Personalized recommendation via cross-domain triadic factorization. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 595–606, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[51] T. Ideker, J. Dutkowski, and L. Hood. Boosting signal-to-noise in complex biology: prior knowledge is power. *Cell*, 144(6):860–863, 2011.

[52] F. Iorio, R. Bosotti, E. Scacheri, V. Belcastro, P. Mithbaokar, R. Ferriero, L. Murino, R. Tagliaferri, N. Brunetti-Pierri, A. Isacchi, et al. Discovery of drug mode of action and drug repositioning from transcriptional responses. *Proceedings of the National Academy of Sciences*, 107(33):14621–14626, 2010.

[53] P. Jain, P. Kumaraguru, and A. Joshi. @i seek 'fb.me': Identifying users across multiple online social networks. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 1259–1268, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[54] T. Jambor and J. Wang. Optimizing multiple objectives in collaborative filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 55–62, New York, NY, USA, 2010. ACM.

[55] Y. Jia, Y. Wang, X. Jin, and X. Cheng. Tsbm: The temporal-spatial bayesian model for location prediction in social networks. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, volume 2 of *WI-IAT '14*, pages 194–201. IEEE, 2014.

[56] P. Kazienko and P. Kolodziejski. Personalized integration of recommendation methods for e-commerce. *Journal of Computer Science and Applications*, 3(3):12–26, 2006.

[57] M. J. Keiser, V. Setola, J. J. Irwin, C. Laggner, A. I. Abbas, S. J. Hufeisen, N. H. Jensen, M. B. Kuijer, R. C. Matos, T. B. Tran, et al. Predicting new molecular targets for known drugs. *Nature*, 462(7270):175–181, 2009.

[58] E. Kotelnikova, A. Yuryev, I. Mazo, and N. Daraselia. Computational approaches for drug repositioning and combination therapy design. *Journal of bioinformatics and computational biology*, 8(03):593–606, 2010.

[59] M. Kuhn, M. Campillos, I. Letunic, L. J. Jensen, and P. Bork. A side effect resource to capture phenotypic effects of drugs. *Molecular systems biology*, 6(1), 2010.

[60] A. Kumar, N. Kumar, M. Hussain, S. Chaudhury, and S. Agarwal. Semantic clustering-based cross-domain recommendation. In *Computational Intelligence and Data Mining (CIDM), 2014 IEEE Symposium on*, pages 137–141, Dec 2014.

[61] Y.-K. Kwon and K.-H. Cho. Analysis of feedback loops and robustness in network evolution based on boolean models. *BMC Bioinformatics*, 8(1), 2007.

[62] K. Lakiotaki, N. F. Matsatsinis, and A. Tsoukias. Multicriteria user modeling in recommender systems. *IEEE Intelligent Systems*, 26(2):64–76, Mar. 2011.

[63] K. Lakiotaki, S. Tsafarakis, and N. Matsatsinis. Uta-rec: A recommender system based on multiple criteria analysis. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 219–226, New York, NY, USA, 2008. ACM.

[64] LargestCities, 2014 (accessed 2014-12-08). `http://www.citymayors.com/statistics/largest-cities-density-125.html`.

[65] H.-H. Lee and W.-G. Teng. Incorporating multi-criteria ratings in recommendation systems. In *IRI*, pages 273–278. IEEE Systems, Man, and Cybernetics Society, 2007.

[66] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, et al. Transcriptional regulatory networks in saccharomyces cerevisiae. *science*, 298(5594):799–804, 2002.

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

[68] C.-Y. Li and S.-D. Lin. Matching users and items across domains to improve the recommendation quality. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 801–810, New York, NY, USA, 2014. ACM.

[69] J. Li and Z. Lu. A new method for computational drug repositioning using drug pairwise similarity. *2013 IEEE International Conference on Bioinformatics and Biomedicine*, 0:1–4, 2012.

[70] J. Li and X.-S. Zhang. An optimization model for gene regulatory network reconstruction with known biological information. *Optimization and systems biology. Lecture notes in operations research*, 7:35–44, 2007.

[71] J. Li, X. Zhu, and J. Y. Chen. Building disease-specific drug-protein connectivity maps from molecular interaction networks and pubmed abstracts. *PLoS Comput Biol*, 5(7):e1000450, 2009.

