Personalized Location Selection in Large-scale Geo-social Networks

Nur Al Hasan Haldar

This thesis is presented for the degree of
Doctor of Philosophy of The University of Western Australia

School of Physics, Mathematics, and Computing
Department of Computer Science and Software Engineering

Supervisors: Mark Reynolds (Coordinating), Ajmal Mian,
Jianxin Li (Deakin University), Timos Sellis (Swinburne University of Technology)

Thesis Declaration

I, NUR AL HASAN HALDAR, certify that:

This thesis has been substantially accomplished during enrolment in the degree.

This thesis does not contain material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution.

No part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of The University of Western Australia and where applicable, any partner institution responsible for the joint-award of this degree.

This thesis does not contain any material previously published or written by another person, except where due reference has been made in the text.

The work(s) are not in any way a violation or infringement of any copyright, trademark, patent, or other rights whatsoever of any person.

The work described in this thesis was funded by ARC Discovery Projects under Grant No. DP160102114 and CSIRO Data61 Scholarship Program.

This thesis contains published work and/or work prepared for publication, some of which has been co-authored.

Signature: [Nur Al Hasan Haldar]

Abstract

Nowadays, the social media platform is used not only to share life events, but it is extensively used as a convenient way to communicate with friends, promote businesses, organizing events, etc. With such a variety of advantages, social media provides tremendous opportunities for analyzing the enormous volume of data generated in the networks. Apart from the social connectivity and other structural information available in the network, the location details associated with the social users are important to enhance the performances in various real-time applications ranging from news recommendation systems to disaster management. This strongly motivates the research areas to effectively utilize the socio-spatial information of a network to support the applications where location information of users are crucial. The research on development of methods to select the personalized locations for social users effectively and track their evolution efficiently has been very limited so far. In this thesis, we aim to capture the latent relationships between the users and locations to personalize user preference. To this end, we propose several technical methods to effectively model the relationships between user mobility and social connections to support various online location-based applications.

Firstly, we investigate the problem of location prediction, which intends to infer locations of social users using the implicit information available in the networks. For this purpose, a considerable number of location prediction models have been proposed in the literature that exploit various network features. Unfortunately, these models have not been benchmarked on common datasets using standard metrics, which make it hard to summarize the insights of the existing models in different data settings. Therefore, in this thesis, we first propose a generalized procedure-oriented location prediction framework to do an in-depth empirical comparison of the representative models, and further perform a detailed analysis of the significant insights on the location prediction task. Based on the observations from the empirical study, we noticed that the state-of-the-art models in this research domain do not effectively exploit the latent relationships between the users and their locations. Hence, we develop a location inference model that can effectively propagate the spatial information of a network through the friendship edges maintaining an inference priority sequence. We consider the network-only features to achieve reasonable performance in minimal supervision scenarios, as often happens in real world datasets. We further analyze the socio-spatial characteristics of the check-in locations and formulate a new problem of identifying the top representative locations for the social users. To address the problem, we develop several solutions by deriving relaxed bounds and effective pruning rules, which can significantly reduce the computational cost. We show the selected locations of the social users can cover a large spatial space and their social connections, and also verify the performance of our proposed methods by comparing with several existing approaches on object selection.

In geo-social network, the locations visited by the members of highly cohesive user groups can carry strong intention of personalized common interests in different social groups. Such set of socio-spatially co-engaged locations can be useful for various applications such as location-based marketing, event organization, and user/location recommendation. Identifying the groups of co-engaged locations can help to discover interesting social user groups which are promising to become stable and active in

spatial region. Motivated by the significance of the socio-spatially co-engaged location set, this thesis studies the problem on identifying top location group that are highly relevant to various cohesive social user groups. Such location set can greatly enhanced users' togetherness and their experiences. These research topics mentioned in this thesis demonstrate the usefulness of our research to support the online location-based applications. We investigate the proposed problems using real-world social network data, present our application-specific experimental study, and show some interesting insights found in the study.

Acknowledgments

I am grateful to God Almighty with gratitude and humility for the wisdom and good health He bestowed upon me to complete my thesis.

This thesis would not have been possible without the support and guidance of some important people in my life. Firstly, I would like to thank my parents for their unconditional love and encouragement in every stages of my life. Specifically, I am indebted to my Mother, the first teacher and a great source of energy in my life. I also owe my profound gratitude to my wife for her moral support and understanding extended with love. She encouraged me to become mentally stronger and focused.

I would like to express my deepest appreciation to my esteemed supervisors Prof. Mark Reynolds, Dr. Jianxin Li, Prof. Ajmal Mian, Prof. Timos Sellis for their visionary guidance, insightful comments, encouragements and continuous support towards the completion of my Ph.D. thesis. They were a great source of motivation during the entire period of my candidature. Their guidance and positive feedback helped me to improve my research ability towards becoming an independent researcher in my field. I also acknowledge Dr. Quanxi Shao and Dr. Cecile Paris (Data61, CSIRO) for their invaluable comments and feedback on my research problems. I would like to thank the CSSE admin staffs Charise Baker and Hass for their administrative support and assistance.

I am thankful to Professor Mohammed Eunus Ali who has guided me unconditionally to explore critical researches in my field. He helped me to learn how to drive deep into a research problem, and to present profound research ideas clearly. His inspiration encourage me to move forward when I got stuck in my research. I also thank Dr. Farhana Choudhury for her feedback on the technical writings.

I would like to acknowledge Swinburne University of Technology, Melbourne for providing various facilities and support during my visit as a postgraduate student researcher. Special thanks to Prof. Timos Sellis for arranging my visit. I would also like to acknowledge the financial and travel support provided to me through UWA Faculty Scholarship for International Research Fees, Ad Hoc Postgraduate Scholarship, UWA Graduate Research School travel funding, and CSIRO Data 61 Ph.d. Scholarship.

AUTHORSHIP DECLARATION: CO-AUTHORED PUBLICATIONS

This thesis contains work that has been published and/or prepared for publication, in which the candidate is the first author and primary contributor:

- **Nur Al Hasan Haldar**, Jianxin Li, Mark Reynolds, Timos Sellis, "Location prediction in large-scale social networks: an in-depth benchmarking study," The VLDB Journal, 28(5), pp.623-648., 2019
  Location in thesis: Chapter 3
  Student contribution to work: 85%

- **Nur Al Hasan Haldar**, Mark Reynolds, Quanxi Shao, Jianxin Li, Yunliang Chen, "Geolocating Activity Location in Location-based Social Network", Under review at World Wide Web Journal, 2021
  Location in thesis: Chapter 4
  Student contribution to work: 85%

- **Nur Al Hasan Haldar**, Jianxin Li, Mohammed Eunus Ali, Taotao Cai, Timos Sellis, Mark Reynolds, "Top-k socio-spatial co-engaged location selection", Prepared for submission to PVLDB, 2021, arXiv preprint: https://arxiv.org/pdf/2009.00373.pdf
  Location in thesis: Chapter 5
  Student contribution to work: 85%

- **Nur Al Hasan Haldar**, Mohammed Eunus Ali, Jianxin Li, Mark Reynolds, "Co-engaged Location Group Search in Location-based Social Network", Prepared for submission to TKDE, 2021
  Location in thesis: Chapter 6
  Student contribution to work: 85%

Other published works as co-author (not included in the thesis).

- Taotao Cai, Jianxin Li, **Nur Al Hasan Haldar**, Ajmal Mian, John Yearwood, Timos Sellis, "Anchored Vertex Exploration for Community Engagement in Social Networks," IEEE 36th International Conference on Data Engineering (ICDE), pp. 409-420., 2020.

- Arun Chaitanya Mandalapu, Saranya Gunabalan, Avinash Sadineni, Taotao Cai, **Nur Al Hasan Haldar**, Jianxin Li, "Correlate Influential News Article Events to Stock Quote Movement," In International Conference on Advanced Data Mining and Applications, pp. 331-342. Springer, Cham, 2019.

Co-authors signature and dates:

Co-author 1: Mark Reynolds

Signature: Date: _____

Co-author 2: Jianxin Li

Signature: Date: _____

Co-author 3: Timos Sellis

Signature: Date: _____

Co-author 4: Taotao Cai

Signature: Date: _____

Co-author 5: Mohammed Eunus Ali

Signature: Date: _____

Co-author 6: Quanxi Shao

Signature: Date: _____

Co-author 7: Yunliang Chen

Signature: Date: _____

Student signature: Md Al Hasan Haldar Date: _____

I, Mark Reynolds, certify that the student statements regarding their contribution to each of the works listed above are correct.
Coordinating supervisor signature:
Date: _____

Contents

Contents vii

List of Figures xi

List of Tables xv

1 Introduction 1

1.1 Motivation . . . 2

1.1.1 Location Prediction in Social Networks . . . 2

1.1.2 Co-engaged Location Selection for Social Users . . . 3

1.1.3 Location Group Search in Social Media . . . 5

1.2 Contribution . . . 6

1.2.1 Location Prediction in Social Networks . . . 7

1.2.2 Co-engaged Location Selection for Social Users . . . 8

1.2.3 Location Group Search in Location-based Social Network . . . 9

1.3 Organization . . . 10

2 Literature Review 13

2.1 Location Prediction in Social Network . . . 13

2.1.1 Content based Location Prediction Approaches . . . 13

2.1.2 Network based Location Prediction Approaches . . . 15

2.1.3 Hybrid Models for Location Prediction . . . 16

2.1.4 Neural Network based Location Prediction Approaches . . . 17

2.1.5 Comparative Studies on Location Prediction Models . . . 18

2.2 Object Selection in Location-based Social Network . . . 20

2.2.1 Spatial Clustering and Sampling . . . 20

2.2.2 General Queries in LBSN . . . 21

2.2.3 Diversified Object Selection . . . 22

2.2.4 Spatial Object Selection and Map Generalization . . . 22

2.3 Group Queries in Social Network . . . 24

2.3.1 Group Queries based on Social connectivity . . . 24

2.3.2 Attribute-driven Group Queries . . . 25
2.3.3 Socio-spatial Group Queries . . . 26

3 Location Prediction in Large-scale Social Networks 29
3.1 Introduction . . . 30
3.1.1 Challenges . . . 31
3.1.2 Generalized Procedural Framework . . . 32
3.2 Preliminaries and Background . . . 33
3.2.1 Preliminary . . . 33
3.2.2 Prediction Models and Algorithms . . . 35
3.3 The Generalized Procedure . . . 37
3.3.1 Initialization Phase . . . 38
3.3.2 Model driven Prediction . . . 38
3.3.3 Validation Phase . . . 39
3.4 Within Framework Implementation . . . 39
3.4.1 Probabilistic Language Model . . . 39
3.4.2 Generative Influence based Model . . . 40
3.4.3 Generative Relationship based Model . . . 41
3.4.4 Probabilistic Likelihood Estimation based Model . . . 42
3.4.5 Social Tie-strength based Model . . . 42
3.4.6 Social Coefficient based Model . . . 43
3.4.7 Label Propagation based Model . . . 44
3.4.8 Social Concentration based Model . . . 45
3.5 Benchmarking Evaluation . . . 45
3.5.1 Datasets . . . 45
3.5.2 Ground-truth Information of Datasets . . . 47
3.5.3 Friendship, Distance and Check-in Characteristics . . . 47
3.5.4 Node Locality . . . 49
3.5.5 Parameter Settings . . . 50
3.5.6 Metrics for Evaluation . . . 51
3.5.7 Performance Evaluation Configuration . . . 52
3.5.8 Effectiveness on different types of social media datasets with different parameter settings . . . 53
3.5.9 Effectiveness on Local vs. Global Inference . . . 71
3.5.10 Effectiveness on Different Types of Users . . . 71
3.5.11 User Prediction Coverage . . . 72
3.5.12 Running Time and Memory Consumption . . . 75

3.5.13 Region-specific Comparison of Overall Prediction Performance of the Models . . . 77
3.6 Discussions and the Findings . . . 77
3.7 Summary . . . 79

4 Geolocating Activity Location in Location-based Social Network 81
4.1 Introduction . . . 82
4.2 Problem Definition . . . 83
4.3 Methodology . . . 84
4.3.1 Selecting Neighbors for Location Inference . . . 84
4.3.2 Creating User Sequence for Location Inference . . . 86
4.3.3 Algorithm . . . 86
4.4 Location-based Social Network Datasets and Characteristics . . . 87
4.5 Experiments and Results . . . 91
4.5.1 Effectiveness . . . 92
4.5.2 User Inference Coverage . . . 93
4.5.3 Running Time . . . 95
4.6 Summary . . . 96

5 Top-k Socio-spatial co-engaged Location Selection in Social Network 97
5.1 Introduction . . . 98
5.2 Problem Formulation . . . 101
5.2.1 Socio-spatial Relevance . . . 101
5.2.2 Socio-spatial diversity . . . 102
5.3 An Exact Approach . . . 104
5.3.1 Computing Bounds on Diversity of an Intermediate Set . . . 105
5.3.2 Algorithm . . . 108
5.4 An Approximate Approach . . . 112
5.5 A Fast Exact Algorithm . . . 115
5.5.1 Computing Bounds on Relevance . . . 115
5.5.2 Advanced Termination . . . 116
5.5.3 Algorithm . . . 117
5.5.4 Fast Approximate . . . 119
5.6 Experimental Evaluation . . . 121
5.6.1 Efficiency Evaluation . . . 123
5.6.2 Comparison with Existing Models . . . 127
5.6.3 A Case Study . . . 132
5.7 Summary . . . 132

6 Co-engaged Location Group Search in Location-based Social Network 135
6.1 Introduction . . . 136
6.2 Problem Formulation . . . 138
6.3 Filter-and-Verify Algorithm (FVA) . . . 142
6.3.1 m-$\widehat{core}$ components of candidate users, $\widehat{C}(m, CU)$ . . . 144
6.3.2 Computing score gains . . . 144
6.3.3 Maximum additional social connectivity δ_g for a new user . . . 147
6.3.4 Lower Bound on Check-in . . . 147
6.3.5 User pruning using check-in bound . . . 148
6.3.6 Location pruning using check-in bound . . . 148
6.3.7 Algorithm . . . 149
6.4 Greedy Forward Expansion Algorithm (GFA) . . . 150
6.4.1 Computing Score Gains . . . 151
6.4.2 Computing upper bound of Δ_g . . . 152
6.4.3 Location Pruning . . . 153
6.4.4 Algorithm . . . 154
6.5 Greedy Incremental Algorithm (GIA) . . . 155
6.6 Experiments . . . 156
6.6.1 Experimental Setup . . . 157
6.6.2 Experimental Results and Discussions . . . 158
6.7 Summary . . . 163

7 Conclusions and Future Works 165
7.1 Conclusion . . . 165
7.2 Future Works . . . 166

Bibliography 169

List of Figures

3.1 The proposed benchmarking framework 33
3.2 Categorization of location prediction approaches 35
3.3 Probabilities of following as function of Distance 46
3.4 Average Node Locality as a function of Node Degree 49
3.5 *AED@d* in Twitter and Gowalla using Different Data settings 53
3.6 *AED@d* in Brightkite and Foursquare using Different Data settings 54
3.7 *AED@k%* in Twitter and Gowalla using Different Data settings 56
3.8 *AED@k%* in Brightkite and Foursquare using Different Data settings 57
3.9 Precision of the Location Prediction Models using Twitter Dataset 59
3.10 Precision of the Location Prediction Models using Gowalla Dataset 60
3.11 Precision of the Location Prediction Models using BrightKite Dataset 61
3.12 Precision of the Location Prediction Models using Foursquare Dataset 62
3.13 Accuracy of the Location Prediction Models using Twitter Dataset 63
3.14 Accuracy of the Location Prediction Models using Gowalla Dataset 64
3.15 Accuracy of the Location Prediction Models using BrightKite Dataset 65
3.16 Accuracy of the Location Prediction Models using Foursquare Dataset 66
3.17 Local Prediction Accuracy of the Location Prediction Models using Twitter Dataset 69
3.18 Local Prediction Accuracy of the Location Prediction Models using Foursquare Dataset 69
3.19 Global Prediction Accuracy of the Location Prediction Models using Twitter Dataset 70
3.20 Global Prediction Accuracy of the Location Prediction Models using Foursquare Dataset 70
3.21 Performance of models with different node degrees 73
3.22 Predicted users proportion (with error distance less than 160 km) with different Node Locality 74
3.23 Prediction coverage of models in different datasets with default configuration . . 75
3.24 Prediction coverage of models when Local prediction is considered 76
3.25 Running Time of the different models 76

3.26 User location distribution in original datasets . . . 77
3.27 Heatmap of user prediction in different models using Twitter and Brightkite data setting I . . . 78

4.1 Probability of Friendship as a function of distance . . . 89
4.2 Cumulative Distribution on Node Locality . . . 90
4.3 Average Node Locality as function of degree . . . 91
4.4 Average Error Distance within 20KM, 50KM, 100KM, 160KM . . . 93
4.5 Accuracy comparison . . . 94
4.6 User Inference Coverage . . . 95
4.7 Running Time of the models in Gowalla and Brightkite datasets . . . 96

5.1 An example of the *SSLS* query . . . 99
5.2 Social Diversity and Social Relevance Scores of u's locations (refer Figure 5.1) . 110
5.3 Spatial Diversity and Spatial Relevance Scores of u's locations (refer Figure 5.1) 111
5.4 Socio-spatial Diversity and Socio-spatial Relevance Scores of u's locations (refer Figure 5.1) . . . 111
5.5 Node exploration steps of `Exact` and `Exact`$^+$ . . . 112
5.6 Proportion of having similar total score as optimal solution w.r.t. number of iterations using `Exact`$^+$ . . . 119
5.7 Cumulative probabilities of first three iteration . . . 120
5.8 Friendship Distribution . . . 122
5.9 Characteristics of user check-ins . . . 123
5.10 Varying k . . . 124
5.11 Varying check-in groups . . . 125
5.12 Varying Number of Friends . . . 126
5.13 Varying ω, α . . . 126
5.14 Varying k . . . 127
5.15 Varying Large k . . . 128
5.16 Varying check-in groups . . . 128
5.17 Precision . . . 129
5.18 Precision when varying ω, α . . . 130
5.19 MMD Comparison . . . 130
5.20 Social Coverage . . . 131
5.21 Social Entropy . . . 131
5.22 A case study . . . 133

6.1 Location-based Social Network and Location Graph . . . 137
6.2 Average running time in default configuration . . . 159

6.3 Average Co-Engagement score and participating user size in default configuration 159
6.4 Running time when k varies 159
6.5 Co-Engagement score when k varies 160
6.6 Participating user size when k varies 160
6.7 Running time when θ varies 160
6.8 Co-Engagement score when θ varies 161
6.9 Participating user size when θ varies 161
6.10 Scalability analysis by varying θ, $k = 40$ 162
6.11 Running time when m varies 162
6.12 Co-Engagement score when m varies 163
6.13 Co-Engagement score when α varies 163

List of Tables

2.1 Comparing existing works on group search queries 27

3.1 Features and Time Complexity of the Models. 34
3.2 Summary of the Datasets used 47
3.3 Summary of the Twitter Dataset 47
3.4 Summary of the Gowalla Dataset 48
3.5 Summary of the Brightkite Dataset 48
3.6 Summary of the Foursquare Dataset 48
3.7 Parameter Settings in different models 50
3.8 Metrics used in different models 50
3.9 Mutual Prediction Ratio in TW-IV data 68
3.10 Mutual Prediction Ratio in TW-IV data 68
3.11 Mutual Prediction Ratio in FS-I data 68
3.12 Mutual Prediction Ratio in FS-IV data 68

4.1 Statistics of the datasets. 89
4.2 Summary of the Labeled and Unlabeled User Information. 92

5.1 Basic Notations 101
5.2 Dataset Statistics 121
5.3 Parameters and their values 123

6.1 Basic Notations 142
6.2 Dataset Statistics 157
6.3 Parameters and their value ranges 158

To My Parents and Mejo Dada

Chapter 1

Introduction

The location information of social users are important in various applications such as emergency reporting system, online marketing, etc. Due to the advancement of mobile technologies and easy access of the internet, a large portion of worldwide populations are using social media platforms to share their daily updates and communicate with their linked online friends. The modern social networking sites such as Facebook, Twitter have enabled their locations features, where users can tag their daily activities through check-ins functionality. Therefore, it generates huge amount of social and spatial data every day, which leads to contribute an in-depth socio-spatial data research to improve the efficacy of location based services. However, users' mobility and their social interactions in the physical world have not been fully investigated in the existing works that analyze the geo-social properties of networks.

In this thesis, we have explored the research by investigating the social and spatial properties of a network including the socio-spatial engagement of the users in location recognition. One direction of this thesis is to focus on location prediction of social users using implicit network information. The other directions are to utilize the user engagement to different locations, where selecting a small subset of relevant and personalized representative point-of-interests (POIs) of a social user can be useful to various location based real-time applications. Meanwhile, it is also interesting in a location-based social network to analyze the relationships between a set of locations and cohesive groups of participating users to the location set. Considering the engagement scope of a set of locations to socially cohesive groups participating to the location set will enrich the research scopes in the area on geo-social analytics. From the above mentioned research directions, we have studied three fundamental research problem in this thesis: (1) Inferring the locations of unlabeled social users by exploiting various implicit information available in the networks; (2) Selecting socially and spatially relevant but diverse location set for each social user in the networks; (3) Search for the location group in geo-social networks that are highly co-engaged by the members in cohesive user groups.

Section 1.1 presents the motivation, background, research gaps, and the fundamental idea of the problems introduced above. In Section 1.2, the primary contributions of the studied

problems in this thesis are given. The organization of thesis is outlined in section 1.3.

1.1 Motivation

As mentioned in the previous section, the location information of social users has significant importance to various online applications. In a location-based social network, the check-in locations signify user preferences for various socio-spatial activities. In this thesis, we have studied the socio-spatial properties of the locations and proposed three research problems that integrates the spatial information and user relationships to improve the quality of services in various real-time location-based applications. Below, we discuss the motivation of the research problems in detail.

1.1.1 Location Prediction in Social Networks

User location information contributes to in-depth social network data analytics. Discovering physical locations of users from social media helps us to bridge the online and offline worlds. This also supports many real-life applications like emergency reporting [6, 131], disaster management [94, 150], location-based recommendation [84, 112, 165, 166], location-based advertisement [158], region-specific topic summarization [125], disease outbreak monitoring [116]. Therefore, it is important to extract the spatial information from large social networks to support the real-life applications. Below, we highlight the challenges and objectives of the research problems on location prediction in social networks.

- **Exploiting the Implicit Network Features**

Location information is not always available in a social networks because most users do not want to disclose locations in their profiles for reasons such as users' privacy, users' attitude, or even lack of interest to disclose [66]. For instance, only 16% of the users in Twitter register location information in their profiles [90]. In another study, Cheng et al. [25] report that 21% of Twitter users from USA provide their location as city name, and 5% provide their geo-coordinates in tweet texts. This calls for the development of location prediction methods that can effectively exploit various implicit information inside the network to estimate users' locations.

- **Generalized Procedure-oriented Location Prediction Framework**

Location information of users in a social network are available as implicit information in the network. There exist significant interest in predicting user locations through public posts, metadata [13, 24, 66, 100, 128, 160], user-generated contents (UGC) [24, 66, 160], network features [9, 74, 101, 127]. On the other hand, some location prediction models [89, 90, 121, 124] consider hybrid strategy to exploit both the user-generated contents as well as the network

information. Such hybrid models have the flexibility to use either one or both information types. With so many different models available for location prediction, it becomes important to compare their performance on standard benchmark datasets using similar metrics. The majority of the existing models are based on different internal configurations that best suit their targeted applications, and hence it is difficult to analyze, compare, and evaluate their suitability in a common base. It is also not clear how these models will perform in different scenarios such as different social network, different types of users, and location sparsity. Since the list of location prediction models is extensive, it is important to choose the representative approaches from each prediction category and develop a generalized procedure-oriented benchmark to compare their relative performances. Therefore, in this thesis we investigate the location prediction problem that infer the stable location (i.e., home location) of social users considering the network properties and structure. Our aim is to gain insights into the existing prediction models.

- **Improving Existing Location Prediction Model**

On the other hand, the activity locations of social users are important to various real-time applications like emergency reporting system, news recommendation. An LBSN user has higher chance to be available at or near her top activity location [136]. However, obtaining the activity locations of the users in a large network is not an easy task as majority of the users do not allow an LBSN application to disclose their check-in information in the public. Hence, it is important to predict such locations that have high chance to be located at. Inferring such locations for LBSN users can effectively reduce the spatial search space without affecting the socio-spatial relationships in a network. For example, in case of emergency, an emergency reporting system may smartly target to the selective activity locations of an affected region where users have higher chance to be available. In this thesis, we propose a network-based location prediction model that can use the friendship information to propagate the location information efficiently in the network. We notice that the existing label propagation based location prediction model, e.g., SLP [74], suffers from several issues: (1) an incorrect location estimation of a friend may lead to increase error distance. (2) if some noisy social connections exists in the network, the estimated locations of users may shift far from the original locations. (3) consumes much time to converge the location points of each user as they consider 'all' the neighbors in the process irrespective of their importance to the user. Therefore, these existing issues motivate us to develop an iterative algorithm, Sequential Spatial Location Propagation (SSLP), that can improve the efficacy of the existing label propagation model.