[72] D. Lian, X. Xie, V. W. Zheng, N. J. Yuan, F. Zhang, and E. Chen. Cepr: A collaborative exploration and periodically returning model for location prediction. *ACM Trans. Intell. Syst. Technol.*, 6(1):8:1–8:27, Apr. 2015.

[73] S. Liang, S. Fuhrman, and R. Somogyi. REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In *Pacific Symposium on Biocomputing*, volume 3, pages 18–29, 1998.

[74] M. D. Lieberman and J. Lin. You are where you edit: Locating wikipedia contributors through edit histories. In *ICWSM*, 2009.

[75] K. H. Lim. Recommending tours and places-of-interest based on user interests from geo-tagged photos. In *Proceedings of the 2015 ACM SIGMOD on PhD Symposium*, pages 33–38. ACM, 2015.

[76] J. Lindamood, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. Inferring private information using social network data. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 1145–1146, New York, NY, USA, 2009. ACM.

[77] H. Liu and P. Maes. Interestmap: Harvesting social network profiles for recommendations. *Beyond Personalization 2005*, page 54, 2005.

[78] X. Liu and K. Aberer. Soco: A social network aided context-aware recommender system. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 781–802, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[79] X. Liu, Y. Liu, K. Aberer, and C. Miao. Personalized point-of-interest recommendation by mining users' preference transition. In *Proceedings of the*

22Nd ACM International Conference on Conference on Information Knowledge Management, CIKM '13, pages 733–738, New York, NY, USA, 2013. ACM.

[80] B. Loni, Y. Shi, M. Larson, and A. Hanjalic. Cross-domain collaborative filtering with factorization machines. In *Advances in Information Retrieval - 36th European Conference on IRResearch, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings*, pages 656–661, 2014.

[81] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 287–296, New York, NY, USA, 2011. ACM.

[82] J. Mahmud, J. Nichols, and C. Drews. Where is this tweet from? inferring home locations of twitter users. In *ICWSM'12*, 2012.

[83] M. Maleszka, B. Mianowska, and N. T. Nguyen. A method for collaborative recommendation using knowledge integration tools and hierarchical structure of user profiles. *Knowledge-Based Systems*, 47:1–13, 2013.

[84] N. Manouselis and C. Costopoulou. Analysis and classification of multicriteria recommender systems. *World Wide Web*, 10(4):415–441, Dec. 2007.

[85] N. Manouselis and C. Costopoulou. Experimental analysis of design choices in multiattribute utility collaborative filtering. *IJPRAI*, 21(2):311–331, 2007.

[86] D. Marbach, R. J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, and G. Stolovitzky. Revealing strengths and weaknesses of methods for gene network inference. *PNAS*, 107(14):6286–6291, 2010. WingX.

[87] D. Marbach, T. Schaffter, C. Mattiussi, and D. Floreano. Generating Realistic In Silico Gene Networks for Performance Assessment of Reverse Engineering Methods. *Journal of Computational Biology*, 16(2):229–239, 2009. WingX.

[88] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. D. Favera, and A. Califano. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC bioinformatics*, 7(Suppl 1):S7, 2006.

[89] P. Massa and P. Avesani. Trust-aware recommender systems. In *RecSys*, pages 17–24, 2007.

[90] S. Memoizer. Sequence memoizer, 2015 (accessed 2015-06-01). `http://www.sequencememoizer.com/`.

[91] G. Michailidis and F. d'Alché Buc. Autoregressive models for gene regulatory network inference: Sparsity, stability and causality issues. *Mathematical biosciences*, 246(2):326–334, 2013.

[92] M. Motoyama and G. Varghese. I seek you: Searching and matching individuals in social networks. In *Proceedings of the Eleventh International Workshop on Web Information and Data Management*, WIDM '09, pages 67–75, New York, NY, USA, 2009. ACM.

[93] S. Naseri, A. Bahrehmand, C. Ding, and C.-H. Chi. Enhancing tag-based collaborative filtering via integrated social networking information. In *ASONAM*, 2013.

[94] M. Nicolau and M. Schoenauer. On the evolution of scale-free topologies with a gene regulatory network model. *Biosystems*, 98(3):137 – 148, 2009. Evolving Gene Regulatory Networks.

[95] T. Noeske, B. C. Sasse, H. Stark, C. G. Parsons, T. Weil, and G. Schneider. Predicting compound selectivity by self-organizing maps: Cross-activities of metabotropic glutamate receptor antagonists. *ChemMedChem*, 1(10):1066–1068, 2006.