1.1.2 Co-engaged Location Selection for Social Users

Almost every modern social networking site has enabled their locations features, which are now commonly referred as Location-based Social Networks (LBSN). These LBSN include Foursquare, Yelp, Flickr, Twitter, Instagram, Facebook, and many more. The huge popularity of these

LBSN is playing a vital role on the explosive growth of location-based services (LBS) market, which is projected to reach USD157.34 billion by 2026 [52]. Therefore, it is important to exploit the Socio-spatial characteristics of relationship in social networks to capture the mobility and user preferences in sophisticated manner.

With the location enabled features, users tagged their daily activities in different locations through check-ins. Since these check-ins capture user preferences, they are being heavily used in many applications such as recommendation systems [79, 163], location based advertising [105], group formation [7], and so on. However, not every check-in location of a user is equally important as various socio-spatial factors may lead to give a particular location more importance than the others. For example, one may give preference to a location where most of the user's friends has visited than the other locations with few visits by friends. As the number of candidate locations of each user through different form of check-ins (e.g. location tags, review places, etc.) is growing exponentially day by day, one of the fundamental problem in LBS applications is to select the best set of locations from a large candidate set. More specifically, the social factors are significant in distinguishing the preferences of the locations to a friend. Similarly, spatial factors can influence the user and her friends' interest in different spatial proximity. Therefore, it is important to exploit both the social and spatial characteristics of relationships among the social network users and their locations to better support the location-dependent applications by capturing the mobility and user preferences.

In this thesis, we introduce a new concept of Top-k *relevant and diversified* location selection for a user in an LBSN that considers both spatial and social aspects of the locations in the network space. We name this problem, as Socio-Spatial Location Selection (*SSLS*). To better motivate the importance of the socio-spatially relevant and diversified location selection problem, consider some practical application scenarios mentioned below.

Event Organization. Let us consider a scenario where a social network user wants to organize a series of social events in multiple locations, which will be preferable and convenient for both the user and her friends. More specifically, the user wants to select these locations such that they are (i) related: locations are the user's favorite ones where she visited a number of times earlier; (ii) socially and spatially relevant: locations where many of the user's friends also visited these places or some nearby places; (iii) spatial diversified: the selected locations are spatially distant, e.g., in different cities; and (iv) social diversified: each selected location should cover a set of friends such that the selected locations together can cover a maximum number of friends and any two selected locations have the minimum overlap of friends to be covered.

Outlet Opening. Nowadays, online business shops often maintain a Facebook page with many followers who like their products. Suppose an online business wants to open new outlets at k number of locations that can attract most of its customers (followers) and their friends (potentially new customers). The business shop can pre-define multiple regions (e.g., suburbs, cities) suitable for their future business. One can consider the candidate locations of the business

shop as the check-ins of the customers within the predefined regions. Thus, to select the k locations from a large number of candidate locations (check-ins of the customers), the shop owner would like to consider the following: the selected locations are relevant to the current followers (spatial relevance) and their friends (social relevance). Also, the selected locations should be distant so that they can cover different areas (spatial diversity) and attract different groups of potential customers through these outlets (social diversity). In this example, the check-ins of all users who liked the business page or their products (in a city) are considered as the candidate locations from where we need to select the top k locations for opening outlets. Therefore, this example shows that without the loss of generality, our approach can be applied for location selections for a linked group of users.

Furthermore, the *SSLS* problem discussed in this thesis, can advance the other applications with the considerable extension in a networked metric space, where both the relevance and diversity are important. Consider the following scenario.

Paper Topic Selection. Solutions to the *SSLS* problem can advance other applications with considerable extensions. Consider a high-dimensional networked space of a co-authorship network. Each author is attributed with a set of keywords denoting her expertise, while an edge represents the co-authorship relation. If a author wants to know k top trending (relevant) and diverse keywords for her collaborations, then it requires to retrieve the k topics that she and her co-authors will be mostly and jointly interested in. In essence, the set of keywords are analogous to locations in location recommendations, and authors having expertise in a particular keyword can be considered as 'check-ins' to these keywords.

1.1.3 Location Group Search in Social Media

Searching for communities based on query nodes in a social network has been extensively investigated. The majority of the existing works emphasize on finding user groups that are socially cohesive and their locations are close spatially. In general, a socio-spatial group query returns a cohesive user group spread over a small spatial region. The existing works mainly focus on user group search where each user has single location, and few studies consider location-based social network with multiple check-ins. However, for a given query location, it is equally important to identify the best group of locations (containing the query node) which are highly co-engaged w.r.t. cohesive user groups. Such set of locations can be useful to various applications ranging from marketing to location recommendation.

As mentioned in the previous section, the study of socio-spatial properties in location-based social network can help us to better understand mobility and user preferences in a network. Among the large number of check-ins, the set of socio-spatially significant locations of each user represents the user-level preferences only in a spatial space. However, the user-level preferences may not fully represent other friends who together form cohesive communities. Therefore,

given a query location, identifying a highly co-engaged set of locations containing the query location can enhance community engagement near the query point. For example, an event manager can easily select a set of locations to organize events which are visited together by large number of users from socially cohesive groups. Such set of locations have much impact in real-time applications as a large number of users can easily participate together in various events associated with the location group. Therefore, identifying such locations may provide more sophisticated information to the location based services that can greatly enhance user experience and togetherness in a spatial region.

In this thesis, we study the problem of Co-engaged Location group Search (CLS) that identifies the top locations which are visited by cohesive user groups. Among the various usability of the co-engaged groups of locations, we detailed two relevant applications below.

Tour Planning. Let us assume, a tour planner wants to plan a city-tour for a group of visitors. The likely locations for the tour should be selected in a way that are previously visited by the members from other socially cohesive user groups, and should contain a particular place of tourist attraction suggested by the tour planner. By identifying the co-engaged groups of locations with high quality in terms of user engagement at the selected locations can help to emerge a successful tour to the visitors. The selected group of locations should have potentiality to become active and attractive to new user groups. Therefore, visiting a set of locations which are already known among the other socially connected user groups are more preferable to participate various activities together. The Co-engaged Location Search (CLS) can be applied to identify the best groups of co-engaged locations to suggest the tour planner to arrange the tour for the visitors.

Event Organization. The event organization applications such as Eventbrite, Meetup support to organize events in various physical locations. Let's assume, an organization wants to organize various events in a group of locations such that, the selected locations should be already known (e.g., visited previously) to the participating future users to the events. Also, the users should be socially connected satisfying some social cohesiveness constraints. The CLS can be used to search for the best co-engaged group of locations to organize the events such that the targeted users can easily participate the events together. Thus the participating users to the location groups will determine the scores to identify the best location group for organizing the events. Meanwhile, the social cohesiveness and size of the participating users can be used as an estimator to the number of participator for the events.

1.2 Contribution

This thesis contributes towards the development of technical methods for capturing and analyzing the socio-spatial properties of social networks. We focus on studying the relationship between user mobility and social connections in a network to better understand the user engagement in

the geo-social space. In this thesis, we have explored several tasks on selecting personalized locations that have great importance in perspective of various real-time applications. The detailed contributions of the thesis are as follows.

1.2.1 Location Prediction in Social Networks

The existing location prediction models in the literature are based on various internal configurations and it is difficult to analyze how better a particular model can perform in different scenarios. Moreover, the current evaluation practices of the existing models reveal significant disparity between different evaluation settings. There exists several challenges in order to analyze the relative performances of the existing models, such as,

- A unified benchmark framework is difficult to abstract due to the diversity in the existing models.
- It is critical to diagnose the functionality of the existing models from a common viewpoint.
- To do a unbiased comparisons of the performances, it is essential to re-implement the representative models in a common coding platform and such task is quite challenging.
- Defining a suite of metrics to evaluate the multiple aspect of the existing models is a challenging task.

Therefore, first, we have conducted a comprehensive benchmarking study that performs in-depth analyses and comparisons of the different location prediction models. We address the above mentioned challenges in our proposed framework. Next, we propose a location inference approach that considers the social relationships in a network and can incorporate the location granularity by effectively propagating the location information using small amount of ground truth location. We also define an inference sequence to converge the location propagation process efficiently. Specifically, we make the following major contributions for the task of location prediction in social network:

- We review existing location prediction techniques and re-implemented eight representative models in a common code-base.
- We perform an in-depth evaluation of the models using four real-world large-scale social media datasets with five different data settings on user location sparsity.
- We evaluate eight representative prediction models using five evaluation metrics under different user-centric parameters and data settings to demonstrate their strengths and limitations in a transparent comparison framework.

- We have adapted the network-based location prediction techniques for predicting user location in check-in datasets.
- We provide significant insights into the location prediction problem within large-scale social media and draw some interesting take-away conclusions about the compared models.
- We demonstrate that social relationship in a network is a useful source of information to infer the activity locations of the users.
- We demonstrate that a proper location inference sequence can help to effectively infer user locations in sparse dataset.

1.2.2 Co-engaged Location Selection for Social Users

The existing works on spatial object selection (or equivalently location selection) from a large dataset of spatial objects mainly focus on selecting a subset of diversified objects w.r.t. various spatial distance constraints among objects. Two closely related works in this research domain are DisC [37] and spatial object selection (SOS) [61]. DisC essentially selects the subset of diversified objects, where any two selected objects must be at least r distance from each other and there should be at least one object (un-selected) in the dataset in less than the r distance from every object. On the other hand, SOS selects k diversified objects in such a way that any two selected objects must be at threshold distance from each other and the aggregate similarity (computed based on semantic attributes) from the selected set of objects to the whole dataset is maximized, which ensures the representativeness of the selected objects.

However, there exists some noticeable gaps that make them inapplicable, including (i) Both DisC [37] and SOS [61] define diversity based on spatial distance only. Thus, they do not account for the important aspect of diversity in geo-social networks, which we refer to as *social diversity*. We argue that both the spatial and social aspects need to be considered for selecting diversified spatial objects in geo-social network domain in order to get the best *SSLS* set. (ii) Both the approaches depend on a user-defined distance threshold to get a better diversified object set of size k. But it is hard for an end user to define such threshold values without knowing the underlying data distribution. (iii) As the selection models in both works are based on parameters, e.g. predefined threshold, the selection process cannot be personalized towards individual users with their particular preferences.

To address the above limitations, we introduce a new concept of Top-k *relevant and diversified* object selection for a user in an LBSN that considers both spatial and social aspects of objects in the network space. The primary contributions of this work on selecting socio-spatially top-k locations for social users are as below:

• **SSLS Formulation.** We formally define the problem top-k Socio-Spatial co-engaged Location Selection. We provide detailed algorithms and metrics for using social and spatial

relevance, and diversity in order to maximize the spatial and social coverage of the search space.

• **Exact and Approximate approaches.** We first propose an `Exact` approach by developing some pruning strategies based on the *derived lower bounds on the diversity* of an already explored feasible set. Such an approach avoids the exploration of a large number of locations that are irrelevant to users and their social connections. We also devise an efficient exact method (`Exact`$^+$), a variation that derives bounds based on the *relevance* of candidate locations, and hence avoids repetitive complex diversity computation of groups of locations as in the `Exact` approach. In addition, we present an approximate approach, in which we derive relaxed bounds and propose an advanced termination criteria based on the score of the best feasible set, and the diversity of remaining locations. We also introduce a greedy-based Fast Approximate approach that uses the bounds of `Exact`$^+$ and greedily selects the best locations.

• **Extensive Experimental Evaluation.** Finally, we have conducted extensive experiments to evaluate the effectiveness and efficiency of our proposed approaches using four real-world datasets. We have compared the proposed algorithms with two adaptive greedy-based approaches namely, *GMC* [149], and *GNE* [149] that consider relevance and diversity. We also have compared our approaches with an adapted version of Spatial Object Selection (SOS) [61]. Our experimental results show that `Exact`$^+$ outperforms `Exact` and the Approximate approach by 3 to 6, and 2 to 3 times in default data settings, respectively. Moreover, we show that our approaches result in better social and spatial coverage, and diversity of the selected location set for a user when compared with the adapted algorithms.

1.2.3 Location Group Search in Location-based Social Network

As mentioned earlier, the socio-spatial user group queries in LBSN network have great importance to understand the behavior of cohesive user groups in the spatial space. Given a set of query nodes and other constraints, the socio-spatial group queries [54, 138, 161, 174] aim to find the best user group near to a single or multiple query location where the users have strong social connections with the group members and have spatial closeness to the query locations. However, it is also equally interesting to suggest the locations to the users which are also visited by different socially cohesive communities. Therefore, visiting a group of locations which are already known among the other socially connected user groups are more preferable to participate various activities together.

In this thesis, we demonstrate the spatial level co-engagement of the social users to search for the best location group that can maximize the overall involvement of social users in the selected locations. The participating user groups to the selected locations should satisfy structural constraint, and the check-in density of the user groups to the selected location set should be higher. Meanwhile, the locations in the result set should be distance reachable (close spatially and connected) containing the query location, which confirms the spatial connectivity between

the selected location set. Therefore, identifying such locations with high quality in terms of (i) social connectivity of the participating users groups, (ii) engagements of the participating user group members to the selected locations, (iii) spatially connected within a distance, have good potential to become more useful in location-based real life applications.

The primary contributions on the task of co-engaged location group search in location-based social networks are summarized below:

• **Define co-engaged location group search problem in LBSNs.** This work introduces the location group search problem from the perspective of user community engagement. To the best of our knowledge, this is the first work to find a location group that are highly co-engaged w.r.t. participating social user groups. We also have suggested relevant applications of this problem.

• **Solutions.** We propose three solutions to solve the *CLS* problem effectively and efficiently. The first solution is based on filter-and-verify algorithm where the key idea is to remove some locations and their corresponding check-in users based on the social constraint. The second solution is a greedy approach that iteratively selects locations that can increase the co-engagement score of existing location set. At the same time, the greedy approach prunes some locations using social connectivity properties of set of users associated with the locations. The third solution heuristically selects locations based on some predefined rule sets.

• **Comprehensive Evaluation.** We conduct comprehensive experiments using real-world LBSN datasets in terms of efficiency and scalability of our proposed algorithms.

1.3 Organization

The remainder of this thesis is organized as follows:

- Chapter 2 introduces the related work on location prediction, spatial object selection, socio-spatial group queries in social network.
- Chapter 3 presents the location prediction problem in social network. A detailed study of the existing algorithms and their relative performances on similar metrics and dataset settings are studied.
- Chapter 4 presents a network based location prediction approach, the corresponding algorithm and experimental results.
- Chapter 5 presents the top-k socio-spatial co-engaged location selection problem for social users, corresponding algorithms and the experimental results.
- Chapter 6 presents the co-engaged location group search problem, corresponding algorithms and the experimental results.

- Chapter 7 concludes our research and provides the possible extension of this thesis and other unexplored areas as future research direction.

Chapter 2

Literature Review

In this chapter, we conduct a comprehensive literature review on the works related to the problems studied in this thesis. In Section 2.1, we first review the existing research on location prediction tasks, including the classical methods, and the advanced approaches based on neural networks. Next, we discuss some existing works on comparitive studies on location prediction models in social networks. Further, in Section 2.2, we present the related works on location selection problem that mainly covers the existing researches on spatial object selection, spatial clustering and sampling, socio-spatial queries in social network, etc. Finally, in Section 2.3, we introduce the existing studies on group queries in social network, which mainly focus on attribute driven community search and geo-social group queries in LBSN graphs.

2.1 Location Prediction in Social Network

Based on the available input information in the social network (e.g., contents, relationship information), the existing classical models on location prediction in social media can be divided into three broad categories e.g., content based, network based, and hybrid models. We summarize the existing works of the above mentioned three categories in Section 2.1.1, Section 2.1.2, and Section 2.1.3, respectively. We further discuss the recent works on neural-network based location prediction models in Section 2.1.4. Further, in Section 2.1.5, we summarize the existing comparative studies on location prediction in social network.

2.1.1 Content based Location Prediction Approaches

Content-based approach is the most well-known location prediction approach among the literature. The home location of a social user can be reveled by certain keywords in user posts (e.g., tweets). For example, the users from Philadelphia often call themselves "phillies" and the phrase "howdy" is usually used by the residents from Texas region. The content-based approaches analyze the user-generated contents to predict location of a social user. The prediction

performance of the models in this category depends on the availability of location indicative words in their post texts. Therefore, the underlying challenge for the content-based location prediction approaches is to specifically map the social users to some locations via the location indicative words (LIW) available in the network. A large amount of studies on identifying local words are available, either unsupervised or supervised.

For example, Cheng et al. [24] consider the problem of local word identification as a supervised classification problem. The authors identify a set of location indicative words (e.g., "Sydney") as features to label a user to a location. They also extracted some locally used terms (e.g., "rocket") and mapped them with the locations ("Houston") where these keywords are frequently posted. However, the model [24] suffers from data sparsity problem, as all the local keywords can not be assigned to locations as labeled user information is sparse in the network. To alleviate the issue, Gaussian mixture models (GMM) are used to achieve smoothed word usage distributions in [18, 118]. Similar to [24], Ryoo et al. [128] extract the correlation between location indicative words and GPS locations and build geographic distributions of the local words. Using the distribution, the user location is selected as the location with maximum likelihood probability [9]. The work in [128] achieves an acceptable performance by filtering the top 1000 local words from a Korean tweets dataset to infer the location from user tweets. The majority of the models that consider the local words to predict user location, use probabilistic approach to characterize the conditional distribution of users' locations using the post contents. Given a set of words $W(u)$ from user u's post contents, the probability $P(l|u)$ of a location l to be associated with a user u is calculated as, $P(l|u) \propto \sum_{w \in W(u)} P(l|w)P(w)$. Here, $P(w)$ is the probability of word w over the whole corpus and $P(l|w)$ is the location distribution of word w.

Unlike [24], Chang et al. [18] identify the location indicative words using unsupervised learning. Instead of estimating a language model for a particular location (e.g., city), Chang et al. [18] suggest to estimate the location distribution on spatial word usage probability with Gaussian Mixture Models. The approach shows that using a very few local words, it can achieve a comparable performance than the supervised approach proposed by Cheng et al. [24]. In another work, Hulden et al. [71] propose a classification based approach to classify the tweet text into discretized cell grids with the keywords as features. The authors address the data sparsity problem in the network through smoothing out the relevant only features by kernel density estimation. They show that the proposed model generates significant improved results compared with fully discritized models. Lee et al. [82] propose another location prediction approach that build a language model using the locations and geotagged tweets. To increase the accuracy of the location prediction, the work [82] utilizes local words and applies three smoothing techniques, e.g, Jelinek-Mercer smoothing, Laplace smoothing, and Absolute discounting. The model proposed by Bo et al. [13] is also based on language model which was built using location indicative words. The authors use Inverse Location Frequency (ILF) and Inverse City Frequency (ICF), to measure the locality of the keywords available in the post contents (e.g., tweets).

Yamaguchi et al. [160] calculate KL-divergence scores of the words to identify the local words from the post contents. Mahmud et al. [100] propose a hierarchical ensemble algorithm for predicting the home location of social users. The proposed model is based on classification where content-based, behavior-based features were used to train the classifier.

2.1.2 Network based Location Prediction Approaches

The second category is the network based location prediction approach. In this section, we discuss various location prediction models, known as network-based location inference models, that use the structural information of a network. A social user not only share their thoughts or moments in the network using posts (e.g. tweets), they also make and interact friends of their interests. In general, the majority of the friendship relationships are established with other social users who are living or interacting physically nearby [9]. Therefore, the relationship features are important to reveal the locations of social users. The network-based location prediction models measure the relationship between geography and friendship, and leverages the assumption that nearby users are more likely become friends. One of the well known network based approaches was introduced by Backstrom et al. [9] to infer locations of Facebook users. The basic assumption of the model is that socially connected users are likely to be located nearby. The model is based on likelihood of observing a relationship among a user pair, given a distance between them. There exists other similar models, e.g., Davis. et al [33], Ren et al. [126], based on the likelihood estimation approach that also study the interplay between relationships and geographical distance. Davis. et al [33] infer a user location considering the frequently seen locations among their social connections. Additionally, Ren et al. [126] study an explicit smoothing technique called circular-based neighborhood smoothing that considers all geographic neighbors within 40 miles to the center of a location.

The above mentioned models implicitly assume that friendship observed on social network implies the physical locations are also nearby. However, not all the friends are equally important to a user to infer their locations from. Hence, it is important to analyze the social closeness between friends who can provide strong indication to be co-located in a location. The tie-strength based models [19, 101] identify the stronger social edges to improve the location prediction accuracy. For example, McGee et al. propose FriendlyLocation [101] model that extends the approach in [9]. This model [101] considers several additional social factors such as number of friends, number of mentions, etc. to measure the user proximity w.r.t. the neighbors. The FriendlyLocation model is a semi-supervised model. Considering the geographical proximity as ground truth, the FriendlyLocation [101] model is trained using a decision tree considering several social factors, e.g., following relationships, number of friends, etc., as features. Further, the model assigns users with different social closeness to ten quantiles. The relationship information of location labeled users are used to train the decision tree of the FriendlyLocation [101] model.

For the inference task, the edges in each quantile are processed to predict the likelihood having an edge of a user with friends at a location. However, in the absence of quantile partitions, the FriendlyLocation model reduces to [9]. The basic assumption of these tie-strength based models [19, 101] is that strong social connections are more likely to share the same location of the friends.

Another network-based location prediction model, SLP [74], is based on the intuition that individuals establish social relationships with those friends they meet nearby. This model leverages the geographic distribution of an individual's ego network to infer location. SLP is a semi-supervised iterative label propagation algorithm that can effectively infer locations by propagating small ground-truth information in the network. This model considers bi-directional mention relationship in their experiments and demonstrated that the proposed approach has quite acceptable performance in terms of accuracy. In [29], Compton et al. extend the SLP [74] model to consider into the edge weights in the social network and to limit the propagation of noisy location information. Unlike SLP that calculates the median using all the friends' locations to annotate a unlabeled user, Compton et al. [29] select the locations of those friends with whom a stronger evidence of a close relationship exist. The model also optimizes the objective function by parallel coordinate descent. Kong et al. [78] propose SPOT model that calculate cosine similarity to measure the social closeness among the neighbors. The maximum likelihood of a neighbor location w.r.t. social closeness score is assigned to the user location. The SPOT model considers confidence-based iteration, where the estimated location of a user is allowed to pass if it has higher confidence to be predicted with lower error distance. Different from the above mentioned social-closeness based models [29, 78], Yamaguchi et al. [159] propose a location inference model that assumes a higher proportion of friends live in a small dense region. The proposed model is called landmark mixture model (LMM) that identifies a set of landmark users who resides in close proximity. The authors argue that landmark friends are reliable source to label a user, and the maximum likelihood location of landmark friends is used to annotate unlabeled users.

2.1.3 Hybrid Models for Location Prediction

The hybrid models integrate both the user-generated contents and the user relationships available in the social network. Li et al. [90] proposed an Unified Discriminative Influence (UDI) model to infer user location in large social network. This approach models users' influence as a bivariate Gaussian distribution and the variance of the distribution is considered as influence scope. User nodes with larger influence scope (e.g., celebrity users) are more likely to be followed by spatially diverse range of users. Therefore, a social user having a higher influence score may not be useful to infer a better location. In UDI [90], two types of influence models are generated using the *following* and *messaging* relationships. A location maximizing the joint probability of

generating such relationship edges is considered as the predicted location. Li et al. [89] propose another hybrid model, Multiple Location Profiling (MLP), based on the probabilistic generative model. This model uses supervised extension of Latent Dirichlet Allocation (LDA) to model the relationship between users and locations. This model combines the power law and multinomial distributions in a non-trivial manner. The explicit correlation between location and the *following* relationships are measured using the following probability. The MLP [89] model captures the location based relationships using random generative models. The random generation measures the following and tweeting probabilities that a user can randomly participate to follow some users or tweet an odd venue. MLP model can discover a user's multiple locations. Rahimi et al. [121] proposed a hybrid geolocation model using label propagation with Modified Adsorption where the text-based geolocation approach is applied to improve the prediction coverage. This model filters the celebrity node in the process to update the social graph. In [124], Rahimi et al. propose another hybrid approach where the text information are used to estimate the users of the disconnected component of a social graph.