[96] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. Mining user mobility features for next place prediction in location-based services. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, ICDM '12, pages 1038–1043, Washington, DC, USA, 2012. IEEE Computer Society.

[97] F. Ortega, J.-L. SáNchez, J. Bobadilla, and A. GutiéRrez. Improving collaborative filtering-based recommender systems results using pareto dominance. *Information Science*, 239:50–61, Aug. 2013.

[98] A. Özgür, T. Vu, G. Erkan, and D. R. Radev. Identifying gene-disease associations using centrality on a literature mined gene-interaction network. *Bioinformatics*, 24(13):i277–i285, 2008.

[99] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, Brisbane, Australia, 1998.

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

[101] O. Phelan, K. McCarthy, and B. Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 385–388, New York, NY, USA, 2009. ACM.

[102] T. Pontes, G. Magno, M. Vasconcelos, A. Gupta, J. Almeida, P. Kumaraguru, and V. Almeida. Beware of what you share: Inferring home location in social networks. In *Proceedings of the 2012 IEEE 12th International Conference*

on Data Mining Workshops, ICDMW '12, pages 571–578, Washington, DC, USA, 2012. IEEE Computer Society.

[103] T. Pontes, M. Vasconcelos, J. Almeida, P. Kumaraguru, and V. Almeida. We know where you live: Privacy characterization of foursquare behavior. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 898–905, New York, NY, USA, 2012. ACM.

[104] A. Popescu, G. Grefenstette, et al. Mining user home location and gender from flickr tags. In *ICWSM*, Washington, United States, 2010.

[105] R. J. Prill, D. Marbach, J. Saez-Rodriguez, P. K. Sorger, L. G. Alexopoulos, X. Xue, N. D. Clarke, G. Altan-Bonnet, and G. Stolovitzky. Towards a rigorous assessment of systems biology models: The dream3 challenges. *PLoS ONE*, 5(2):e9202, 02 2010.

[106] A. Qabaja, M. Alshalalfa, E. Alanazi, and R. Alhajj. Prediction of novel drug indications using network driven biological data prioritization and integration. *Journal of cheminformatics*, 6(1):1–14, 2014.

[107] D. I. S. N. C. T. Rankings. Dream4 in silico network challenge team rankings, 2015 (accessed 2015-04-30). `http://wiki.c2b2.columbia.edu/dream/results/DREAM4/`.

[108] B. Ristevski. A survey of models for inference of gene regulatory networks. *Nonlinear Anal Model Control*, 18:444–465, 2013.

[109] M. Rodriguez, C. Posse, and E. Zhang. Multiple objective optimization in recommender systems. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 11–18, New York, NY, USA, 2012. ACM.

[110] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 851–860, New York, NY, USA, 2010. ACM.

[111] B. M. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT)*, 2002.

[112] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl. Application of dimensionality reduction in recommender systems: A case study. In *WebKDD Workshop at the ACM SIGKKD*, 2000.

[113] M. Sarwat, J. J. Levandoski, A. Eldawy, and M. F. Mokbel. Lars*: An efficient and scalable location-aware recommender system. *IEEE Transactions on Knowledge and Data Engineering*, 99(PrePrints):1, 2013.

[114] M. Sattari. Hybrid geo-activity recommendation system using advanced feature combination and semantic activity similarity. Master's thesis, Middle East Technical University, Ankara, Turkey, 2013.

[115] J. Schafer and K. Strimmer. Learning large-scale graphical gaussian models from genomic data. *Science of Complex Networks From Biology to the Internet and WWW*, 776:263–276, 2005.

[116] A. Shermin and M. A. Orgun. Using dynamic bayesian networks to infer gene regulatory networks from expression profiles. In *Proceedings of the 2009 ACM Symposium on Applied Computing*, SAC '09, pages 799–803, New York, NY, USA, 2009. ACM.

[117] I. Shmulevich, E. R. Dougherty, and W. Zhang. From boolean to probabilistic boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE*, 90(11):1778–1792, 2002.

[118] M. Sirota, J. T. Dudley, J. Kim, A. P. Chiang, A. A. Morgan, A. Sweet-Cordero, J. Sage, and A. J. Butte. Discovery and preclinical validation of drug indications using compendia of public gene expression data. *Science translational medicine*, 3(96):96ra77–96ra77, 2011.