2.1.4 Neural Network based Location Prediction Approaches

Besides the traditional methods on location inference in social network, the neural network based location prediction approaches are getting much attention. Lourentzou et al. [98] propose a text-based neural model for geolocation prediction. The work suggests that appropriate configuration of the neural network features can indeed improve the performance of location prediction model comparing to traditional content-based approaches. Miura et al. [103] proposed a complex neural network based geolocation prediction model that integrates the text, metadata, and network information in an unified manner. In [122], Rahimi et al. propose a neural network based user geolocation model on the Multilayer perceptron (MLP) with one hidden layer. The parameters of the hidden layer is used as word and phrase embeddings. The authors consider l_2 normalized bag-of-words representation as the input features for a given user. The output of the model is a predefined discretization of real-valued coordinates of training locations, generated by either a k-d tree or k-means. The experiments using twitter datasets, the model shows a better performance than the state-of-the-art text-based methods on geolocation. In another work, Rahimi et al. [123] propose a hybrid geolocation model on Graph Convolutional Network (GCN) where the neighbor information is propagated through GCN layers. To control how much neighbor information should be passed w.r.t. a node, Rahimi et al. [123] use layer-wise highway gates for smoothing neighbor nodes in GCN. In terms of performance, the authors show a better accuracy with a minimal supervision scenario. Tian et al. [147] propose a twitter geolocation method based on representation learning and label propagation (ReLP). The proposed method combines heterogeneous relationships in network, effectively filters unrelated relationships, and learns to represent characteristics of user-geographic attributes. A multiview

neural network architecture is proposed by Do et al. [35] for Twitter user geolocation. The architecture combines multiview data representation into a unified model to infer the locations of users. The model leverages the features from different sources, such as, from textual information (TF-IDF, doc2vec [81]), user interaction network (node2vec [58]) and metadata (timestamp). The authors show that the performance of the multiview neural network based architecture is heavily dependent on user graph features. With the similar concept to [35], two Multiview learning models are proposed in [155] for the location prediction task. These models are based on the Graph Attention and Graph Convolution Network that exploit both the text and social connection information. In that work, the representations of textual information is build using TF-IDF, LDA [12], and doc2vec [81]; the network features were introduced by mention network. The main difference between the multiview learning model proposed in [35] and [81] is that Do et al. [35] integrate the features through concatenation and dense layers, where Wang et al. [155] uses Graph Attention Network and Graph Convolution Network. Another unified user geolocation method is proposed by Ebrahimi et al. [41] that also incorporates different types of information such as user network, tweet text, and metadata. Using the different data sources, the model first generates vector representation of the available information, and then concatenate the representations as the feature for classification. The model identifies the location indicative words using bidirectional Long Short-Term Memory (LSTM) networks reinforced with a context-aware attention mechanism. Similar to the work [41], Huang et al. [68] propose a hierarchical location prediction neural network (HLPNN), that combines seven features from text, network, metadata information for user location prediction. In [170], Hybrid-attentive User Geolocation (HUG), a hybrid attention mechanism is introduced that can automatically determine the importance of texts and social networks for each user. The social media posts and interactions in HUG are modeled by a graph attention network and a language attention network. A multi-task CNN model is proposed in [47] that combine classification and regression in an attention-based convolutional neural network. The model integrates the convolutional channels and pass through an attention mechanism to emphasize the meaningful pattern recognized by the convolutions. Another Multiview Attention-based Convolutional Model is proposed by Fornaciari et al. [46] that learns continuous node representation using network structure and user mention.

2.1.5 Comparative Studies on Location Prediction Models

There exist some comparative studies in the existing literature that compare the available models for location prediction in social media. Ajao et al. [4] studied the basic concepts in location inference techniques on Twitter social network and reported the accuracy of ten existing models. The comparisons are limited to the results presented in those ten works. From the survey in [4], it is not possible to derive a fair comparison of the models because the evaluations in the

original papers were not performed on the same datasets and standard configurations. Another survey on location prediction on Twitter is reported by Zheng et al. [171]. This survey focused on comparing the models on three types of location (i.e. home location, tweet location, and mention location) prediction tasks. However, their comparisons are based on the summaries of the prediction models and lack comparative analysis. The survey [171] also does not provide the technical backgrounds of the prediction models.

Jurgens et al. [75] conducted a comparative review and analysis of nine network-based geolocation inference techniques using a bi-directional Twitter mention network dataset. They investigated the performance of the models on the task of predicting "tweet location" of an arbitrary user's post. However, they did not investigate the models' effectiveness under different parameter settings. Moreover, the study [75] did not provide any insights into the models' designs. The effectiveness and efficiency of the existing location prediction models may vary due to model-centric parameter settings as well as dataset properties. A comprehensive comparison of the models requires testing the models under different data-centric parameters and on different types of social networks. Hence, the comparisons reported in [75] are insufficient as they use only one dataset under limited model-centric settings. Moreover, the prediction performance of network-based models is significantly affected by variations in location sparsity. However, the analysis of Jurgens et al. [75] does not consider variations in location sparsity at all. More precisely, they consider the data setting with a majority of the users (i.e. 80%) with location annotated to predict the locations of the remaining 20% users only. This setting is far from real-world scenarios.

The prediction of 'post' location (e.g. Tweet location) and 'user' location are two different tasks [18, 24, 126], and hence require different approaches and evaluation metrics. Another limitation of the analysis in [75] is the choice of evaluation metrics. The metrics used in [75] are AUC (Area Under Curve), Median-Max, and User Coverage (instead of Post Coverage) which can not be designed for similar types of prediction tasks. For example, AUC is used to evaluate the predicted locations of the posts, whereas the Median-Max and User Coverage measure the user-level performances. In this case, the highest error of a user's predicted posts' locations are identified and then the median of these errors across all users is reported as the Median-Max of the location prediction. There is a high chance of getting a misleading conclusion when the majority of the user's posts have lower error distance and few posts are predicted very far from the original post locations. In this case, the Median-Max distance errors of each user may give a higher value, but the performance of the corresponding models may yet generate a better accuracy. In such a case, while comparing different models, the Median-Max metric fails to produce coherent results leading to misleading conclusions.

Also, it is difficult to decide from [75], which models perform better on accuracy and prediction coverage. For example, if a model predicts locations of a few posts of each user, the user coverage of the model will be high, but it will fail to justify the post coverage and

accuracy of the model. Hence, the metrics used by Jurgens et al. [75] are insufficient to produce conclusive comparisons. In addition, the analysis in [75] lacks the functionality-wise comparison of the models in a common frame.

2.2 Object Selection in Location-based Social Network

Selecting socio-spatial relevant and diversified locations set for social users is another focus of this thesis. Therefore, in this section, we summarize the existing works on object selection in social networks. Specifically, we first discuss the related works on spatial sampling, LBSN queries in general, then present the existing works on different forms of diversified object selection in spatial and metric space. Finally, we discuss the relevant works on spatial object selection in social networks.

2.2.1 Spatial Clustering and Sampling

Spatial clustering techniques group together spatial set of objects which are closer to each other. The existing algorithms for spatial clustering can be divided into three broad categories: density-based, hierarchical, and partitioning. The density-based methods group the objects of dense region together and the outliers are labeled with the objects from sparse regions. For example, Ester et al. [43] proposed *DBSCAN*, a density-based clustering algorithm that identifies core location points p and spatial objects within a region centered at p with radius eps are included in the same cluster. The number of spatial objects should be at least $minPts$. A dense eps-neighborhoods are put into the same cluster of the core point p. The GDBSCAN [132] is a generalization of DBSCAN algorithm that can cluster point objects as well as spatially extended objects according to their spatial and non-spatial attributes. Further, Shi et al. [139] extend the density-based spatial clustering DBSCAN and propose a model Density-based Clustering Places in Geo-Social Networks (DCPGS) by considering both the spatial and social relationships between users who visit similar places. Unlike DBSCAN, in DCPGS, both the spatial distance and the social coherence of the places are considered into the cluster. Social entropy and community score measures were used to evaluate the quality of the discovered clusters in DCPGS. Another spatial clustering method, hierarchical spatial clustering, assigns objects to the clusters in either bottom-up or top-down manner. Various partitioning methods including k-means, k-medoids usually return the spatial clusters in spherical-shaped. Further, a randomized search based clustering method called CLARANS was proposed by Ng et al. [111] to identify the spatial clusters. Unlike the existing spatial clustering techniques that assume the distance function is Euclidean, a local search technique is carried-out in CLARANS. The k-medoid partitioning method is chosen as the basis of the CLARANS clustering algorithm.

Unlike the clustering methods, Sampling is based on the theory of probability where different

parameters can control the quality of selected samples. The spatial sampling concerns the selection of a location subset to estimate the characteristics of a given spatial region. Among the various spatial sampling approaches, the spatial auto-correlation [62] follows Tobler's first law in geography to cluster spatial objects. The other categories of spatial sampling, e.g., stratified random sampling [109], systematic sampling [40], cluster sampling [110] assume that population is independent and identically distributed. Another type of sampling is spatial heterogeneity based sampling that evaluates spatial densities. Such sampling techniques consider the objects as independent and evenly distributed in the space. However, these sampling techniques do not consider the relationships between objects.

2.2.2 General Queries in LBSN

Various geo-social queries have been studied [8, 142, 143] that focus on retrieving useful information combining both the social relationships and the locations of the users. For example, the top-k place query [143] fetches k places of a user based on the distances from a query location and the selected locations' popularity among the friends. A recent work on Geo-Social Temporal Top-k query [142] ranks the retrieved locations according to their spatial and social relevance within a time interval. The computation of the relevance scores of these approaches is based on the given query location of a user, and does not exploit the socio-spatial features of a network (e.g. social diversity). Additionally, there exist some other works on socio-spatial queries such as location prediction [63, 89, 90] in social networks. The works investigate the user relationship and spatial information to infer location for a query user. Various personalized location recommendation queries [10, 163, 172] consider location preferences with similar users. For example, Zheng et al. [172] recommend locations from friends' location histories such that the users can discover the places that interest them. However, none of these works well exploit the characteristics of geographical social engagement.

On the other hand, the Geo-Social Keyword (GSK) [3] search query enables the retrieval various information from the network including users, POIs, or keywords that satisfy spatial, social, and textual criteria. The Circle of Friend Query (CoFQ) [95] allows searching a group of friends in a Geo-Social network who are close to each other both socially and geographically. More specifically, the members in the group should have strong social bonding and should be within a small region. One similar work to CoFQ [95], Shim et al. [140] propose a geo-social query called the k-Nearest l-Close Friends query, which retrieves the k nearest objects among the l-hop friends of the query user. The integration of the social factors into spatial keyword query processing has been studied in [156]. The authors propose a query that enriches spatial keyword query considering the social relevance. The approach in [156] returns top k objects related to the query keyword. Therefore, the majority of the existing works on geo-social query consider to retrieve a set of users or objects for a given input query. In another work, a general

geo-social ranking (GSR) framework is proposed in [8] to rank the top-k users w.r.t. social and spatial importance in the selected network. Like [156], the GSR framework also returns the relevant set of friends of top-k users which are close to a query. Additionally, this work employs different ways to combine diverse concepts of spatial and social aspect of various applications.

2.2.3 Diversified Object Selection

The importance of result diversification has been recognized on information retrieval [2]. The diversity among the objects has been extensively studied to improve the object selection problems (e.g. [17, 39, 49, 120]). Diversity among the selected data is necessary to different types of objects in the search results. The existing studies on diverse object selection problems expand in a wide variety of spectrum, e.g., diversified keyword search on documents [5], diversified keyword search over graphs [56], diversified query recommendation [176], etc. There are various definitions of selecting diversified objects which mainly depend on the content dissimilarity [178], information diversity [27], categorical diversity [2]. In [2], Agrawal et al. consider both the relevance of documents and diversity of the search results to retrieve objects. There also exists several greedy solutions [5, 14, 57] that build the diversified result set in an incremental way. Angel et al. [5] propose, `DivGen`, a content-based diversification algorithm which first computes the relevance of each document, and then updates the usefulness of all other documents based on the similarity to the highest scoring document. It ranks the search results based on both the relevance and the dissimilarity to other reported results. Drosou et al. [39] propose a dynamic diversification approach using index on cover trees to select items that are both relevant and diverse to a query. Another diversified query search framework was proposed by Qin et al. [120], where datasets are transformed into Diversity Graph using node properties, and the selected diversified nodes have maximum total score with no two nodes are adjacent. The Maximum Marginal Relevance (MMR) function [16] maximizes relevance and diversity of a set w.r.t. a query element. Variations of MMR are considered in several domain specific greedy-based approaches [36, 42, 149, 173]. These greedy-based approaches are monotone and generate the answer set by adding elements one by one in non-increasing order of their scores. The process stops when an approximate solution containing k elements is identified. The problem of diversifying continuous data has been considered in the studies [36, 38, 102] using the variations of MaxMin, MaxSum models.

2.2.4 Spatial Object Selection and Map Generalization

Works in this category are related to map services, POI selection problems. Existing map services retrieve a subset of spatial objects based on the relative weights of the retrieved objects that maximize the total weights [32]. Nutanong et al. [113] define the problem of sampling large geo-spatial dataset in a region of user interest. Mahdian et al. [99] propose POI selection

problem, that targets to identify a set of POIs with maximum utility according to some reference POIs. Meanwhile, DisC [37] essentially selects the subset of diversified objects, where two selected objects must be at least r distance from each other, and there should be at least one object (un-selected) in the dataset within r distance from a selected object. On the other hand, the Spatial Object Selection (SOS) [61] model selects k diversified objects in such a way that any two selected objects must be at threshold distance from each other and the aggregate similarity (computed based on semantic attributes) from the selected set of objects to the whole dataset is maximized. However, these works do not consider any social factors e.g., social relevance and social diversity. The goal of map generalization is to produce a map at a given scale that achieves the right balance between rendering performance and information quality for end users. An important task of generalization method is to select subsets to be shown at different zoom-levels in a map, subject to a set of spatial constraints [76]. The target of spatial object selection is to determine the appropriate samples of data to be displayed on specific geographical region with different zoom levels. The existing map service retrieve a subset of spatial objects based on user query and show them on the map according to their weights.

Sarma et al. [32] study spatial sampling of large geographical data displayed on predefined geographical region and zoom level on a map. This study is related to map thinning problem and the objects which can maximize the total weights in a map region are selected. The selected optimal location set maximizes objective functions among the other set of locations. The works in [99, 133] deal with selection and transformation of geographic features on a map so that certain visual characteristics are preserved at different map scales. Peng et al. [117] consider similar zooming constraints like [32] where location labels of old window will be visible to the new zooming window if selected. Kefaloukos et al. [76] extend the work of [133] by applying various constraints including proximity and visibility. The approaches [76, 99, 133] are offline computation based and they pre-compute the object selection for all geographical cells of different zoom levels. However, in real-world a user's region may not be fixed to a particular geographical cells and hence the pre-defined cells may not provide a good solution to the spatial object selection problem.

Nutanong et al. [113] define the problem selecting distinct data entries in a dataset and the selected data have the higher relative importance in comparison to other data entries in the proximity. They propose an ensemble of interrelated indexes and SQL sub-queries to achieve several features like visibility constraint and degree of spatial distinctiveness of each entry in a query window. However, it does not explicitly address the interactive data exploration when zooming or panning operations take place in the maps. Mahdian et al. [99] propose optimal selection algorithm for points of interest (POIs) selection problem that targets to highlight a subset of POIs with the maximum user-defined objective function score. The score of the selected POIs are calculated based on the associated value of each location. However, the work considers spatial distance and individual location scores for the subset selection task. This work

do not have control to select a certain number of POIs, rather the selected POIs are dependent on the distance threshold only. Another work in subset selection by Kefaloukos et al. [76] measures the "loss of importance" due to a location point that are removed from the original set. The problems in the above mentioned studies are different from our problem as our work that considers the representativeness of geospatial objects and various socio-spatial visibility constraints. Furthermore, we explicitly address the consistency constraint to support visualized exploration as users navigate a map.

2.3 Group Queries in Social Network

Group Queries in Social Network have been extensively studied in the literature. Majority of the existing works in this category can be categorized into community detection, and community search problems. In general, in community detection tasks, no explicit inputs are provided as searching criteria, whereas the community search problem finds communities based on certain queries as input. Such input queries may be keywords, locations, or even users. For example, given a location point as a query node, the community search problem finds user groups whose members are intensively connected and spatially close. In the following, we summarize the existing works on group queries in social network using the following categorization.

2.3.1 Group Queries based on Social connectivity

The social connectivity based group queries aim at finding user communities in a social network where each user should have minimum number of relationship edges in the selected group. A user community in social network provides important insights into the organization of networks and related hidden information of social networks [55]. Meanwhile, some of the works in this category may not consider any specific query node as an input, known as *community detection* [28, 48, 108] problem. These works discover cohesive communities from the whole network satisfying social constraints. Different from community detection, the *community search* [30, 31, 45, 70, 91] methods aims at finding maximal connected user group containing some user provided query. Community detection aims at finding all the communities in a graph based on certain criteria such as modularity [1, 80, 107]. Another common approach of community detection is based on topic models [97, 106]. In [129], the underlying topics, interaction types between the members, and the social connections are considered for discovering communities. However, the community detection approaches can take a long time to find all the communities in a large graph. Therefore, the approaches are not so effective for an online retrieval of communities. On the other hand, the community search approaches are query-based, and they can able to retrieve communities online. In community search problem, the query may contain a single [30, 31, 44, 45, 70, 144] or multiple [54, 77, 138] query nodes based on the

application requirement. To measure the structure cohesiveness of a community, the minimum node degree is used as the general constraints [30, 144]. For example, Sozio et al. [144] search for the community containing a query node q and each member of the output community must have at least k links. Cui et al. [31] search for cohesive community containing the query vertices and maximizing the minimum degree of each vertex in a community. Meanwhile, different adaptation of searching communities based on minimum degree metric are known as *k-core* [31, 44, 45, 77, 153], *k-clique* [30, 65], *k-truss* [20, 69, 70], etc. Some recent works [88, 91, 92] find influential community in large social networks capture the influence of a community. For example, Li et al [91] propose a community search algorithm based on the concept of k-core where each member of the community should have large influential weight. The authors rank the communities using internal influential scores. In [88], Li et al. consider k-clique as structural cohesiveness metric and outer influence score as a goodness measure of the communities.

2.3.2 Attribute-driven Group Queries

The other group search queries consider various attributes in the network such as keywords, locations, etc. An attributed community enables a better understanding of how and why a particular community subgraph is formed. For example, Fang et al. [45] consider both the structure cohesiveness and keyword cohesiveness constraints to search for the user groups. The authors claim that considering keywords provide ease of interpretation for the construction of a community. Guo et al. [60] study a spatial keyword query, m-closest keywords (mCK), which retrieves a group of similar objects and cover a set of keywords in the geo-textual object database. For a given set of keywords, the query mCK [60] finds a group of objects that can cover all the query keywords such that the diameter of the group is minimized. Another work on keyword aware community search approach is proposed by Islam et al. [73]. The work targets to find the most influential communities from an attributed graph enriched by word-embedding based keyword similarity model. The influential community in [73] is defined as the cohesive group of vertices having some dominance over other groups of vertices with a set of keywords related to query. In [69], Huang et al. study an attribute-driven community search based on k-truss that considers pairwise distance of nodes using the attributes associated with each user. Chen et al. [22] study a profiled community search problem in an attributed graph.

On the other hand, the spatial information has been used as attributes of social user nodes to find a community [20, 45, 153]. Fang et al. [44] propose a Spatial-aware Community (SAC) search to find a k-core within a covering circle containing the query node, and having a minimum radius. In other word, given a query vertex, the SAC search returns a set of vertices containing the query node that are close structurally and spatially. The maximum co-located community (MCC) [20] search uses the socio-spatial network information to find a community that maximizes the number of user nodes such that each user pair resides within a certain distance. The work

identifies the largest k-truss community by considering both the social and spatial cohesiveness. The spatial cohesiveness is measured by the pairwise distance between a user pair that should be lower than a threshold specified by end user. In [153], Wang et al. investigate the problem of computing the radius-bounded k-cores that finds cohesive subgraphs satisfying both social and spatial constraints. For a given query node and a radius, the work finds all k-core cohesive subgraphs containing the query node within the circle of the given radius.

In attributed graphs, user engagement and similarity are measured to find cohesive subgraphs. Zhang et al. [168] propose a community search approach that aims to find maximal (k, r)-core subgraph where the concept of k-core is introduced to guarantee the user engagement, s.t., each user in the resultant group must have at least k connections. Meanwhile, the similarity (calculated using user attributes) between node pairs of the selected social graph should exceeds a given threshold r. By identifying the user groups with high quality in terms of the engagement and similarity, the (k, r)-core groups have good potential to become active and stable. In other works, Zhang et al. [167, 169], study a cohesive subgraph model that considers both user engagement and tie strength to discover strong communities. The proposed models identify the user groups with the actively engaged users having strong tie strength.

2.3.3 Socio-spatial Group Queries

The socio-spatial group queries [44, 138, 161, 174] aim to find the 'best' group against a point-of-interest (POI) or query user, where the members possess social tightness within the group and have spatial closeness to a POI. For example, Fang et al. [44] propose spatial-aware community (SAC) search to discover socially and spatially cohesive user group which are near to a query user. The resulting community must be a connected k-core where the members are located within a spatial circle of having a minimum radius. In [161], Yang et al. propose a socio-spatial group query (SSGQ) that finds a set of users where each member may have at most a certain number of acquaintance users in the group. The SSGQ needs a fixed location as query point and the aggregated distance between the selected users' locations and the query location should be minimized. Using the concept of SSGQ problem, Shen et al. [138] propose multiple rally-points socio-spatial group query (MRGQ) that selects the best rally-point among the multiple location points provided in the query, and returns the best corresponding user group that minimizes the spatial distance between users' locations and the best rally point. The resultant user group in MGRQ is of fixed size and the members of the user group satisfy the minimum familiarity constraint. All the above mentioned methods, e.g. SAC, SSGQ, and MRGQ returns a fix number of users within a fixed spatial radius of the query point. MRGQ selects most suitable activity location from a set of candidate locations, whereas, SAC and SSGQ consider single query user's location. Both the SSGQ and MRGQ methods consider the familiarity constraints but SAC is strict to the social connectivity. In another work, Zhu et al. [174] propose geo-social group

Table 2.1: Comparing existing works on group search queries

Parameters	GCS [77]	FSSGQ [54]	MCC [20]	SAC [44]	(k,r)-core [168]
Check-in Graph	Yes	No	No	No	No
Input Query	locations, users, or both	locations	-	user	-
Location per User	multiple	single	single	single	single
Output	community and location cluster pair	top-k groups of users	communities	community	maximal (k, r)-cores
Objective	maximize check-in density	maximize group score	maximize size	minimize radius	maximal (k, r)-cores

queries (GSGQ) with minimum social acquaintance constraints which guarantees worst-case acquaintance of each user.

Inspired by the work on multiple rally-points socio-spatial group query [138], Ghosh et al. [54] propose flexible socio-spatial group query (FSSGQ) that also consider multiple location query like MSGQ [138], but returns a list of top-k groups (instead of one group as in MSGQ), and their corresponding best locations that minimize the aggregated distances to the query locations. The groups are further ranked using score of each group using geo-social scoring function [8] that considers the social connectivity score, spatial closeness score, and group size score. In FSSGQ [54], the group size is not fixed, rather it can return a flexible number of users based on application requirement. However, this work do not consider the engagement of social users in the spatial space and also consider that each user in the network is associated with a single location. In a recent work, Chen et al. [21] propose a geo-social group search query, MKCSSG, that considers textual, social and spatial information as graph data, where each user node in the network is attached with a set of keyword attributes and a location. Given a set of query keywords φ, a keyword capacity parameter r, the proposed geo-social group search satisfies the social, spatial requirements with minimum keyword and capacity constraints.

In a recent work, Kim et al. [77] propose GeoSocial Community Search problem (GCS) which aims to find a social community and a cluster of spatial locations that are densely connected in a location-based social network simultaneously. The work focuses on finding a community and the location cluster pair, consisting of a group of users and a set of locations, respectively. The users in the result community should also be firmly connected socially and their visiting locations should be close spatially. The authors model the closeness in the result with structural/spatial constraints. The inputs of the work were LBSN graph and set of query nodes that the users in the subgraph are socially close and they frequently check-in to a cluster of closeby locations. In particular, the work maximizes the check-in density between the user group and location cluster containing all the query nodes. Here, the query nodes may be users, locations, or/and both. Based on the various properties, including input query, objective functions, output, etc., we compare the existing models on geo-social group search problem in Table 2.1.