[119] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular biology of the cell*, 9(12):3273–3297, 1998.

[120] M. Tan, M. Alshalalfa, R. Alhajj, and F. Polat. Combining multiple types of biological data in constraint-based learning of gene regulatory networks. In *Computational Intelligence in Bioinformatics and Computational Biology, 2008. CIBCB '08. IEEE Symposium on*, pages 90–97, Sept 2008.

[121] S. Tan, J. Bu, X. Qin, C. Chen, and D. Cai. Cross domain recommendation based on multi-type media fusion. *Neurocomput.*, 127:124–134, Mar. 2014.

[122] S. Tan, Z. Guan, D. Cai, X. Qin, J. Bu, and C. Chen. Mapping users across networks by manifold alignment on hypergraph. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 159–165, 2014.

[123] L. Tang, Y. Jiang, L. Li, and T. Li. Ensemble contextual bandits for personalized recommendation. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 73–80, New York, NY, USA, 2014. ACM.

[124] M. Tavakolifard and K. C. Almeroth. Social computing: an intersection of recommender systems, trust/reputation systems, and social networks. *IEEE Network*, 26(4):53–58, 2012.

[125] C.-Y. Tsai and B.-H. Lai. A location-item-time sequential pattern mining algorithm for route recommendation. *Knowledge-Based Systems*, 73:97–110, 2015.

[126] Twitter. Twitter:about, 2016 (accessed 2015-05-02). `https://about.twitter.com/company`.

[127] Y. Wang, J. Xiao, T. O. Suzek, J. Zhang, J. Wang, and S. H. Bryant. Pubchem: a public information system for analyzing bioactivities of small molecules. *Nucleic acids research*, page gkp456, 2009.

[128] Y. R. Wang, K. Jiang, L. J. Feldman, P. J. Bickel, and H. Huang. Inferring gene association networks using sparse canonical correlation analysis. *arXiv preprint arXiv:1401.6504*, 2014.

[129] L. F. Wessels, E. P. van Someren, M. J. Reinders, et al. A comparison of genetic network models. In *pacific Symposium on Biocomputing*, volume 6, pages 508–519, 2001.

[130] Wikipedia_Haversine_Distance, 2014 (accessed 2014-12-08). `http://en.wikipedia.org/wiki/Haversine_formula/`.

[131] P. Winoto and T. Tang. If you like the devil wears prada the book, will you also enjoy the devil wears prada the movie? a study of cross-domain recommendations. *New Generation Computing*, 26(3):209–225, 2008.

[132] D. S. Wishart, C. Knox, A. C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam, and M. Hassanali. Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic acids research*, 36(suppl 1):D901–D906, 2008.

[133] L. Yang and P. Agarwal. Systematic drug repositioning based on clinical side-effects. *PloS one*, 6(12):e28025, 2011.

[134] S. Yardi and D. Boyd. Tweeting from the town square: Measuring geographic local networks. In *International Conference on Weblogs and Social Media*. American Association for Artificial Intelligence, 2010.

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

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

[137] H.-F. Yu, C.-J. Hsieh, S. Si, and I. S. Dhillon. Scalable coordinate descent approaches to parallel matrix factorization for recommender systems. In *IEEE International Conference of Data Mining*, 2012.

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

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

[140] R. Zafarani and H. Liu. Connecting users across social media sites: A behavioral-modeling approach. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 41–49, New York, NY, USA, 2013. ACM.

[141] J. Zhang and C. Chow. Point-of-interest recommendations in location-based social networks. *SIGSPATIAL Special*, 7(3):26–33, 2015.

[142] J. Zhang and C. Chow. Ticrec: A probabilistic framework to utilize temporal influence correlations for time-aware location recommendations (to be published). *Services Computing, IEEE Transactions on*, PP(99):1–1, 2015.

[143] J. Zhang and C. Chow. Ticrec: A probabilistic framework to utilize temporal influence correlations for time-aware location recommendations (to be published). *Services Computing, IEEE Transactions on*, PP(99):1–1, 2015.

[144] J.-D. Zhang and C.-Y. Chow. igslr: Personalized geo-social location recommendation: A kernel density estimation approach. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'13, pages 334–343, New York, NY, USA, 2013. ACM.

[145] P. Zhang, P. Agarwal, and Z. Obradovic. Computational drug repositioning by ranking and integrating multiple data sources. In H. Blockeel, K. Kersting, S. Nijssen, and F. Zelezný, editors, *ECML/PKDD (3)*, volume 8190 of *Lecture Notes in Computer Science*, pages 579–594. Springer, 2013.

[146] W. Zhang and J. Wang. Location and time aware social collaborative retrieval for new successive point-of-interest recommendation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1221–1230, New York, NY, USA, 2015. ACM.

[147] W. Zhang, J. Zang, X. Jing, Z. Sun, W. Yan, D. Yang, F. Guo, and B. Shen. Identification of candidate mirna biomarkers from mirna regulatory network with application to prostate cancer. *J Transl Med*, 12(66):28, 2014.

[148] Y. Zhang. Browser-oriented universal cross-site recommendation and explanation based on user browsing logs. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 433–436, New York, NY, USA, 2014. ACM.

[149] Y. Zhang, Z. Deng, H. Jiang, and P. Jia. Inferring gene regulatory networks from multiple data sources via a dynamic bayesian network with structural em. In S. Cohen-Boulakia and V. Tannen, editors, *Data Integration in the Life Sciences*, volume 4544 of *Lecture Notes in Computer Science*, pages 204–214. Springer Berlin Heidelberg, 2007.

[150] S. Zhao, T. Zhao, H. Yang, M. R. Lyu, and I. King. STELLAR: spatial-temporal latent ranking for successive point-of-interest recommendation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 315–322, 2016.

[151] E. Zheleva and L. Getoor. To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 531–540, New York, NY, USA, 2009. ACM.

[152] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 1029–1038, New York, NY, USA, 2010. ACM.

[153] Y. Zheng, B. Mobasher, and R. Burke. *User Modeling, Adaptation and Personalization: 23rd International Conference, UMAP 2015, Dublin, Ireland, June 29 – July 3, 2015. Proceedings*, chapter Integrating Context Similarity with Sparse Linear Recommendation Model, pages 370–376. Springer International Publishing, Cham, 2015.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Özsoy, Makbule Gülçin
Nationality: Turkish (TC)
Date and Place of Birth: 08.04.1985, Kayseri

EDUCATION

Degree	Institution	Year of Grad.
M.S.	Dept. of Comp. Eng., METU, Ankara, Turkey	2011
B.S.	Dept. of Comp. Eng., METU, Ankara, Turkey	2008
High School	Kayseri Fen Lisesi, Kayseri, Turkey	2003

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
June 2012- Cont.	T2 Yazılım	Software Engineer
September 2009 - June 2012	TUBITAK, Ankara, Turkey	Researcher
June 2008 - September 2009	HAVELSAN, Ankara, Turkey	Software Engineer

PUBLICATIONS

International Conference Publications

Makbule Gulcin Ozsoy, Faruk Polat, Reda Alhajj: Inference of gene regulatory networks via multiple data sources and a recommendation method. BIBM 2015: 661-664 (2015).

Makbule Gulcin Ozsoy, Faruk Polat, Reda Alhajj: Modeling Individuals and Making Recommendations Using Multiple Social Networks. ASONAM 2015: 184-1191 (2015).

Makbule Gulcin Ozsoy, Kezban Dilek Onal, Ismail Sengor Altingovde: Result Diversification for Tweet Search. WISE 2014: 78-89 (2014).

Makbule Gulcin Ozsoy, Faruk Polat, Reda Alhajj: Multi-Objective Optimization Based Location and Social Network Aware Recommendation. COLLABORATECOM 2014: 233-242 (2014).

Makbule Gulcin Ozsoy, Ruket Cakici: Contrastive Max-Sum Opinion Summarization. AIRS 2014: 256-267 (2014).

Makbule Gulcin Ozsoy, Faruk Polat: Trust Based Recommendation Systems. MASSN 2013: 1267-1274 (2013).

Makbule Gulcin Ozsoy, Ilyas Cicekli, Ferda Nur Alpaslan: Text Summarization of Turkish Texts using Latent Semantic Analysis. COLING 2010: 869-876. (2010)

International Journal Publications

Makbule Gulcin Ozsoy, Ferda Nur Alpaslan, Ilyas Cicekli: Text Summarization using Latent Semantic Analysis. Journal of Information Science 37(4): 405-417 (2011).