Chapter 3

Location Prediction in Large-scale Social Networks

Location details of social users are important in diverse applications ranging from news recommendation systems to disaster management. However, user location is not easy to obtain from social networks because many users do not bother to provide this information or decline to do so due to privacy concerns. Thus, it is useful to estimate user locations from implicit information in the network. For this purpose, many location prediction models have been proposed that exploit different network features. Unfortunately, these models have not been benchmarked on common datasets using standard metrics. We fill this gap and provide an in-depth empirical comparison of eight representative prediction models using five metrics on four real-world large-scale datasets, namely Twitter, Gowalla, Brightkite, and Foursquare. We formulate a generalized procedure-oriented location prediction framework which allows us to evaluate and compare the prediction models systematically and thoroughly under extensive experimental settings. Based on our results, we perform a detailed analysis of the merits and limitations of the models providing significant insights into the location prediction problem.

Chapter map. In Section 3.1 we provide an overall introduction of the location prediction task in social network. We generalize the problem of location prediction and sketch out the existing works in Section 3.2. Next, we propose a universal framework on location prediction of social users in Section 3.3. In Section 3.4, we map each individual model to the framework without any loss in their accuracy. We conduct extensive experiments to compare these models under similar configurations using various metrics and different data settings in Section 3.5. Finally, we conclude our study in Section 3.7 by giving some interesting insights into the existing location prediction models in Section 3.6.

3.1 Introduction

In this chapter, we study the location prediction problem by exploring the implicit information available in the social network. Social networks provide elementary means for declaring spatial information through (1) self-reported context, and (2) GPS-enabled geo-tagging of posts and check-ins [124]. There is significant interest in predicting user locations through public posts, metadata, and network information. Many researchers have focused on predictive algorithms to infer the locations of social users [13, 24, 66, 100, 128, 160]. Some of these leverage the user-generated content (UGC) from the social stream [24, 66, 160] to predict users' locations using *location indicative words* (or "local" words) available within the users' GPS-tagged posts. The prediction performance of these models depend on the availability of local words in post contents. However, the location information in user-generated posts are too limited. Ryoo et al. [128] report that only 0.4% of tweets (collected from the Korea region) have some GPS-tagged location information. In another study, Hetch et al. [66] report that only 0.77% of tweets among global users have some location information. Therefore, content-based location prediction approaches may not perform well due to the sparseness of location indicative words in users' posts. Hence, instead of using social contents, some prediction techniques [9, 74, 127] rely on the graph structure of a social network. These techniques exploit the network features while inferring users' locations using their social connections. For example, Backstrom et al. [9] assume that an unlabeled user is co-located with one of their friends in the network and a location is estimated by maximizing likelihood of their friends' locations. McGee et al. [101] integrate various social factors (e.g. number of followers) for the location prediction task.

Hybrid prediction models [89, 90, 121, 124], on the other hand, exploit both the user-generated contents as well as the network information. If some neighbors (i.e. followers, friends) provide locations in their profile, or they mention some places in their posts, the hybrid prediction models can use such information to predict locations of unlabeled users. However, these models have the flexibility to use either one or both information types. Neural network based geolocation prediction models are reported in [103, 122]. Recently, some probabilistic frameworks [35, 119] are proposed, which consider features learned through deep learning from social contexts. Apart from "user location" prediction, some studies focus on predicting other types of locations such as *post* (e.g. tweet) location, *mention* location, and *work* location. Meanwhile, the majority of the available works on predicting location of "posts" [23, 24, 93, 126, 130] rely on the social contents. The "mention" location prediction models [53, 85, 86, 96] extract textual fragments in posts that observe some location names. There exists some work [23, 26, 166] aims at predicting location types such as work place, or supermarket. Cho et al. [26] consider the temporal and social information to distinguish home locations and work places. Pang et al. [115] propose a feature learning framework based on deep learning, and it can predict user demographics and location category. Other notable work in predicting the next place visit of social users are

available in [135, 177]. However, in this study we are mainly focusing on the tasks of stable "user location" prediction using the network information. Additionally, we do not consider machine learning based location prediction models in our benchmark study which are heavily dependent on the quality of training datasets.

Based on the discussions above, we can understand that there exist varieties of location prediction models. The majority of the existing models are based on different internal configurations that best suit their targeted applications, and hence it is difficult to analyze, compare, and evaluate their suitability in a common base. In this work, we compare models for *stable* 'user location' predictions in social media. A *stable* location is defined as the long-term residential address (e.g. city level) or location where the majority of the activities are performed. Our main aim is to compare location prediction models that take the *network* features as input and predict users' *stable* locations. We also test whether the existing models can explain the observed data adaptation in Twitter microblog as well as in other Location based Social Network (LBSN) (i.e. check-in) datasets including Gowalla, Brightkite, and Foursquare. We assume that the majority of the activities of a user occur near to her stable location [11, 26, 84]. Meanwhile, locations may require different granularity given the specific application needs. For the sake of standard evaluation, we choose a uniform granularity level i.e., city level user locations. From here onward, we simply use 'user location' instead of '*stable* user location' for brevity.

We divide existing models into four major categories (details to follow in Section 3.2.2) based on the prediction approach they use, and from each category, we choose representative models in a unified framework (see Fig. 3.1) to perform comparative analysis. Our aim is to gain insights into the general approaches (of the four categories) as well as the specific algorithms selected for comparison w.r.t. multiple aspects. Specially, we perform experiments on four social media datasets with different levels of location sparsity, and compare the performances of the models with various user-centric and model specific configurations. These evaluations give us novel insights into the relative merits of the specific location prediction models.

3.1.1 Challenges

The process of benchmarking location prediction models poses three major challenges:

- Due to the large diversity in existing models, it is difficult to abstract a unified benchmarking framework. It is critical to understand and diagnose the existing models from a common viewpoint.
- For a fair comparison and in-depth analysis of different location prediction model types, it is essential to apply these models on the exact same datasets. This requires the software implementation of these models which are not publicly available. Re-implementing the representative models on a common coding platform and setting their parameters is a daunting task.

- Previous researchers have tested their approaches using a small number of metrics with limited scopes leading to the possibility of incomplete views of the model's performance. It is important to identify a suite of metrics that can evaluate multiple aspects of the location prediction outcomes of all existing models. Defining such a suite of metrics is a challenging task.

In our study, we address the above mentioned issues and conduct a systematic in-depth benchmarking study by comparing eight location prediction models on four real-world datasets with different essential settings. We present several comparisons using different location sparsity levels, geographical region specific predictions, agreements between model pairs, and the impact of user-centric information in location prediction tasks. Our comparisons give significant insights into the models' performances under various user- and data-centric settings.

3.1.2 Generalized Procedural Framework

We propose a generalized benchmarking framework for the location prediction problem. We implement eight different representative models and evaluate them on the prediction task: "given a social network and information about geography, infer the locations of the 'unlabeled' users". The 'unlabeled users' is defined at Definition 1 in Section 3.2.1. Our benchmarking framework consists of four core components as shown in Figure 3.1, such as: (1) **Setup** includes a set of location prediction models, real-world datasets, parameter configuration, and the social graph generated from the datasets; (2) **Prediction Framework** presents a generalized location prediction module, with deep cogitation of the common work-flow in the location prediction framework (see Section 3.3 for the framework details, and Section 3.4 for the mapping procedure); (3) **Diagnoses** discusses the key factors that affect the prediction performance of these models; (4) **Evaluation** provides a comprehensive evaluation module to verify and compare the models using both the dataset settings and the model-wise parameters. The structural components of our proposed framework are inspired from the benchmarking framework designed for community detection [154]. However, the internal functions of the components such as model driven procedure, evaluation strategies, initial setup configurations, and the diagnoses approaches of our framework are very different. Moreover, in our study, we have implemented all the selected models in a common code base in a similar software environment allowing for a fair comparative analysis.

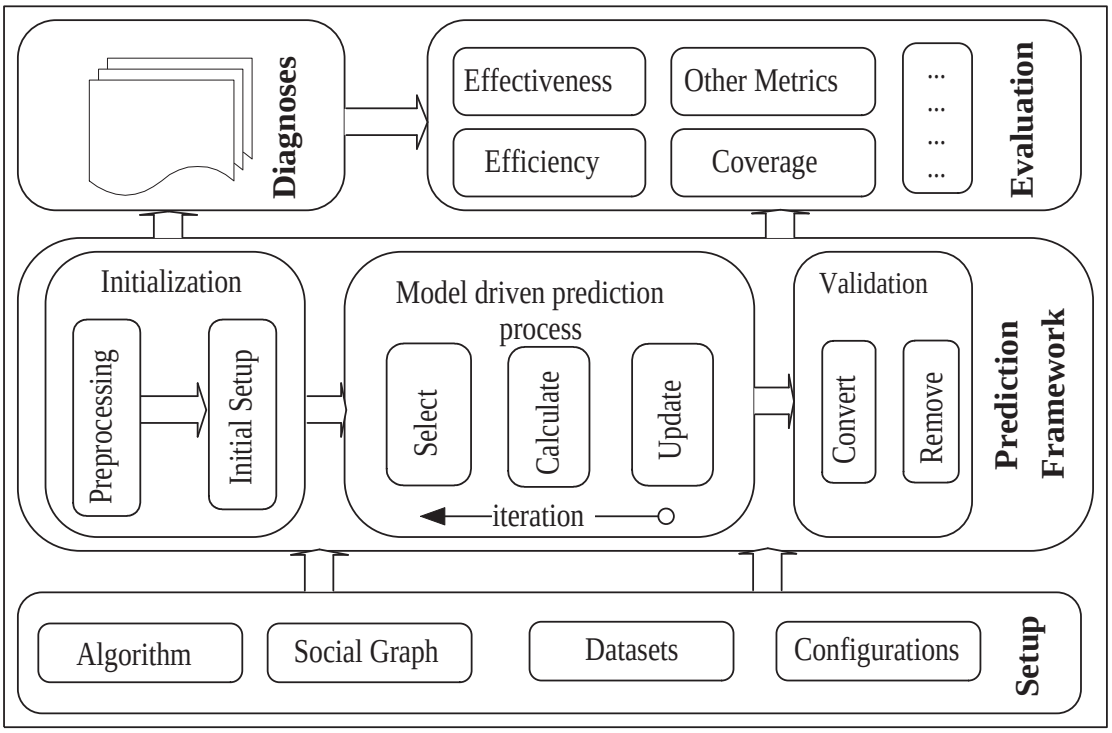


Figure 3.1: The proposed benchmarking framework

3.2 Preliminaries and Background

3.2.1 Preliminary

Since the majority of the models were originally tested on the Twitter data, they used Twitter related terminologies in their discussions. However, in this thesis we use different types of social media datasets and hence, generic terminologies, i.e. 'message' or 'post' instead of 'Tweet', will be used in this thesis.

Definition 1 (Social Networks)**.** *A social network is a mathematical structure consisting of a set of entities (i.e. social users and locations) and their relationships. We define it as* $G(V, E, L, T)$ *where,*

- V *is the set of social users. It includes the labeled* (V^*) *and the unlabeled users* (V^N) *i.e.* $V = V^* \cup V^N$*. The 'labeled' users* ($u_i^* \in V^*$) *are location annotated users who have disclosed their locations in profiles. In some check-in datasets (e.g. Gowalla, Brightkite) if no profile locations are available, we can choose a 'single' representative location among the multiple check-ins (discussed in Section 3.5.2) of the users where a majority of the activities occur. These locations are used to annotate the 'labeled' users in check-in datasets. The remaining users, i.e.* $(V - V^*)$ *are considered as unlabeled users* (V^N)*.*

- L *is the set of locations available in social network which contains the users' profile location, check-in locations, and locations available in users' posts (e.g. Tweets).*

- *E is the set of directed edges $e\langle v_i, v_j\rangle$ from v_i to v_j, which consists of 'following relationships' $E_F : \{V \times V\}$ and 'messaging relationships' $E_T : \{V \times L\}$. Edge $e\langle v_i, v_j\rangle$ is written as $f\langle i, j\rangle \in E_F$ where user $u_i \in V$ follows another user $u_j \in V$, or as $t\langle i, j\rangle \in E_T$ where user $u_i \in V$ mention a location $l_j \in L$ in her posts. If some datasets do not have any user posts, the 'messaging relationships' edges will be absent in G. In Twitter terminology, the 'messaging relationships' is similar to 'twitting relationships'.*

- *T represents the set of posts (or messages) posted by V in G. The messages can be user posts, replies, or even forwarded messages (e.g. re-tweets). If a dataset do not have any message contents, the corresponding tuple of graph $G(V, E, L, \phi)$ remains null.*

Different geo-location models consider different types of inputs, e.g., content, network, and contextual information. Following relationships (f_1) and user location (f_3) are the main input features of network based location prediction models. The other features such as messaging relationship (f_2), message or post contents (f_4), mentioned location frequency (f_5), social tie and closeness (f_6), have been used in different prediction models.

Table 3.1 lists the features that are used by the prediction models. The last column presents the time complexity of each model, 'm' being the number of iterations, 'k' the average number of labelled neighbors, and 'c' the number of partitions for the tree regressions used in [101]. The tree regressor divides the dataset into smaller partitions to sort out the best contacts for the location prediction.

Table 3.1: Features and Time Complexity of the Models.

Models	f_1	f_2	f_3	f_4	f_5	f_6	Complexity
UDI [90]	✓	✓	✓	✓	✓		O($m\|E\|$)
MLP [89]	✓	✓	✓	✓			O($m\|E\|$)
Backstrom [9]	✓		✓				O($\|V\|k^2$)
SLP [74]	✓		✓				O($\|V\|k^2$)
TFIDF [82]			✓	✓	✓		O($\|V\|\|L\|$)
Friendly [101]	✓		✓			✓	O($\|V\|ck^2 + k\|V\|log\|V\|$)
SPOT [78]	✓		✓			✓	O($\|V\|k^2$)
LMM [159]	✓		✓			✓	O($\|V\|k^2$)

Definition 2 (Location Type)**.** *In social media, there are three types of locations, i.e. location of 'post', 'mentioned' location, and 'user' location. The 'post' location is generally available in the geo-tags of a post (e.g. tweet). 'Mentioned' location refers to the locations available in post contents, whereas 'user' location is available in self-reported profile and other check-in activities. Such locations may be home location, work location, or favorite location. We consider 'stable' user location (e.g. home location) at city level where the majority of user activities occurs.*

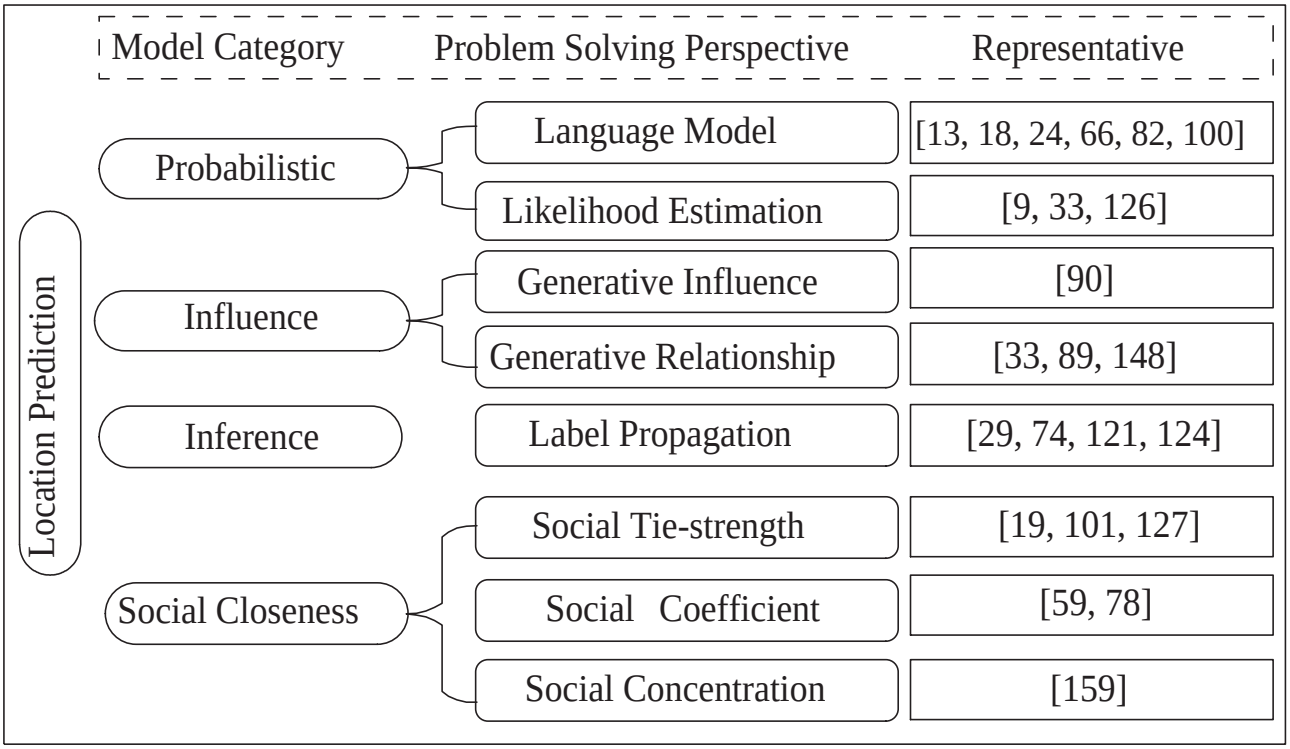


Figure 3.2: Categorization of location prediction approaches

Definition 3 (Location Prediction Problem)**.** *Given a social network* $G(V = V^* \cup V^N, E, L, T)$*, location prediction problem is to label a set of users* $\hat{V}^N \in V^N$ *with the locations selected from set* L *using a specified prediction model* M_x*, such that the predicted location* $\hat{l}_{u_i}$ *of* $u_i \in \hat{V}^N$ *is close to the actual location* l_{u_i}*.*

3.2.2 Prediction Models and Algorithms

We focus on the fundamental problem of location prediction that aims to identify the locations of unlabeled users as precisely as possible. In most studies, the stable user locations are predicted at city-level, state-level or sometimes at the country level. Existing models have used three main input types namely, contents, network, and context. The models are broadly categorized into either content- or network-based approaches. The hybrid models, on the other hand, use both the content and network information simultaneously. However, we categorize the existing models based on their key approaches rather than the input type. For each category as shown in Figure 3.2, we re-implement their representative models.

Probabilistic Approaches. The models in this category examine the probability distribution of different characteristics in the social network. *Language Model* [13, 18, 24, 66, 82, 100, 160], a sub-category of probabilistic approaches, analyzes the text-based contents of labelled users and build a 'language model' (LM) using location indicative words (e.g. local words) available in users' posts. The local words have strong correlation with a specific location. These models

calculate the word distribution extracted from the labelled users' posts (e.g. tweets) and scores of locations are measured using probability distribution. Hecht et al. [66] and Yamaguchi et al. [160] calculate CALGARI and KL-divergence scores of words respectively to identify the local words in social contents. Mahmud et al. [100] apply heuristic rules to identify such words, whereas Cheng et al. [24] propose a model-driven approach based on the observed geographical distribution of the words. We choose *TFIDF* [82] model as the representative language model where a similar approach is considered to build a language model based on word distribution. A word is considered as a 'local' word to a location if the corresponding *tf-idf* score is higher than a threshold parameter.

The *likelihood estimation* approach [9, 33, 126] estimates the location of an unlabeled user by constructing a probabilistic model, and measures the likelihood of users' friendship within a distance. Ren et al. [126] assume that if the majority of the user's friends live at a particular area, there is a higher chance for the user to co-locate with their friends. The *Backstrom* [9] model is the base model in this category and we consider it as the representative model.

Influence-based Approaches. This type of model captures influence scope of nodes (i.e. user, location) by considering the relationships factors of social users and locations. The *generative influence* based approach [90] integrates the social network and user-centric data in a generative framework to model the '*following*' and '*messaging*' relationships jointly. It predicts user's location by calculating the influence probability of how likely they follow other users or mention a location in their posts. *Generative relationship* based approaches [33, 89, 148], in this category, measure the probability of various relationships using structural and spatial properties. Davis Jr. et al. [33] consider the structural relationship where the popular locations among the friends are considered as the location of a user. Li et al. [89] propose a generative approach that models the probability of generating '*following*' and '*messaging*' (i.e. twitting) relationship based on users' location. The most likely locations are assigned to an unlabeled user using the relationship probability of followers and mentioned locations.

We choose *UDI* [90] and *MLP* [89] as representative models of *generative influence* and *generative relationship* sub-categories respectively. These two models use similar features (*following* and *messaging* relationships). However, the approaches of observing such relationships have significant differences as below:

(1) A *Generative influence* based approach (e.g. *UDI*) calculates influence scope of nodes (i.e. user, venue) using Gaussian distribution and models them in a generative way. On the other hand, *Generative relationship* based approaches directly observe *following* and *messaging* relationships and models them using power law and multinomial distributions respectively (e.g. *MLP*).

(2) A *Generative influence* based model (e.g. *UDI*) iteratively computes location of a user and update the influence scope of her neighbors and mentioned places. This process continues until the likelihood converges. On the other hand, *Generative relationship* based model (e.g. *MLP*)

calculates the joint probability of the observed (e.g. *following* and *messaging*) relationships, and the maximum likelihood locations are assigned as the inferred locations.

Inference-based Approaches. These approaches are based on semi-supervised iterative algorithms which consider the spatial distribution of locations and infer a suitable location based on the social relationships. The Factor Graph Model [119] uses location inference techniques to propagate the labeled locations by incorporating the deep features learned from the social context. The *Label Propagation* method [29, 74, 121, 124] infers location by spatially propagating locations using the neighborhood information. Rahimi et al. [121, 124] propose a hybrid approach which combine logistic regression with network-based label propagation to improve the location predictions. Both the models [121, 124] use label propagation technique as the main prediction approach, but the initial estimation of user location is made by text-based geolocation techniques. For example, the model [124] considers logistic regression model prior for test users and the similar label propagation approach as *SLP* [74] is used to infer users' locations using updated median of neighbors' locations. Among the existing label propagation models [29, 74, 121, 124], the *SLP* [74] is the basic one and have extended the label propagation concept of [175]. We choose *SLP* [74] as the representative model of this category.

Social Closeness-based Approaches. The models in this category consider different network properties such as friendships, interactions, social trust etc. to estimate the locations of users. These models are based on the concept that social closeness of two users is better indicator of home proximity. The *social tie-strength* based approach [19, 50, 101, 127] predicts the location of an unlabeled user considering the tie strength of the user and their labeled neighbors. Various social relationships like friendship, user mention, node degree are used to measure the tie strength. We select *FriendlyLocation* [101] as the representative model of this sub-category. The *social coefficient* based models [59, 78] in this category measure the closeness of two users based on their quantitative neighbor information. Gu et al. [59], proposed the concept of social trust to measure the closeness in the social structure using the number of common friends. Kong et al. [78] propose *SPOT* model that calculates social closeness using cosine similarity between a user pair. We consider *SPOT* [78] as the representative of *social coefficient* based model sub-category. The *social concentration* based model (e.g. *LMM* [159]) infers locations from neighbors who have the higher spatial concentration with their social connections. The representative models of each category are discussed in Section 3.4.

3.3 The Generalized Procedure

We propose our benchmarking framework and use a generalized procedure to map the functionalities of eight location prediction models. To re-implement the existing models from a common view point, we formulate an adaptive procedure so that the models can be adjusted easily in different types of social networks. Our framework comprises three main phases namely, *Initial-*

Algorithm 1: Generalized Location Prediction Procedure

Input: Social graph $G = (V^* \cup V^N, E, T, L)$, Model ($M_x$, $\mathcal{P}$)
Output: $\hat{V}^N$

```
1  Initialize: P ← Φ;
2  for each user u_i in V^N do
3      Seq ← (u_i, FeatureInit(u_i))
4  PreCompute(G, P, M_x);
5  while Iteration != Iteration_Max do
6      for each user u_i in Seq do
7          Select(G, M_x, Seq);
8          l_{u_i}^{tmp} ← Calculate(u_i, G, P, M_x);
9          V̂^N ← Update(l_{u_i}^t, l_{u_i}^{tmp}, V̂^N);
10     Iteration++;
11 Validate(G, l_{u_i});
12 return V̂^N
```

ization, *Model driven location prediction process*, and *Validation* in the "Prediction Framework" as illustrated in Figure 3.1. These phases are the key steps to characterize the generic procedure of the location prediction models. Algorithm 1 shows the generalized procedure of the proposed framework and the details of the phases are discussed below.

3.3.1 Initialization Phase

The *Initialization* phase initializes the primary configuration parameters of the models. We extract the model specific features using FeatureInit() method (Line 3 of Algorithm 1) and create user prediction sequence *Seq*. We initialize the maximum number of iterations, $\text{Iteration}_{\text{Max}}$, as suggested by the original authors and set the parameter to 1 for the models which do not have multiple passes (e.g. *TFIDF*, *LMM*). Some models need to pre-calculate some parameters in PreCompute() (Line 4), such as power law distribution parameters in *MLP* [89].

3.3.2 Model driven Prediction

Initialization is followed by the prediction process. We abstract the three common key steps including Select, Calculate, and Update at Lines 5 - 10. In Select() method, we pick the unlabeled users one by one, and the corresponding features of each user are loaded. The Calculate() method infers a location to the unlabeled user $u_i \in V^N$. Finally, the method Update() assigns a new location by replacing any previously predicted geo-points. The three methods, i.e. Select, Calculate, and Update iterate until the termination criteria of the respective model are met.

3.3.3 Validation Phase

In the final phase, we transform the geo-points into our predictable location type using the nearest city name. Some existing models assign geo-points to the predicted users, while the other return city names. This step ensures that the locations are validated consistently. Some model may return a 'null' value corresponding to the users locations. Such invalid locations are removed in this step and the corresponding users remain unlabeled.

3.4 Within Framework Implementation

We recapitulate eight representative models with necessary adaptations to the proposed generic framework. The detailed descriptions of the models are given below:

3.4.1 Probabilistic Language Model

Language Model based location prediction models characterize word distributions in users' texts (i.e. posts) and follow a *probabilistic* approach to infer users' locations. The models in this category construct a Language Model (LM) using 'local words' available in labeled users' posts. Local words are tightly coupled with semantic locations. Though, the Language Model based prediction approaches use the social contents to predict users' locations, we consider such models to compare with the other network-based models. Significant amount of research (e.g. [13, 24, 82]) in this category have been carried out on identifying 'local words'. For example, Bo et al. [13] proposed a model based on Inverse Location Frequency (ILF) and Inverse City Frequency (ICF) to measure the probability of words in a location. A representative probabilistic model proposed by Cheng et al. [24] uses the distribution of user's home location l with the post (tweet) contents. Given a set of words w extracted from user u's posts $T(u)$, the probability of the user u being located at location l is calculated as,

$$p(l|T(u)) = \sum_{w \in T(u)} p(l|w) * p(w) \tag{3.1}$$

Here, $p(w)$ is the probability of word w in the dataset and is calculated using the occurrence of the word w in the local word dataset. We consider *TFIDF* [82] as the representative model of this category. The mapping of this representative model in our framework is described as follows:

In PreCompute method, a language model (LM) is built using the location information of the labeled users and their corresponding post contents. The purpose of creating an LM is to compute the probability distribution of each location indicative words. The probability of a word w is calculated using Term frequency and Inverse Document Frequency (TF-IDF) in a

location l as,

$$p(l|w) = \frac{c(w,l)}{\sum_{i=1}^{n} c(w_i,l)}, \quad c(w,l) = \sum_{s \in post(l)} tf(w,s) \tag{3.2}$$

$c(w,l)$ calculates the total number of occurrences (term-frequency) of word w in the posts of 'labeled' users who have location l. In SELECT, each unlabeled user is chosen one by one, and in CALCULATE, the probability of a user u located at location l is calculated using Equations 3.1 and 3.2. A location with the maximum likelihood probability is considered as the predicted location. However, the content based probabilistic model may not perform well, as the availability of location indicative textual information is very rare in ordinary users' posts[66, 128].

3.4.2 Generative Influence based Model

The "Generative Influence Model" is based on modeling the influence scope of nodes in generative way. It follows a probabilistic approach to model the influences. The Unified and Discriminative Influence model (*UDI* [90]) considers both the influence of neighbors and the locations mentioned in their posts. This approach models user's influence as a bivariate Gaussian distribution and the variance of the distribution is interpreted as influence scope. Further, an iterative process is followed to update an unlabeled user's location using neighbor information. The newly predicted locations are subsequently used to estimate other users in the network.

The influence probability of a 'node' n_i at a location l is modeled using a Gaussian distribution (refer equation 3.3). It considers a 'node' as both a user or a location.

$$P(l|\theta_{n_i}) = \frac{1}{2\pi\sigma_{n_i}^2} e^{-[\frac{(X_{n_i}-X_l)^2}{2\sigma_{n_i}^2} + \frac{(Y_{n_i}-Y_l)^2}{2\sigma_{n_i}^2}]} \tag{3.3}$$

Here, θ_{n_i} is the node n_i's influence model and σ_{n_i} is the influence scope of n_i. Two types of influence models are generated to measure the probabilities of generating *following* and *messaging* relationships using equation 3.3. A location that maximizes the joint probability of generating such relationship edges with the labeled neighbors and mentioned locations, is inferred as the corresponding user's location. The generative influence model has two types of prediction methods. The *Local* prediction method observes the direct edges to infer the location of a user. The *Global* prediction method utilizes all relationships available in the entire graph and it allows to iterate multiple times until it converges. Initially, this model assigns unlabeled users with random locations, and then iteratively updates those locations using their neighbors' locations and mentioned locations.

Remarks. The *UDI* [90] model assigns a random location to the unlabeled users first. It follows multiple inner iterations to converge the assigned locations by updating influence scope of friends' and their mentioned locations. We notice that location prediction process in *UDI* using 'random' location initialization takes time to converge. To optimize the process, we initialize

the unlabeled users with the centroid of (at most) ten labeled neighbors' locations (rather using random locations). Such an approach has reduced 18% of the total inner iterations without affecting the overall accuracy. However, if a user has less than ten labeled neighbors, we consider all her labeled neighbors (to calculate centroid) to initialize the user. Meanwhile, we assign a random location to the unlabeled users if they do not have any 'labeled' neighbor.

3.4.3 Generative Relationship based Model

MLP [89] model, a representative of Generative Relationship based model category, uses a supervised extension of Latent Dirichlet Allocation (LDA) to model the relationship between users and locations. The *MLP* model considers the effect of noisy relationships generated due to influences of famous personalities (e.g. 'Lady Gaga') and popular venues ('Hollywood'). However, the approaches of considering the influences in a generative model is different from the Influence based Model (discuss in Section 3.4.2).

MLP calculates the likelihood of a user following spatially close friends and mentioning nearby places. Locations are observed from labeled users and explicit correlations between locations and *following* relationships are measured. The following probability at distance 'd' can be expressed as $P(d|\alpha, \beta) = \beta.d^{\alpha}$ and the values of the parameters α and β are learned using the labeled user information. Additionally, the location based messaging model captures the messaging relationships using the mentioned location information. The *messaging* probabilities are modeled as multinomial distribution. However, this component (e.g. location based messaging model) is not effective when there is no content based information available in dataset. However, some relationships may not be generated based on the location distances. The *MLP* model captures noisy and location-based relationships using random generative models that measures the probability of randomly following a user or tweeting a venue.

This model combines the discrete (power law) and continuous (multinomial) distributions in a non-trivial manner, and Gibbs sampling-based algorithm is used to estimate the location assignments. After obtaining the location assignments for relationships of each user, their corresponding location distribution θ_i is measured with the maximal likelihood estimation, $p(l|\theta_i) = \frac{\varphi_{i,l}+\gamma_{i,l}}{\varphi_i+\sum_{l=1}^{L}\gamma_{i,l}}$, where φ is the user location assignments and γ is the prior distribution parameter of θ. The locations with largest probabilities in θ_i are estimated as the 'stable' multiple locations of user u_i.

Remarks. MLP model can discover a user's multiple locations. We make a small addition to this approach to identify single 'stable' location among the estimated multiple locations. We select the closest location to the centroid of the 'multiple' predicted locations as the 'stable' location of the user.

3.4.4 Probabilistic Likelihood Estimation based Model

The location prediction models (e.g. Backstrom et al. [9], Davis et al. [33], Ren et al. [126]) in this category study the interplay between geographic distance and social relationships. We choose *Backstrom* model ('Back' in short), one of the primitive models as the representative of this category. Location inference begins by building a probabilistic model representing the likelihood of observing a relationship between the users when a geographic distance is given. Based on the location distribution of labeled neighbors, a user is assigned a location which has the maximum likelihood. This model assumes that the location distribution of a typical user does not have many friends at long distances. Although the original paper [9] mentioning *Backstrom* model is conducted on Facebook, we adapt this model to other social media like Twitter, Foursquare.

In a large social network, the probability of friendship is roughly inversely proportional to the physical distance between the social friends [9]. Given a distance 'd' between two users, the probability of having an edge (i.e. following relationship) between them is measured as $p(d) = a(b + d)^{-c}$. As mentioned in the original paper [9], the value of the constants $a = 0.0019, b = 0.196, c = 1.05$ are empirically determined using Facebook data. However, these values may vary in different datasets with different population distributions. For a given location l_u of user $u \in U^*$, if $L_v \in L(ngbr(u))$ are the locations of the labeled friends of u; the edge probability for each neighbor location is computed as, $p(|l_u - l_v|) = a(b + |l_u - l_v|)^{-c}$ s.t. $l_v \in L_v$. A location l_u co-located with one of u's friends is considered as the location of user u if the value of $\gamma(l_u)$ (refer equation 3.4) returns maximum value than considering the other neighbors' locations.

$$\gamma(l_u) = \prod_{e(u,v_j) \in E_F} \frac{p(|l_u - l_v|)}{1 - p(|l_u - l_v|)}, \qquad l_u \neq l_v \tag{3.4}$$

Computing γ for each location is itself an expensive operation. As suggested in [9], the value of γ likelihood of each location can be pre-computed and we compute γ using PreCompute() method in our generalized framework.

Remarks. In the original paper [9], the authors mention that the model performs better for the users with 16 or more located friends. Hence, we exclude inferring some users who have 'a few' (e.g. one or two) neighbors and it helps to improve the efficiency of this model.

3.4.5 Social Tie-strength based Model

The Tie-strength based models [19, 101, 127] investigate social relationships that have stronger social tie and incorporate them in predicting users' locations. *FriendlyLocation* [101] (abbreviated as *Friendly*) is the representative model of this category. It leverages the relationship between tie-strength of users pairs, and their mutual distances. The basic assumption of this model is that users with strong ties are more likely to live near each other. Several social factors

e.g., following relationships, number of friends, conversations between social users, etc. are considered to measure the user proximity.

The *Friendly* model is semi-supervised model where the aforementioned social factors are used to train a decision tree classifier to distinguish between users' pairs who are likely to live nearby, and those who are distant. This model divides the predicted distance returned by the regression tree into 'm' number of quantiles. Let $\{q_0, q_1, ..., q_m\}$ be the boundaries. Each predicted distance d_i^p of i^{th} edge (s.t. $e_i\langle u, v\rangle \in E$) is assigned with a quantile number:

$$qntl(d_i^p) = \max_{j \in \{0,..,m\}} \{j : d_i^p < q_j\}$$

The number of socially connected edges ($actEdges$) in each quantile with distance 'd' is measured as,

$$actEdges(k, d) = |f_i\langle u, v\rangle \in E_F : d = d_i^a \wedge k = qntl(d_i^p)|$$

Similarly, possible number of edges ($stgrEdges$) that could have existed at a distance d is calculated as,

$$stgrEdges(d) = |e\langle u, v\rangle : u \in V \wedge v \in V \wedge d = dist(l_u, l_v)|$$

Finally, the probability of a neighbor in a quantile j lives within d distance is measured as,

$$p^*(k, d) = \frac{actEdge(k, d)}{stgrEdges(d)}$$

Using training data, $p^*(k, d)$ function can be fit into curve for each quantile:

$$p^*(k, d) = a_k(b_k + d)^{-c_k}$$

Now, the likelihood of a location $l \in L(ngbr(u))$ is maximized and the best location is inferred to the user:

$$F(l, L) = \prod_{l(ngbr_i(u)), d_i^p \in D^p} \frac{p^*(qntl(d_i^p), |l, l(ngbr_i(u)|)}{(1 - p(|l, l(ngbr_i(u)|))}$$

3.4.6 Social Coefficient based Model

Social Coefficient based model is based on the hypothesis that social distance can identify the closest friends in location estimation. In this model category, Kong et al. [78] propose *SPOT* model that calculates the energy of a user $u_i \in U^N$ locating at location l and having the social closeness score s_{ij} with neighbors. The 'social closeness' is calculated using the cosine similarity between a pair of users u_i and u_j,

$$s_{ij} = |ngbr(u_i) \cap ngbr(u_j)| / \sqrt{|ngbr(u_i)||ngbr(u_j)|}$$

Given the information of the labeled users in network, the probability $p(d_{ij}, s_{ij})$ of users $u_i \in V^*$ and $u_j \in V^*$ located at distance (d_{ij}) with their social closeness s_{ij} is measured. The maximum likelihood of a neighbor location w.r.t. social closeness score is predicted as the user location.

SPOT [78] model improves the location estimation errors due to highly uneven neighbor distribution and location sparsity problem. In these scenarios, to enhance the performance of 'social closeness' based models, the energy and local social coefficient based approach is introduced to measure total energy of a user u_i locating at a location l_i:

$$Q(u_i, l_i) = -\sum_{j=1}^{|ngbr(u_i)|} s_{ij}.g(u_i, u_j)$$

Here, $g(u_i, u_j) = -e^{-|l_i,l_j|/d(s_{ij})}$, $u_j \in ngbr(u_i) \wedge U^*$ and $d(s_{ij})$ is the average distance of user u_i and neighbors u_j when the social similarity score is s_{ij}.

Local social coefficient of each user is calculated as,

$$C(u_i) = \frac{3Q_\triangle}{3Q_\triangle + Q_\wedge}$$

Here, $Q_\triangle$ is the number of closed triplet and $Q_\wedge$ is the number of open triplet connected by u_i with her neighbors. The energy value, $Q(u_i, l_i)$ and the social coefficient, $C(u_i)$ of each friend location are ranked to fit a logistic response function. The location with the highest probability is predicted as the user location.

3.4.7 Label Propagation based Model

The label propagation based location prediction approaches (e.g. *SLP* [74]) predict a user's location by propagating the location labels among their neighbors. It follows a multi-pass iterative process. *SLP* [74] model, a representative of this category assigns a location of a user with the geometric median of neighbors' locations. The inferred locations can be used further to predict the location of the adjacent users while making new inferences.

In *SLP*, a location among the neighbors is selected by analyzing the spatial arrangement of the neighbors' locations. The geometric median 'm':

$$m = \arg \min_{l \in L(ngbr(u))} \sum_{l_v \in L(ngbr(u)} |l, l_v|$$

is estimated as the location of the user $u \in U^N$ in the first pass. In each iteration, the newly predicted user location is further used to infer unlabeled neighbors' locations. This process continues until it satisfies convergence criteria. In *SLP*, the concept of the iteratively propagating the newly predicted location generates a flatter population distribution, which contradicts the concept that the majority of users live in dense area.

This model iterates multiple times until the stopping criterion are satisfied. In each iteration, the estimated locations are further used to predict the neighbors. In this way, some users with no labeled friends at the beginning may be predicted after a certain number of iterations. However, two problems may arise: (a) the incorrect estimation of a friend may lead to decreased accuracy when such predicted location is further used to infer the user. (b) Extensive iterations required for convergence may shift the correctly predicted location away, if large number of incorrect neighbor locations are included in each iteration.

3.4.8 Social Concentration based Model

The social concentration based model assumes that a higher proportion of a user's friends live in a dense location region. The representative *Landmark Mixture Model* (LMM) [159] model first identifies the set of *landmark* users who resides in a close proximity to others. After that, a location with the maximum likelihood among the landmark users' locations is inferred. Each landmark user is connected with a number of labeled neighbors and the centroid of the neighbors are used to calculate dominance distribution using Gaussian Mixture Model (GMM). The users with lower variance in dominance distribution and having sufficient neighbors (e.g. landmark user) are used to infer the unlabeled neighbors. The maximum likelihood location of landmark users (among labeled neighbors) is selected as the predicted location.

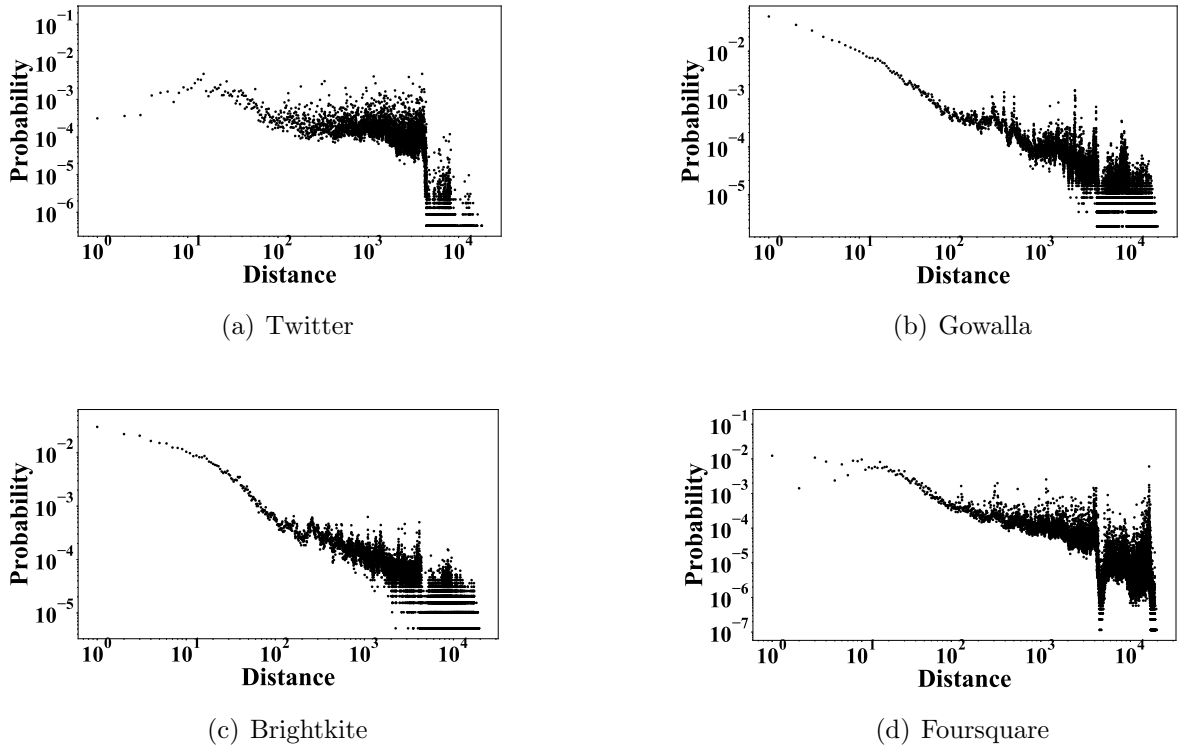
The landmark users must have a large number of immediate neighbors and the probability density of neighbor location distribution at the mode point should be high. However, the selection process of landmark users in *LMM* [159] is trivial. It lacks theoretical proof in identifying such users and the optimal parameter ranges are not mentioned. Moreover, this model may fail to predict a suitable location if the neighbors are distributed sparsely.

3.5 Benchmarking Evaluation

We perform extensive evaluation using our benchmark to measure the effectiveness, efficiency, memory consumption, prediction coverage, and combined performance of the eight representative models. The framework is implemented using Python in a Windows environment with Intel i7 CPU and 40 GB memory.

3.5.1 Datasets

We use four real world datasets including Twitter microblog, and Gowalla, Brightkite, Foursquare LBSN. All these four datasets have graph structure where each user is considered a vertex (i.e. node) and the relationship between two users (if exists) is represented with an edge. The spatial distribution of users' network can be used to infer individuals' locations. We have adapted the functionalities of the selected location prediction models in LBSN (i.e. check-in) datasets. It



(a) Twitter (b) Gowalla

(c) Brightkite (d) Foursquare

Figure 3.3: Probabilities of following as function of Distance

is noted that in real-world, many LBSN users might have social connections but no check-ins (or profile locations) [137]. Hence, it is useful to infer locations of such users from the network properties of LBSN. Table 3.2 gives a summary of the four datasets.

Twitter. The Twitter (TW) dataset [90] was originally collected in May 2011 and the users are distributed in different cities of USA. We select 138,012 active users from this dataset who have both network information as well as tweet contents. This dataset is location annotated (see [90] for details) and we transform the location name to geo-points using Google Geo-location API[1].

Gowalla. The Gowalla (GW) LBSN dataset is collected from SNAP repository[2] and contains 6,442,892 check-ins on 1,280,969 places worldwide over a period spanning from Feb. 2009 to Oct. 2010. In this dataset, 107,092 users have multiple check-in locations and form an explicit social network.

Brightkite. Brightkite (BK) is another publicly ava-ilable LBSN dataset in SNAP repository. The original data was collected over the period Apr. 2008 - Oct. 2010. This dataset has multiple check-in locations and a social graph is constructed using 50,686 users who have both check-ins and network information.

Foursquare. The Foursquare (FS) LBSN dataset was collected using public API [84]. A

[1]https://developers.google.com/maps/documentation/
[2]http://snap.stanford.edu

Table 3.2: Summary of the Datasets used

Dataset Name	# Users	# Edges	Average Degree	Average Neighbor Distance (km)	Average Node Locality
Twitter	138,012	2,274,416	32.95	1,402	0.82
Gowalla	107,092	456,830	8.53	1,722	0.52
Brightkite	51,406	197,167	7.67	1,819	0.69
Foursquare	2,127,093	8,640,352	8.12	2,629	0.79

Table 3.3: Summary of the Twitter Dataset

Dataset ID	# Unlabeled users	# Labeled users	# Average Labeled Neighbors
Dataset TW-I	27,602	110,410	28.86
Dataset TW-II	55,205	82,807	23.20
Dataset TW-III	82,807	55,205	15.44
Dataset TW-IV	110,410	27,602	7.68
Dataset TW-V	124,211	13,801	5.72

total of 2.12 million users have self-reported location profile. We create a social graph, G using the users who have self-reported locations and social connections.

3.5.2 Ground-truth Information of Datasets

Self-reported profile locations are used as the ground-truth in Twitter [90] and Foursquare [84]. Since no profile locations are explicitly available in Brightkite and Gowalla datasets, we select the ground-truth location using approaches similar to Cho et al. [26]. We discretize the spherical earth surface into 0.2 degree by 0.2 degree cells, which is approximately equal to 22 by 22km w.r.t. equatorial region. For a given user, we find the cell with the 'most number of check-ins' [136] and within this cell, we select the average check-in position as the ground truth for Gowalla and Brightkite datasets.

3.5.3 Friendship, Distance and Check-in Characteristics

In Figure 3.3, we plot the following probability with distance between a pair of users u_i and u_j, s.t. $e(u_i, u_j) \in E$. The figure shows that, (1) the *following* probability decreases when the distance between the user pairs increases, (2) in Twitter, the distribution is much flatter than the other three datasets. Gowalla and Brightkite have similar distributions. All the patterns successfully capture the fact that a user is likely to follow others who live close. The following probabilities in each dataset can be fitted into a power law distribution curve if we ignore the larger distance pairs in each dataset. The user check-in activities also follow a heavy-tailed distribution [137] and the majority of check-in venues are near to users' stable locations [26].

Table 3.4: Summary of the Gowalla Dataset

Dataset ID	# Unlabeled users	# Labeled users	# Average Labeled Neighbors
Dataset GW-I	21,418	85,674	11.08
Dataset GW-II	42,830	64,262	8.85
Dataset GW-III	64,262	42,830	7.09
Dataset GW-IV	85,674	21,418	5.43
Dataset GW-V	96,383	10,709	4.15

Table 3.5: Summary of the Brightkite Dataset

Dataset ID	# Unlabeled users	# Labeled users	# Average Labeled Neighbors
Dataset BK-I	10,137	40,549	6.54
Dataset BK-II	20,274	30,412	4.93
Dataset BK-III	30,412	20,274	3.43
Dataset BK-IV	40,549	10,137	2.02
Dataset BK-V	45,617	5,069	0.80

Table 3.6: Summary of the Foursquare Dataset

Dataset ID	# Unlabeled users	# Labeled users	# Average Labeled Neighbors
Dataset FS-I	425,418	1,701,674	7.73
Dataset FS-II	850,837	1,276,2557	5.76
Dataset FS-III	1,276,255	850,837	3.83
Dataset FS-IV	1,701,674	425,418	2.08
Dataset FS-V	1,914,382	212,710	0.61

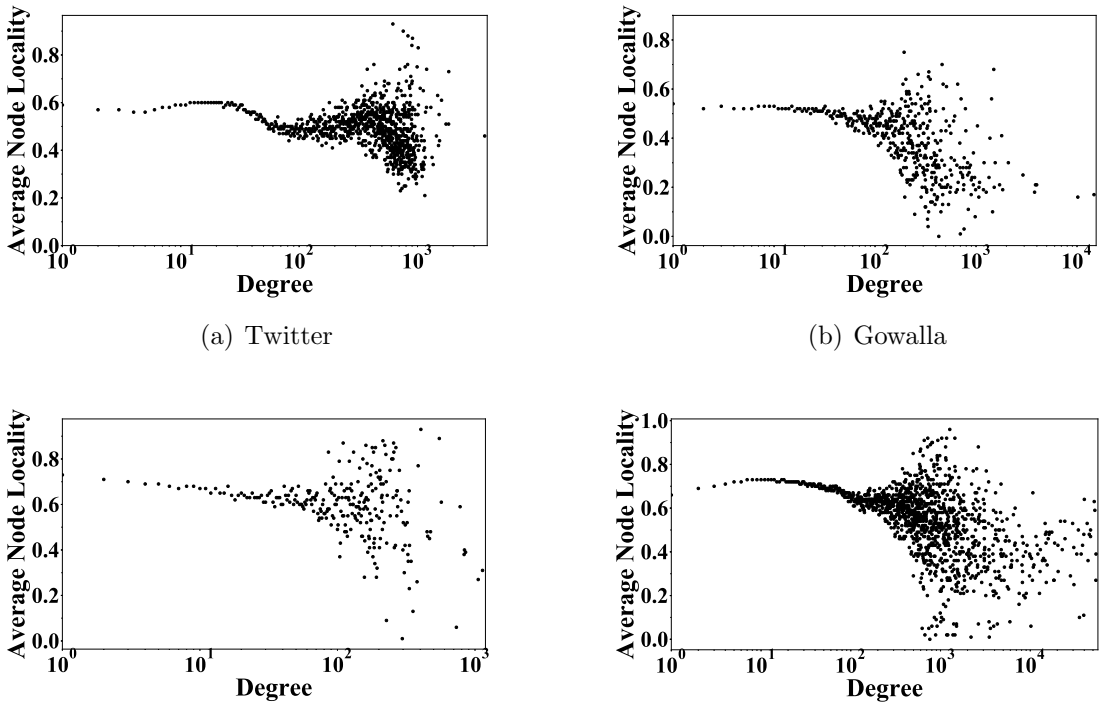
3.5.4 Node Locality

Node Locality is useful to quantify the geographic closeness of the neighbors to a certain user. For a user u_i with $|ngbr(u_i)|$ 1-hop neighbors, the *node locality* is calculated as [134]:

$$NL(u_i) = \frac{1}{|ngbr(u_i)|} \times \sum_{v_j \in ngbr(u_i)} e^{-\frac{d(u_i,v_j)}{\beta}}$$

where β is a scaling factor and it is calculated as follow:

$$\beta = \frac{1}{|E|} \times \sum_{u,v \in V, e(u,v) \in E} d(u,v)$$



(a) Twitter

(b) Gowalla

(c) Brightkite

(d) Foursquare

Figure 3.4: Average Node Locality as a function of Node Degree

Table 3.2 provides the average neighbor distance and the average node locality of each dataset. A dataset with higher average node locality should have large number of social connections within a close geographic region [134]. The users in Twitter dataset are distributed within USA and this dataset has a higher average node locality score (0.82). Meanwhile, Gowalla has a lower node locality score (0.52) among the four datasets. This provides evidence that users in Gowalla are engaged with a geographically spread set of individuals rather than only with users at closer distances.

The correlation between node degree and node locality is useful to understand the socio-

Table 3.7: Parameter Settings in different models

Model	Parameters and Value
UDI [90]	outer iteration = 3, convergence error = 0.1
MLP [89]	Location-based following probability distribution parameters, α = -0.55, β = 0.0045 (values of α, β depend on data type)
Back [9]	Friendship distance coefficient: $a = 0.0019$, $b = 0.196$, $c = -1.05$ (values of a, b, c depend on data type)
SLP [74]	no of iterations = 4
TFIDF [82]	*tfidf* threshold = 0.1
Friendly [101]	LCR min dist = 40 km, quantile number = 10, max sample leaf = 1000
SPOT [78]	no of iterations = 4

Table 3.8: Metrics used in different models

Metrics	Model / Work Reference
AED@d, MeanED	UDI [90], Friendly [101], LMM [159], SLP [74], Cheng et al. [24]
AED@k%	UDI [90]
MedianEd	Friendly [101], LMM [159]
Acc, Acc@d	UDI [90], MLP [89], Friendly [101], Rout et al. [127], Cheng et al. [24]
Acc@K	Cheng et al. [24], [25], MLP [89], Bo et al. [64]
Precision, DP@K	OLIM [160], MLP [89], Davis Jr. et al. [33]
Recall, DR@K	OLIM [160], MLP [89], Davis Jr. et al. [33], Compton et al. [29], LMM [159]
CDF	Back [9], SLP [74]

spatial properties of users. Figure 3.4 shows the average node locality as a function of node degree. It shows a fairly constant trend in each dataset when the average node degree is less than 100. These set of users may have similar properties with a similar proportion of neighbors living distant. In Gowalla, the average node locality drops significantly with the increase of node degree.

3.5.5 Parameter Settings

In our evaluation, the essential parameters of the models are configured as recommended in the original papers. Table 3.7 shows the parameter settings of the models. One of the important parameters is the number of times a model is allowed to iterate. We set the default value as 'two' for those models that follow multiple iterations but do not clearly mention in the original papers (e.g. *Backstrom* [9]). Meanwhile, the friendship-distance coefficient [9] and location-based following probability parameters (α, β) [89] are sensitive to the types of data. *MLP* [89] model reports the value of $\alpha = -0.55$ and $\beta = 0.0045$ in Twitter, however, it is calculated as $\alpha = -0.42$ and $\beta = 0.0030$ in our selected Twitter social graph. Similarly, in our experiment, the values of (α, β) in Gowalla, Brightkite, and Foursquare are measured as (-1.52, 0.612), (-1.14, 0.20), and (-0.65, 0.016) respectively.

3.5.6 Metrics for Evaluation

Table 3.8 lists the metrics used in the existing location prediction models. In this section, we discuss the suite of metrics used in our study to compare the models in a transparent comparison frame. We use Haversine [141] formula to measure the distance between two geo-points in kilometer unit. Haversine strikes a good balance between correctness and computational efficiency that works over spherical earth surface.

Metric I. *Average Error Distance* (AED) calculates the average distance between the actual location (l_{u_i}) and the predicted location ($\hat{l}_{u_i}$) of users,

$$AED(\hat{V}^N, M_x) = \frac{\sum_{u_i \in \hat{V}^N} Err(u_i, M_x)}{|\hat{V}^N|}$$

where, $Err(u_i, M_x) = d(l_{u_i}, \hat{l}_{u_i})$ is the *Error Distance* (ED) between the actual and the predicted location of a user u_i in model M_x.

We evaluate the models using both 'distance' and 'percentage' based *AED*. The distance-based *AED@d* metric measures the average of the error distances of those users whose locations are predicted within 'd' km:

$$AED@d = \frac{1}{|u_i|} \sum_{u_i \in \hat{V}^N} Err(u_i | u_i \in \hat{V}^N \wedge Err(u_i, M_x) \leq d)$$

The percentage based *AED@k%* calculates the average of the error distance of top '$k\%$' predicted users.

Metric II. *Precision* (Prec) measures the quality of a prediction model. This metric calculates the percentage of the users predicted with error distance less than *'d'* kilometers. In a set of predicted users $\hat{V}^N$, the *precision* of a model M_x is calculated as:

$$Prec@d = \frac{|\{u_i | u_i \in \hat{V}^N \wedge Err(u_i, M_x) \leq d\}|}{|\hat{V}^N|}$$

Metric III. *Accuracy* (Acc@d) (or *Recall* [89, 159]) measures the proportion of the correctly predicted users (with error distance less than *'d'*) among the test users V^N in a location prediction model M_x,

$$Acc@d = \frac{|\{u_i | u_i \in V^N \wedge Err(u_i, M_x) \leq d\}|}{|V^N|}$$

Metric IV. *Prediction coverage* measures the percentage of the unlabeled users (V^N) who have been assigned a location by a model regardless of the prediction accuracy. Let, a model M_x predicts $\hat{V}^N$ users among V^N unlabeled users in a dataset. The coverage of the model is calculated as ($\frac{|\hat{V}^N|}{|V^N|} \times 100$).

Metrics V. *Mutual Prediction Ratio* (MPR) measures the percentage of similar predictions of two different models M_x and M_y within a given error distance d (e.g., 20 km):

$$MPR(M_x, M_y) = \frac{|\bigcap_{M_x,M_y}\{u_i|u_i \in \hat{V}^N \wedge Err(u_i) \leq d\}|}{|\bigcup_{M_x,M_y}\{u_i|u_i \in \hat{V}^N \wedge Err(u_i) \leq d\}|}$$

This metric measures the mutual agreement of two models on location prediction.

3.5.7 Performance Evaluation Configuration

Evaluation on Data Types and Location Sparsity

To evaluate the models using various 'data-centric' configurations, we perform experiments on the Twitter microblog and three LBSN datasets (e.g. Gowalla, Bright-kite, and Foursquare). Initially, these datasets are location annotated. We investigate the effect of the location sparseness, and randomly choose 20%, 40%, 60%, 80%, and 90% users from each dataset to mask their locations (labeled as I, II, III, IV, V). Data setting I is the 'less' sparse with 20% unlabeled and 80% labeled users whereas data setting IV is the 'highly' sparse containing 80% unlabeled and 20% labeled users. Data setting V has 'extreme' sparsity with 90% unlabeled users. The location masked users are termed as 'unlabeled' (V^N) and not used in the location estimation process. The statistics of the datasets with five sparsity levels are given in Table 3.3 - Table 3.6.

Evaluation on Different Types of Users

We design some experimental settings using Node degree and Node locality to analyze the effects of 'user-centric' properties in the network.

Different Node degree. Different number of neighbors may affect the accuracy of the location prediction models. We divide the users into four groups w.r.t. node degree as: "5-10", "10-20", "20-30", and "> 30".

Different Node locality. Moreover, to explore the effect of neighbours' geographical distances in the prediction accuracy, we also group the users as, "0.0-0.2", "0.2-0.4","0.4-0.6","0.6-0.8","0.8-1.0" on node locality.

Region-specific Model Performance

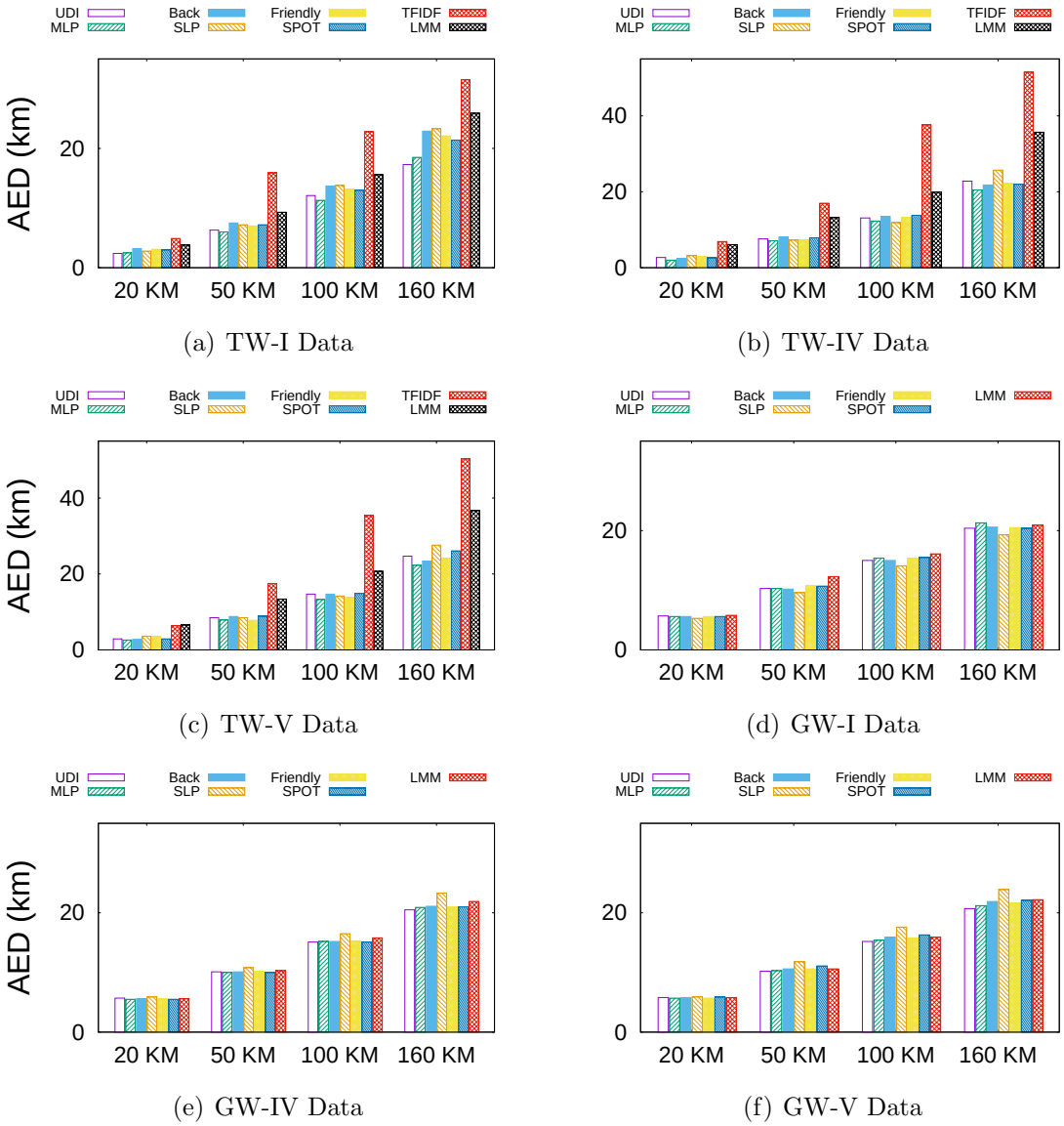
Different social media captures different kind of users distributed in various spatial regions. The network properties of social media in particular regions may have distinct set of characteristics. Hence, for some model, it may be much easier to predict a large number of users in a specific spatial region than the other models.

Scalability Evaluation

Scalability of the prediction models is an important dimension for practical point of view in various applications (e.g. emergency reporting system). We compare the time cost and average memory consumption of the models in different data settings.

3.5.8 Effectiveness on different types of social media datasets with different parameter settings

We analyze the eight models on the metrics defined in Section 3.5.6. Note that, *TFIDF* model can only be tested on Twitter data as the three LBSN datasets do not have content information.

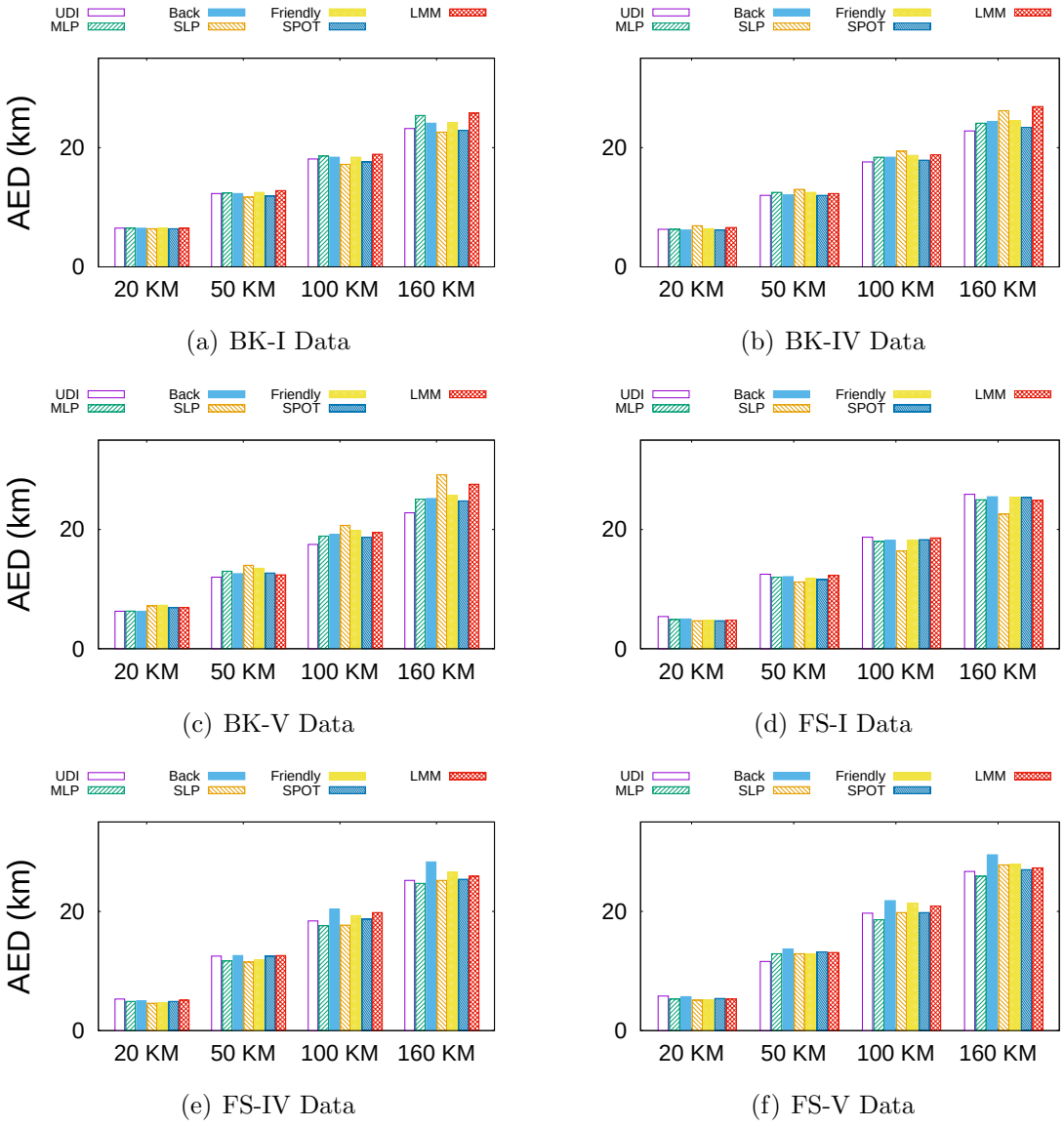


(a) TW-I Data (b) TW-IV Data

(c) TW-V Data (d) GW-I Data

(e) GW-IV Data (f) GW-V Data

Figure 3.5: *AED@d* in Twitter and Gowalla using Different Data settings



(a) BK-I Data

(b) BK-IV Data

(c) BK-V Data

(d) FS-I Data

(e) FS-IV Data

(f) FS-V Data

Figure 3.6: *AED@d* in Brightkite and Foursquare using Different Data settings

AED@d.

In Figure 3.5 and Figure 3.6, we report *AED@d* within 20km, 50km, 100km, and 160km of error distances using data settings I (less sparse), IV (highly sparse) and V (extreme sparse). The models in the Twitter dataset have lower *AED* value than three LBSN data. For example, in Twitter with data sparsity I and IV, the *UDI* model has 2.4km and 2.7km *AED@20* respectively. However, it is observed 5km higher in the LBSN datasets with similar data settings. The *AED* in extreme sparsity (i.e. level V) level is little higher than sparsity level IV. We found *AED@d* is always higher in *TFIDF* model. Below, we discuss models' relative performance using *AED@d*.

SLP Model. The *SLP* model has the lowest *AED@d* values in three LBSN datasets with 'less' sparse data setting (e.g. 20% unlabeled), and it generates *AED@160* as 19.3km, 22.6km, and 22.6km in Gowalla, Brightkite, and Foursquare respectively. However, the *AED@160* increases by 3.0, 3.6, and 2.6 km in these three datasets respectively when data sparsity level changes from I to IV (e.g. from 'less' to 'highly' sparse). In Twitter, the value of *AED@160* increases by 2.4 km only.

Backstrom, SPOT and Friendly Models. The relative *AED@d* of *Backstrom*, *SPOT*, and *Friendly* remains similar in each data types. However, in Foursquare with high data sparsity (e.g. FS-IV), the *AED@160* of *Backstrom* has higher value than the other two models.

LMM Model. The *LMM* model has higher *AED@d* in majority of the data settings in LBSN. A significant increase in *AED@160* is noticed while location sparsity changes in Twitter dataset. However, the *AED* of *LMM* does not change much in LBSN. For example, in Twitter, the *AED@160* is 10km and in Brightkite 1.1km higher when sparsity level changes from I to IV.

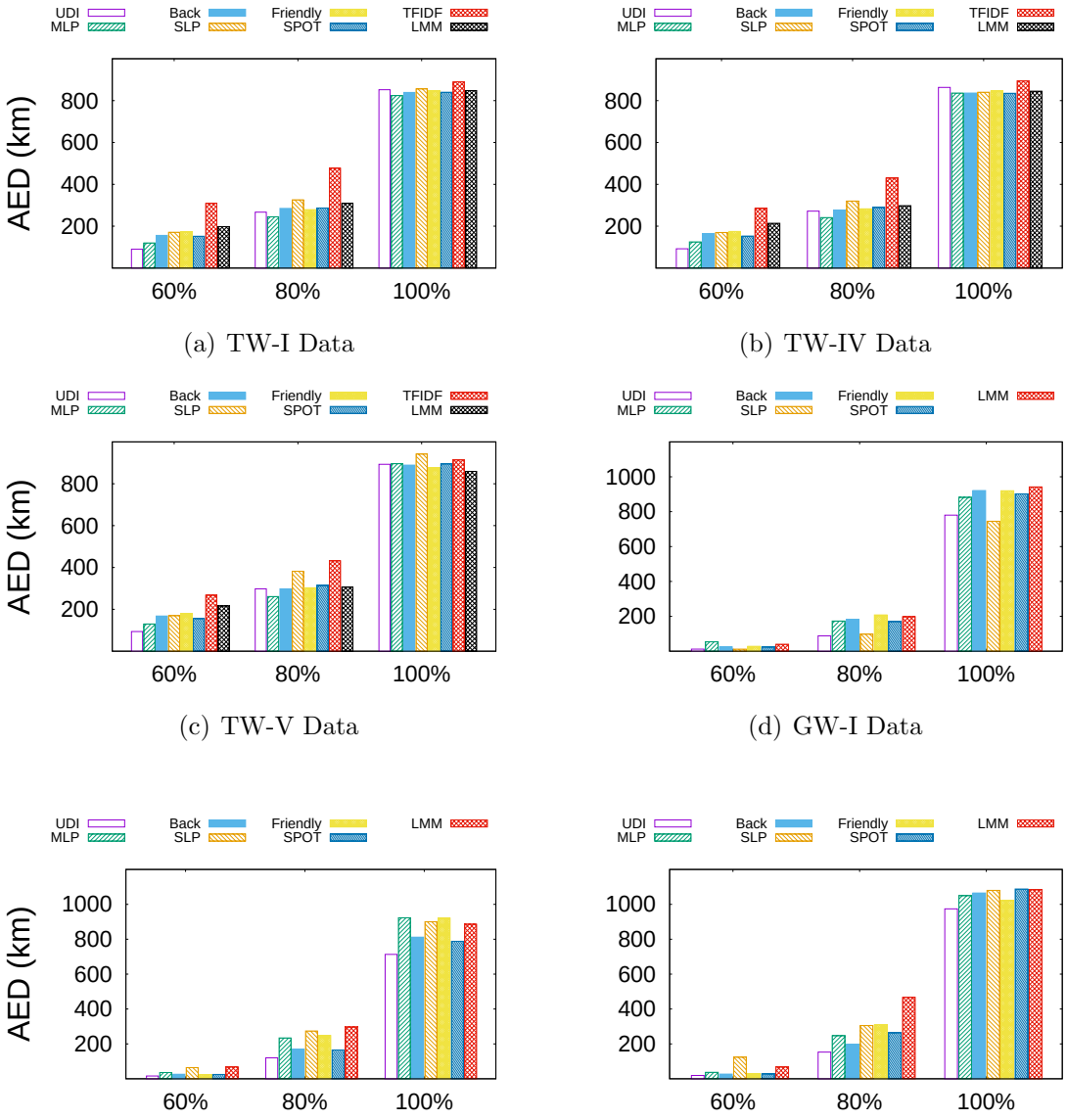
UDI and MLP Model. In comparison with Twitter, the *UDI* and *MLP* models have always higher *AED* in LBSN datasets. This is because, the content information available in Twitter help to predict more precise locations than three LBSN datasets.

AED@k%.

A distance based *AED*@*d* can be easily affected by the outliers in the results. In addition, different amount of predictions within a certain error distance may not make a transparent comparison of the models. Hence, we use percentile based *AED*s that calculate average error distance using top k% (i.e. 60%, 80%, and 100%) of the predicted users ranked by their error distances. Figure 3.7 and Figure 3.8 show the *AED*@k% of the models.

UDI and MLP models. The *UDI* model has lowest *AED@k* in Gowalla and Brightkite. In Twitter, *MLP* and *UDI* have similar *AED@k* in both I and IV settings. However, we have noticed that *MLP* model has predicted more precise locations in FS-IV setting which generates a better in *AED*@100%. Meanwhile, the number of predicted users are lower in *MLP*. We have discussed the effect of prediction coverage in Section 3.5.11.

Backstrom, Friendly and LMM Models. The relative *AED@k%* of *Backstrom* and



(a) TW-I Data

(b) TW-IV Data

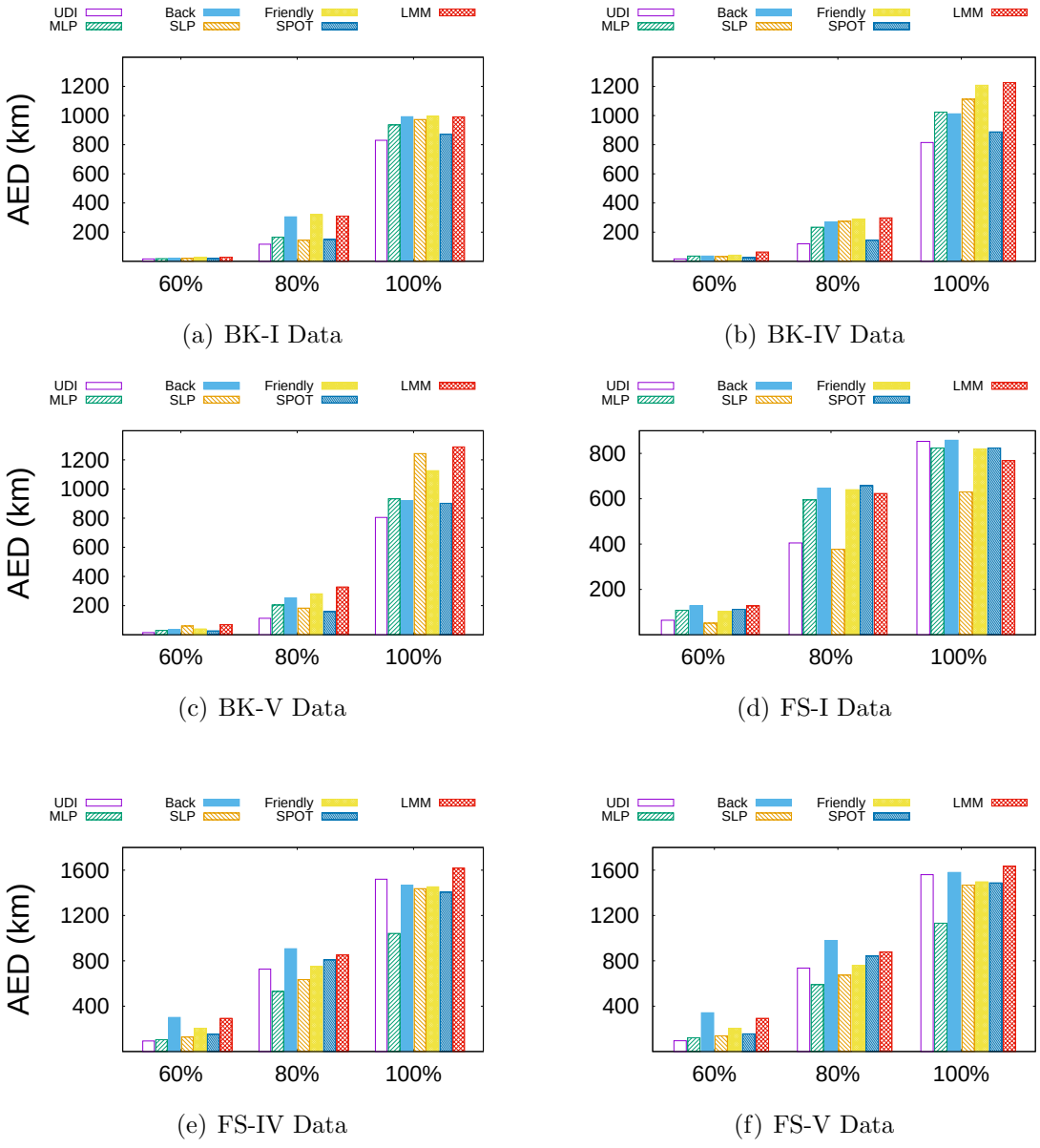
(c) TW-V Data

(d) GW-I Data

(e) GW-IV Data

(f) GW-V Data

Figure 3.7: *AED@k%* in Twitter and Gowalla using Different Data settings



(a) BK-I Data

(b) BK-IV Data

(c) BK-V Data

(d) FS-I Data

(e) FS-IV Data

(f) FS-V Data

Figure 3.8: *AED@k%* in Brightkite and Foursquare using Different Data settings

Friendly models are similar in each dataset. The prediction approaches of these two models maximize the probability of locations based on the curve fit using the edge probability with distance. However, the edge probabilities are different in these two models, but in similar setting these two model can predict users with similar error distances. The *LMM* model has relatively highest $AED@k$ in each of the data settings.

Precision.

Figure 3.9 - 3.12 show the precision of each model in four datasets with five sparsity levels. We only discuss the significant observations of the models' performance w.r.t. precision at 160 km (i.e. *Prec@160*).

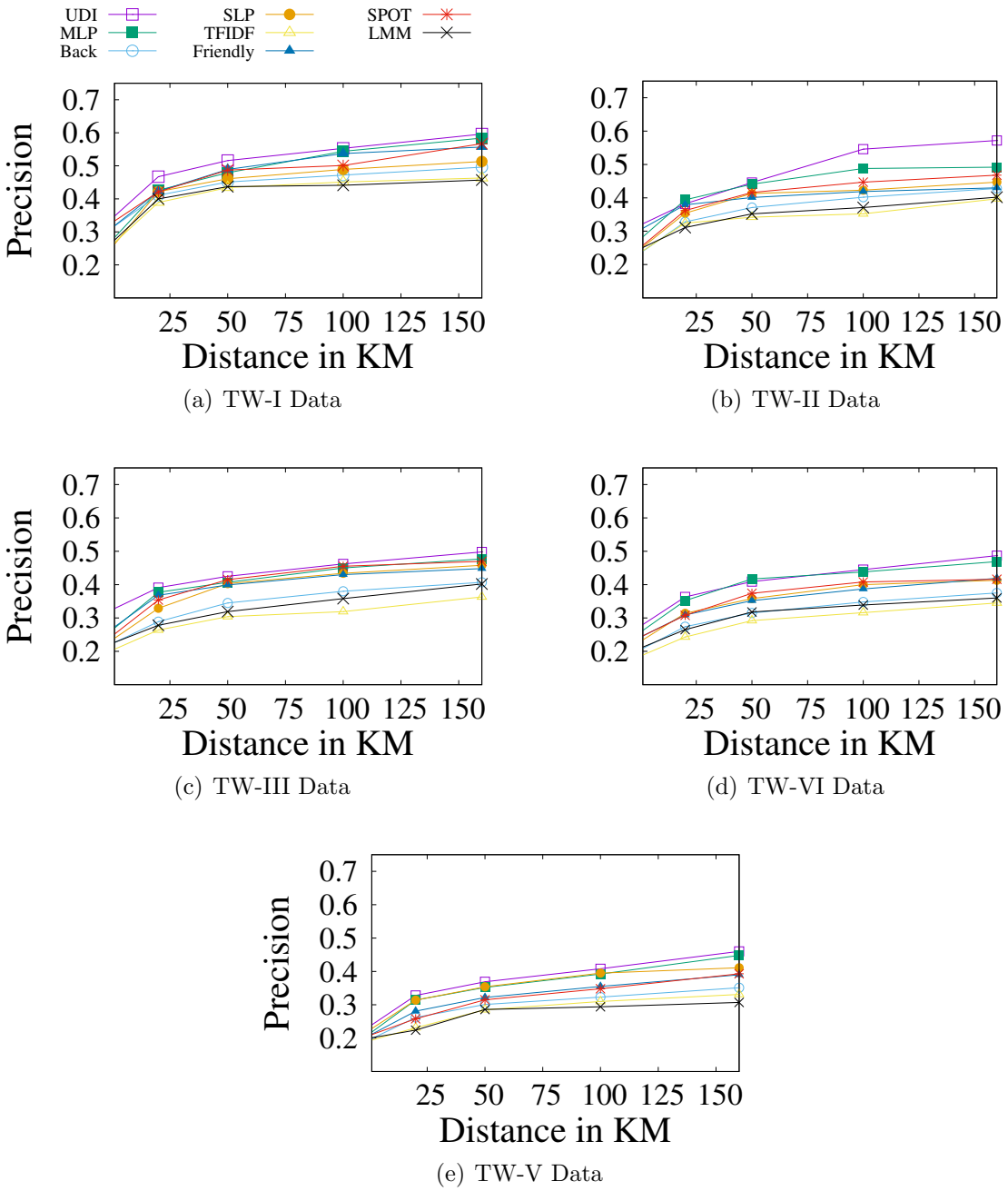
UDI Model. The *UDI* model has higher precision in majority of the datasets in different sparsity levels. In three LBSN datasets, the precision in *UDI* does not change much when location sparsity changes. For example, in Gowalla, the precision of *UDI* model drops only 3.12%, while no significant changes are observed in Brightkite and Foursquare datasets. However, in Twitter, the precision changes notably w.r.t. location sparsity. For example, it drops 11% when sparsity level increases from I from IV. Similarly, in extreme sparseness (e.g. TW-V), the precision drops 14% in Twitter.

SLP Model. In Gowalla and Brightkite datasets, the *SLP* model has second best *Prec@160* in sparsity levels I and II. Similarly, in Foursquare dataset, *SLP* has the highest precision on FS-I (68.33%) and FS-II (67.87%) data settings. However, we notice a significant decrease in precision of *SLP* when the location sparsity increases to higher and extreme higher levels.

Backstrom, SPOT, Friendly Models. The relative precision of these three models in Gowalla and Brightkite datasets are similar, where *SPOT* has higher *Prec@160* than *Friendly* and lower than *Backstrom*. However, in Twitter dataset, *SPOT* has better *Prec@160* than other two models. This may be because, the local social coefficient factor is effective in Twitter dataset and the average neighbor distance is lower in Twitter.

MLP and SLP Models. The *MLP* model has better precision than *SLP* in Twitter dataset. However, *Prec@160* in *SLP* is higher than *MLP* model in the first three data settings (i.e. I, II, and III) in LBSN datasets. Meanwhile, *MLP* has a better precision in sparser data settings. For example, in Gowalla GW-I setting, the value of *Prec@160* is 12% higher in *SLP*, however, *MLP* generates 5% and 12% better *Prec@160* in GW-IV and GW-V respectively.

LMM Model. The precision in *LMM* decreases drastically in three LBSN datasets with the increase in data sparsity. For example, *Prec@160* of *LMM* changes in Gowalla from 54% to 35% with sparsity level changes from I to IV. This is because, *LMM* do not iterate multiple times to precise the predicted users' locations.



(a) TW-I Data

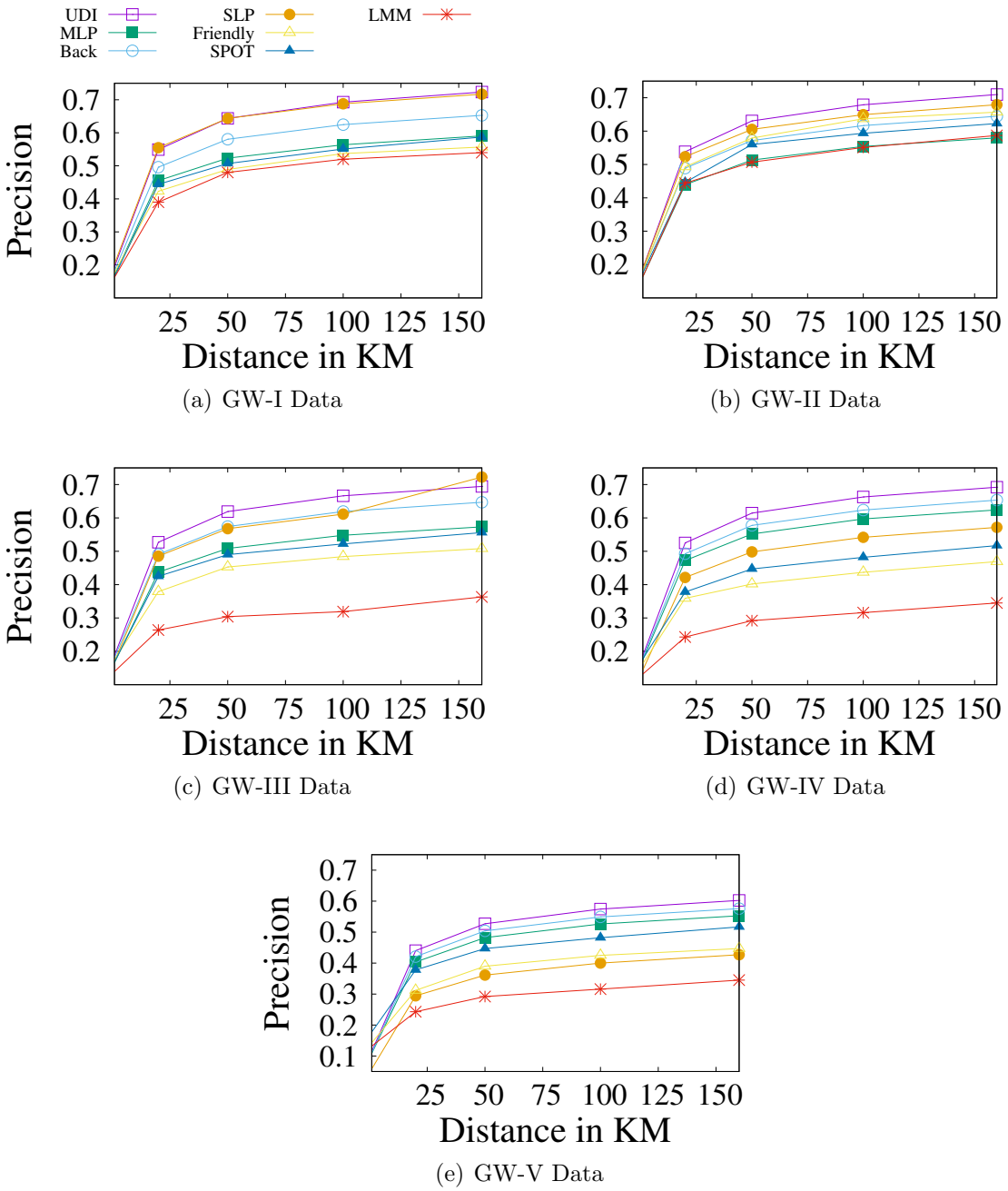
(b) TW-II Data

(c) TW-III Data

(d) TW-VI Data

(e) TW-V Data

Figure 3.9: Precision of the Location Prediction Models using Twitter Dataset



(a) GW-I Data

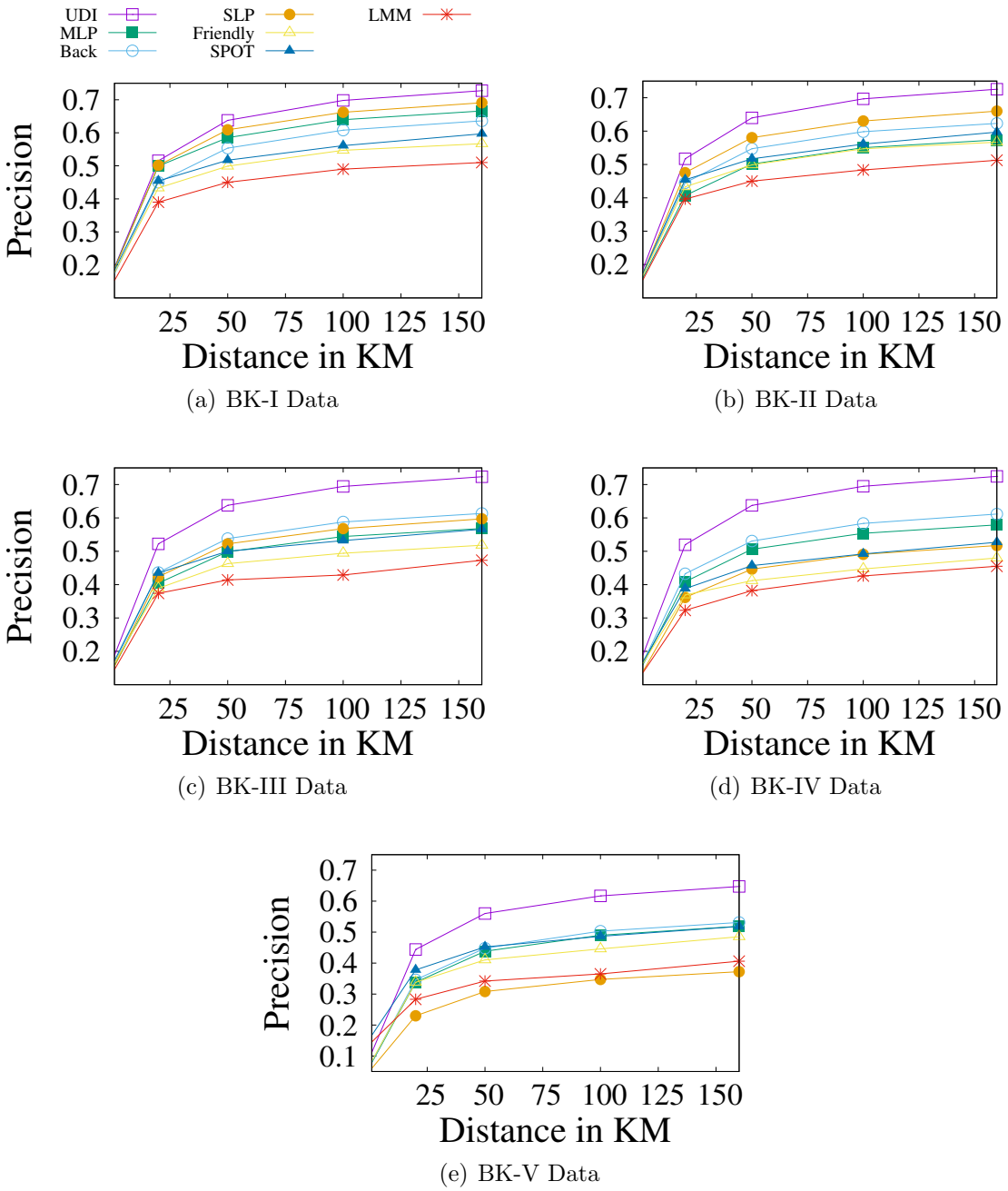
(b) GW-II Data

(c) GW-III Data

(d) GW-IV Data

(e) GW-V Data

Figure 3.10: Precision of the Location Prediction Models using Gowalla Dataset



(a) BK-I Data

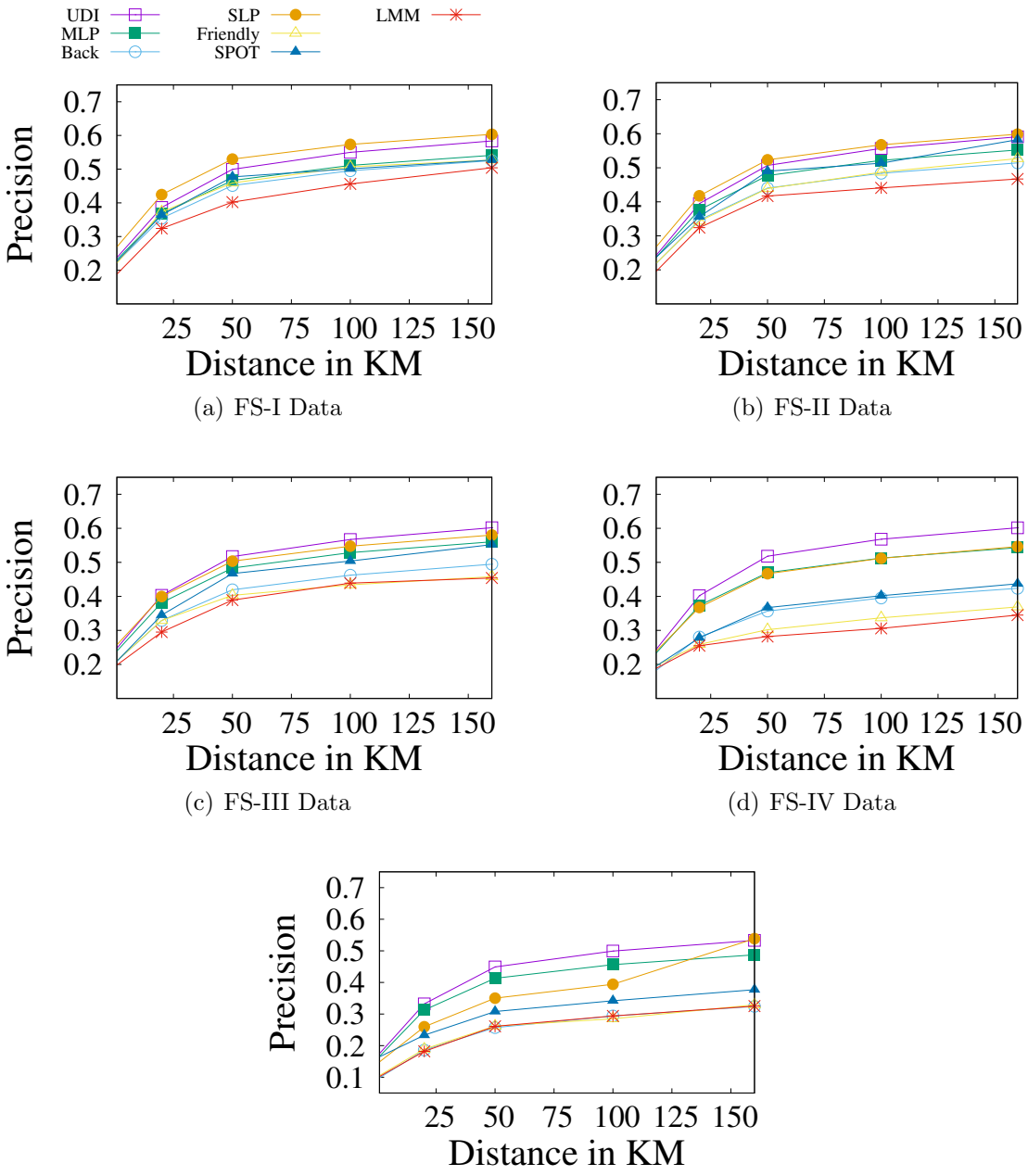
(b) BK-II Data

(c) BK-III Data

(d) BK-IV Data

(e) BK-V Data

Figure 3.11: Precision of the Location Prediction Models using BrightKite Dataset



(a) FS-I Data

(b) FS-II Data

(c) FS-III Data

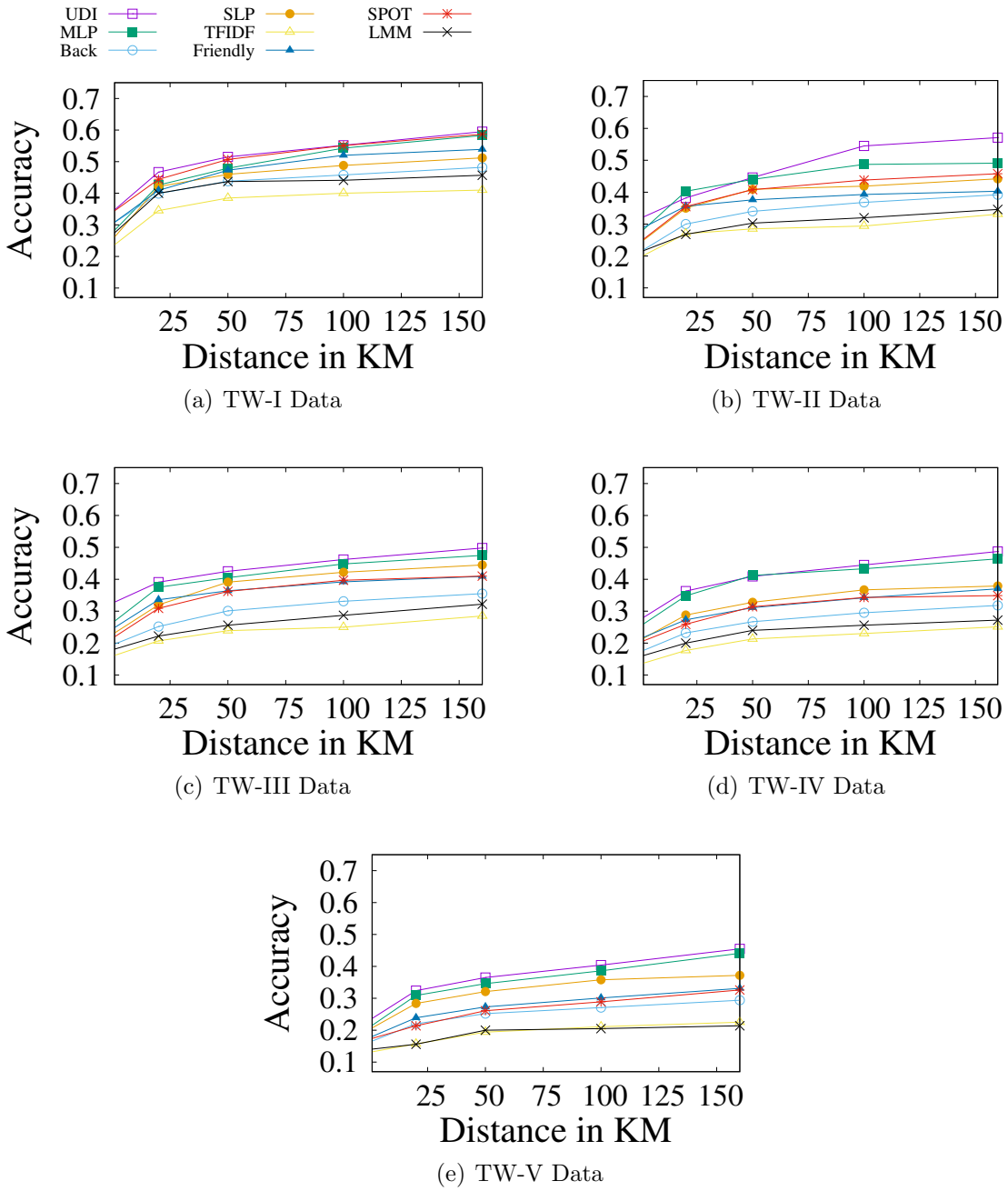
(d) FS-IV Data

(e) FS-V Data

Figure 3.12: Precision of the Location Prediction Models using Foursquare Dataset

Accuracy.

Figure 3.13 - 3.16 show the accuracy of the models in the four datasets under five location sparsity settings and different error distances ranging from 20km to 160km. We discuss accuracy of the models with 160km (i.e. *Acc@160*) error distance in the following discussions.



(a) TW-I Data

(b) TW-II Data

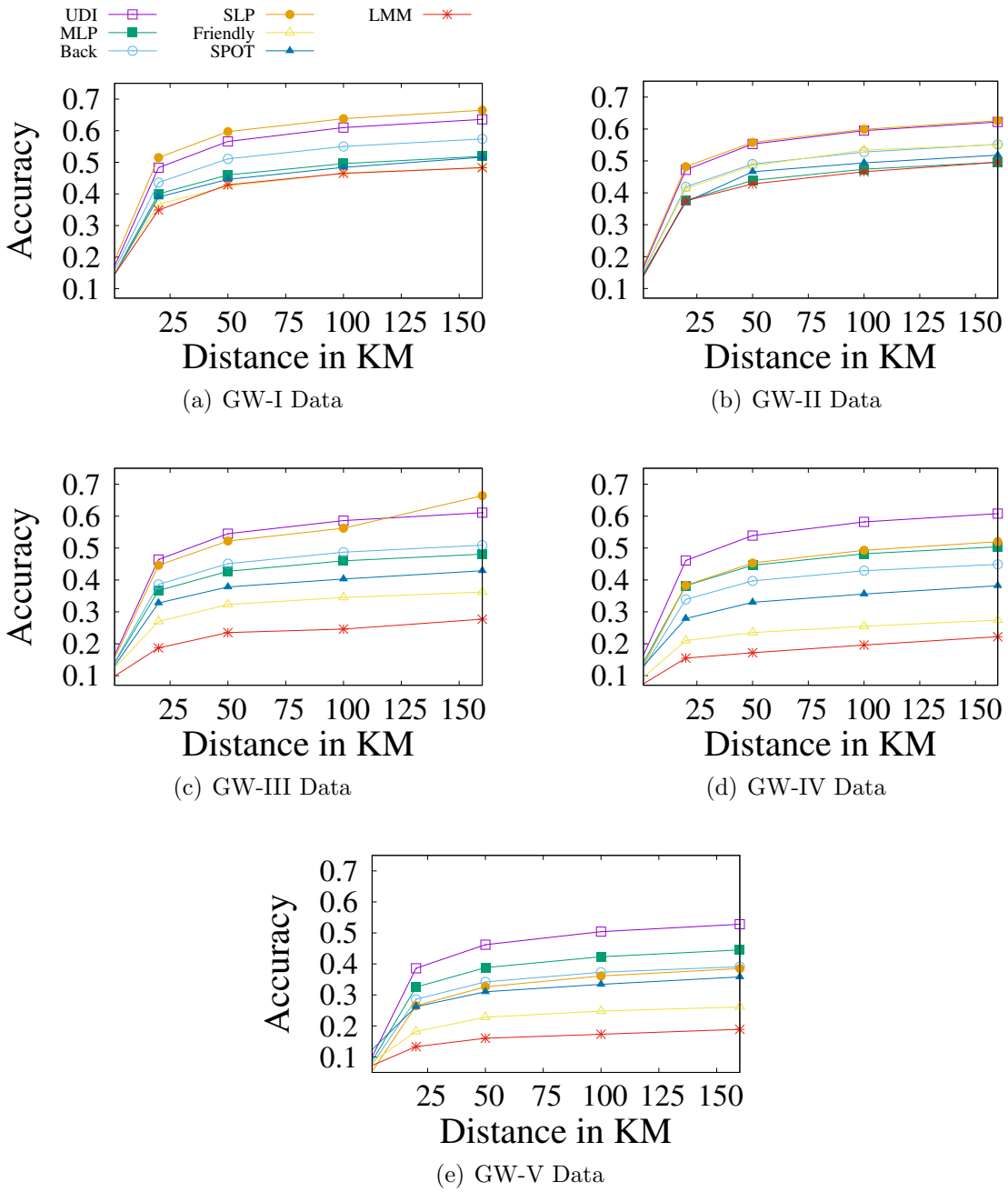
(c) TW-III Data

(d) TW-IV Data

(e) TW-V Data

Figure 3.13: Accuracy of the Location Prediction Models using Twitter Dataset

UDI Model. *UDI* has highest accuracy in Twitter dataset, but it is lower than *SLP* model in less sparse Gowalla and Foursquare data. However, with 'high' and 'extreme' sparsity level in these two datasets, the *UDI* model outperforms the second best model by 2% - 12% in *Acc@160*. This is due to multiple inner iterations perform by *UDI* model, where a better location is



(a) GW-I Data

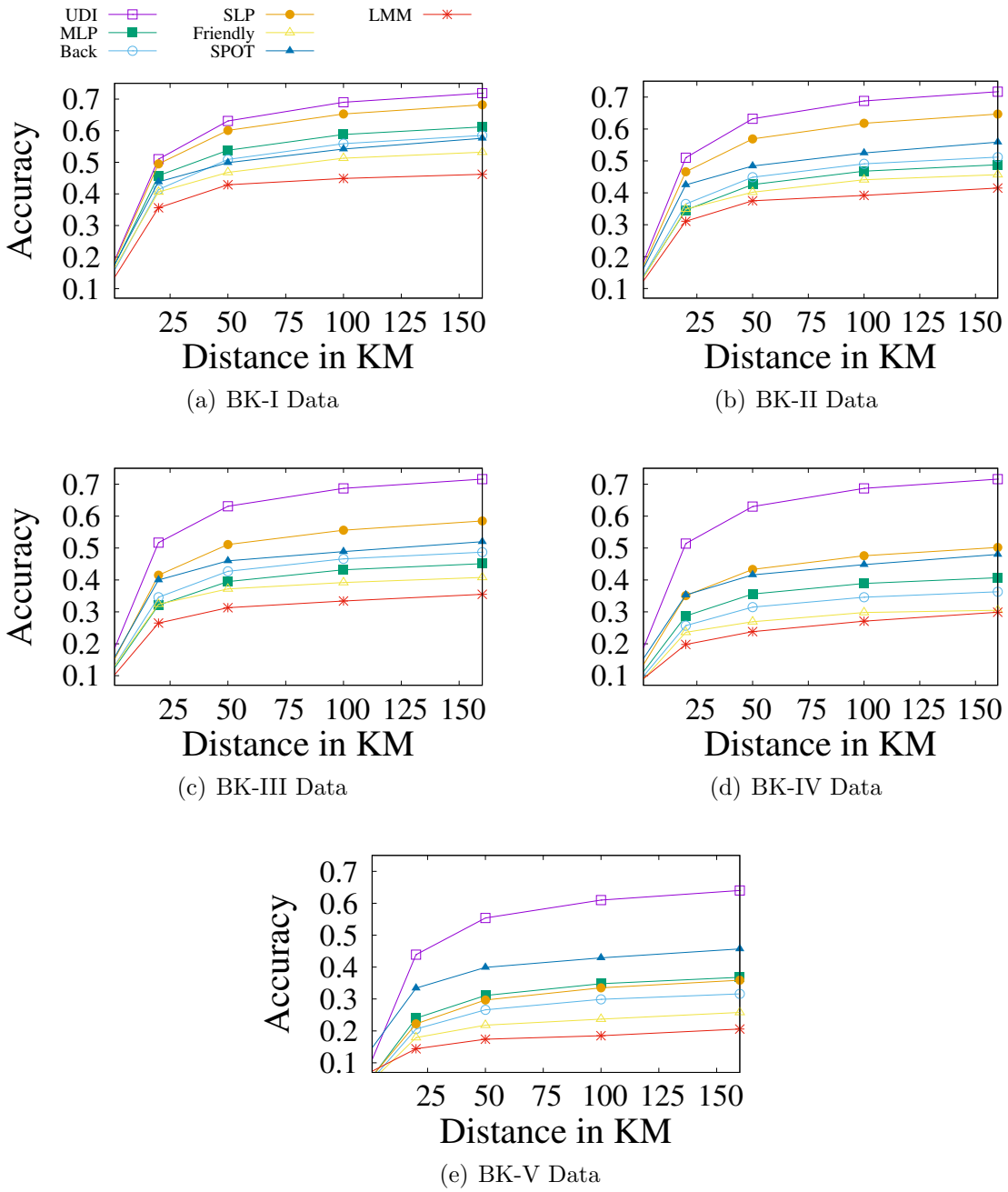
(b) GW-II Data

(c) GW-III Data

(d) GW-IV Data

(e) GW-V Data

Figure 3.14: Accuracy of the Location Prediction Models using Gowalla Dataset



(a) BK-I Data

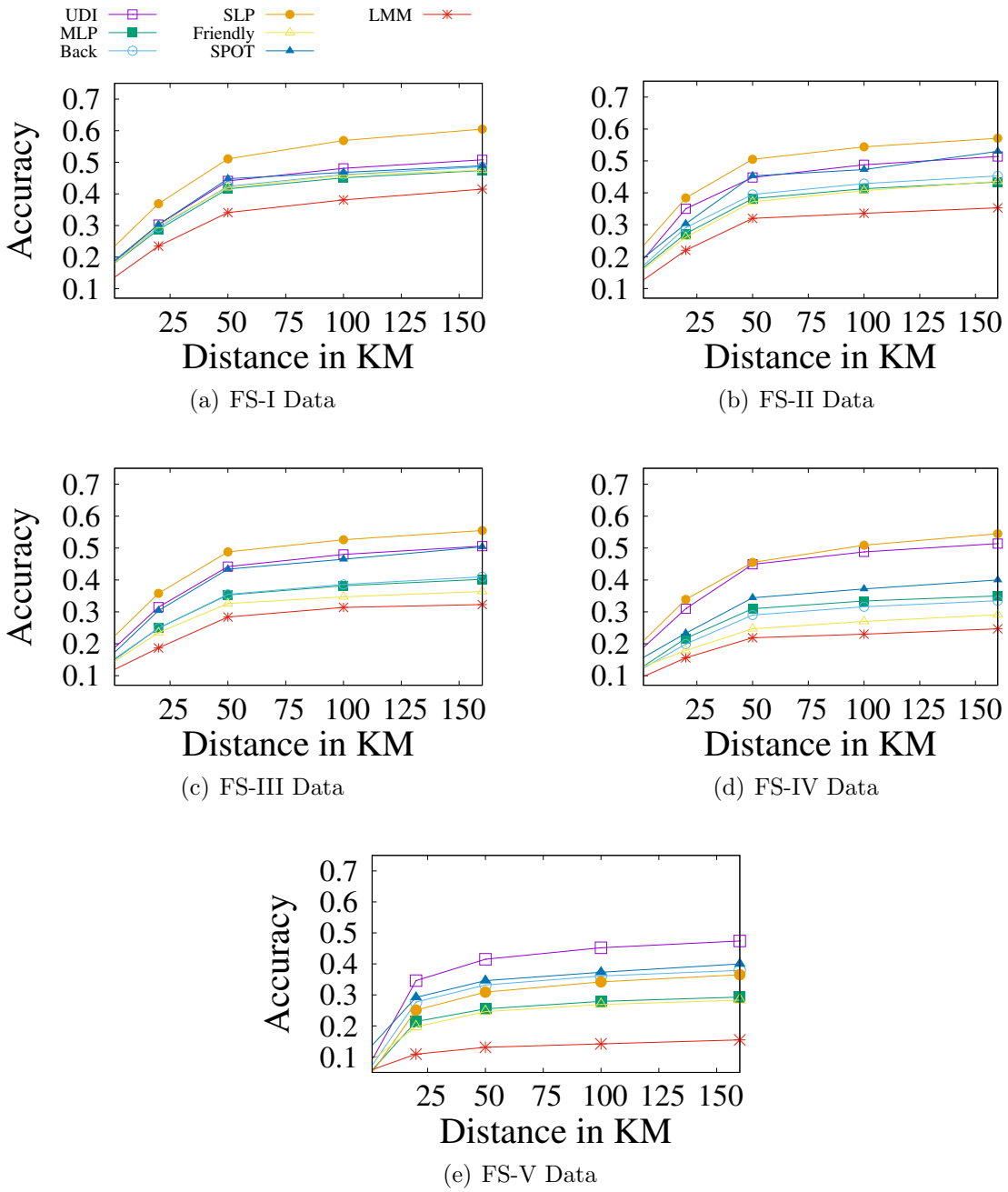
(b) BK-II Data

(c) BK-III Data

(d) BK-IV Data

(e) BK-V Data

Figure 3.15: Accuracy of the Location Prediction Models using BrightKite Dataset



(a) FS-I Data

(b) FS-II Data

(c) FS-III Data

(d) FS-IV Data

(e) FS-V Data

Figure 3.16: Accuracy of the Location Prediction Models using Foursquare Dataset

assigned until it converges. Hence, in sparse datasets, the *UDI* model can predict more users with precise locations. In Brightkite, the accuracy of *UDI* is highest in each of the five different data setting.

SLP Model. *SLP* has better accuracy in less sparse data settings and it decreases heavily when location sparsity increases. The performance of *SLP* model is always better when a large number of users have sufficient neighbor information. It reports highest *Acc@160* in Gowalla and Foursquare with data setting I and outperforms the second best by 7% and 3% respectively. However, in Twitter TW-I the *Acc@160* is 51% only.

MLP Model. The accuracy of *MLP* model decreases smoothly when the number of unlabeled user increases. In Twitter dataset, the *Acc@160* drops 14% when sparsity level changes from I to V. However, in LBSN datasets the accuracy of *MLP* model is always lower than Twitter and it generates similar accuracy trend as *Backstrom* model.

Backstrom, SPOT, and Friendly. These three models generate different pattern in different types of social network. For example, *SPOT* and *Friendly* have higher *Acc@160* than *Backstrom* model in Twitter data-set. However, in LBSN datasets *Backstrom* outperforms the other two models. This is because, in Twitter various factors related to social tie improve the prediction results in *SPOT* and *Friendly*, whereas the LBSN datasets lack such social factor parameters.

Mutual Prediction Ratio.

The *Mutual Prediction Ratio* (MPR) between a 'pair' of prediction models are shown in Table 3.9 - 3.12. Note that, we use data setting I and IV of Twitter microblog and Foursquare LBSN to evaluate similar predictions of model pairs within 20km of error distance. The *SLP* model returns higher MPR when pair with *UDI* and *MLP* models in Twitter dataset. This is because, these models consider user relationships and their neighbor distance as important factors in their prediction task, and similar set of users with higher node locality are predicted within a lower error distance. The *TFIDF* model produces lower MPR score with others models. This is because, the prediction approach and features used in *TFIDF* model are totally different from the remaining network-based models. On the other hand, *Backstrom* model produces a higher MPR score with *Friendly* and *SPOT* models. This is because, these models consider similar social factors such as, friendship and social closeness with the neighbor distance. The MPR scores between model pairs decrease in highly sparse datasets. For example, in FS-I, the majority of the model pairs have MPR score larger than 0.50, whereas in FS-IV the majority of the MPR scores are below 0.40.

Table 3.9: Mutual Prediction Ratio in TW-IV data

	MLP	Back	SLP	TFIDF	Friendly	SPOT	LMM
UDI	0.39	0.32	0.51	0.22	0.37	0.33	0.24
MLP	-	0.35	0.53	0.29	0.39	0.35	0.26
Back	-	-	0.37	0.31	0.42	0.59	0.32
SLP	-	-	-	0.25	0.35	0.43	0.34
TFIDF	-	-	-	-	0.30	0.24	0.20
Friendly	-	-	-	-	-	0.36	0.45
SPOT	-	-	-	-	-	-	0.29

Table 3.10: Mutual Prediction Ratio in TW-IV data

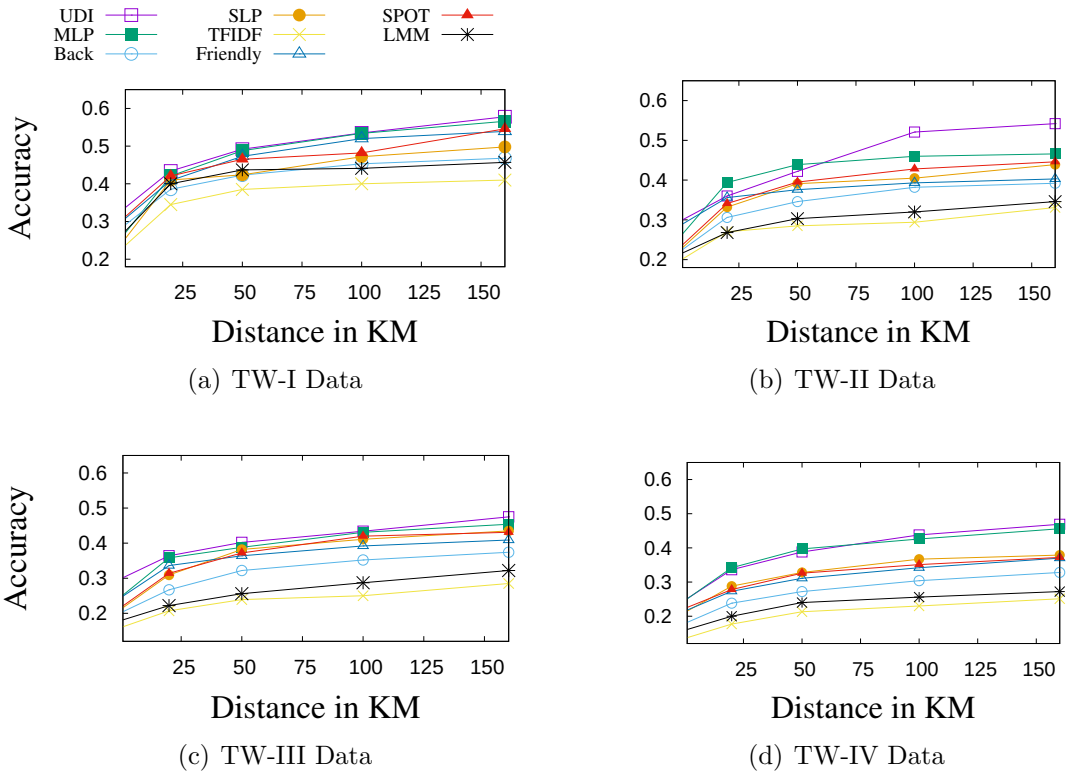
	MLP	Back	SLP	TFIDF	Friendly	SPOT	LMM
UDI	0.25	0.12	0.25	0.26	0.15	0.35	0.23
MLP	-	0.18	0.63	0.20	0.19	0.27	0.17
Back	-	-	0.20	0.11	0.25	0.45	0.18
SLP	-	-	-	0.15	0.22	0.36	0.20
TFIDF	-	-	-	-	0.11	0.15	0.16
Friendly	-	-	-	-	-	0.24	0.21
SPOT	-	-	-	-	-	-	0.18

Table 3.11: Mutual Prediction Ratio in FS-I data

	MLP	Back	SLP	Friendly	SPOT	LMM
UDI	0.67	0.64	0.63	0.60	0.58	0.51
MLP	-	0.60	0.58	0.63	0.52	0.48
Back	-	-	0.62	0.68	0.65	0.55
SLP	-	-	-	0.60	0.56	0.52
Friendly	-	-	-	-	0.52	0.45
SPOT	-	-	-	-	-	0.53

Table 3.12: Mutual Prediction Ratio in FS-IV data

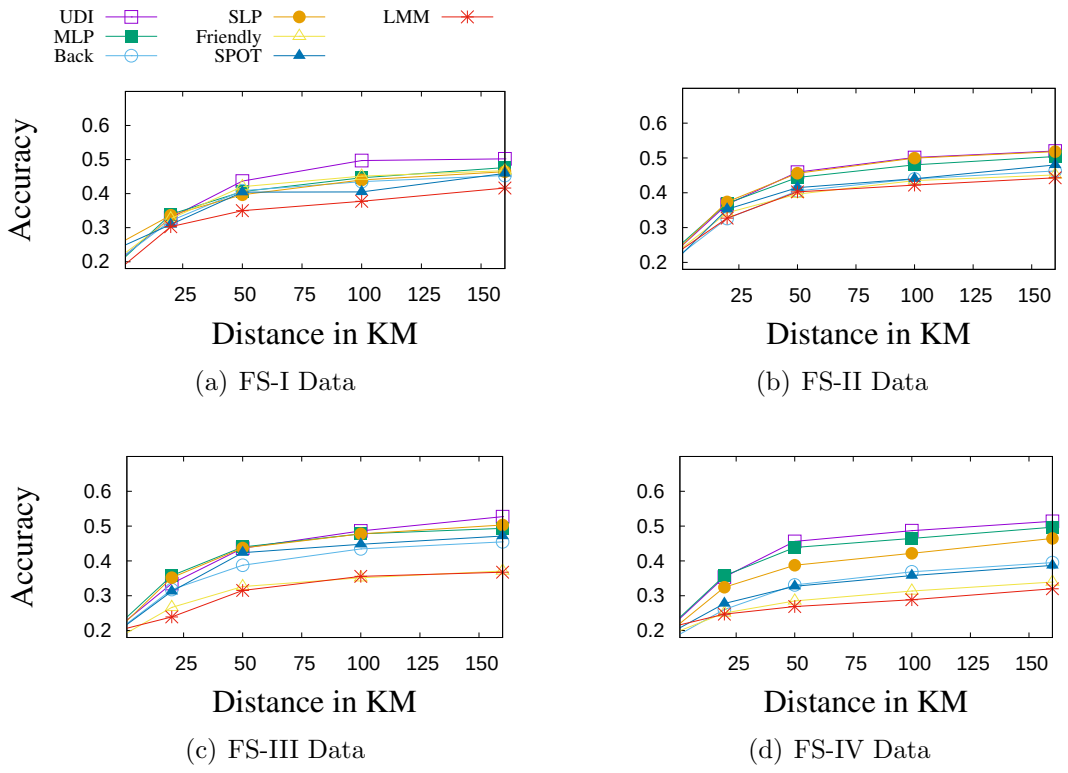
	MLP	Back	SLP	Friendly	SPOT	LMM
UDI	0.30	0.33	0.43	0.30	0.32	0.30
MLP	-	0.51	0.34	0.41	0.34	0.29
Back	-	-	0.37	0.44	0.35	0.25
SLP	-	-	-	0.34	0.36	0.31
Friendly	-	-	-	-	0.32	0.25
SPOT	-	-	-	-	-	0.23



(a) TW-I Data (b) TW-II Data

(c) TW-III Data (d) TW-IV Data

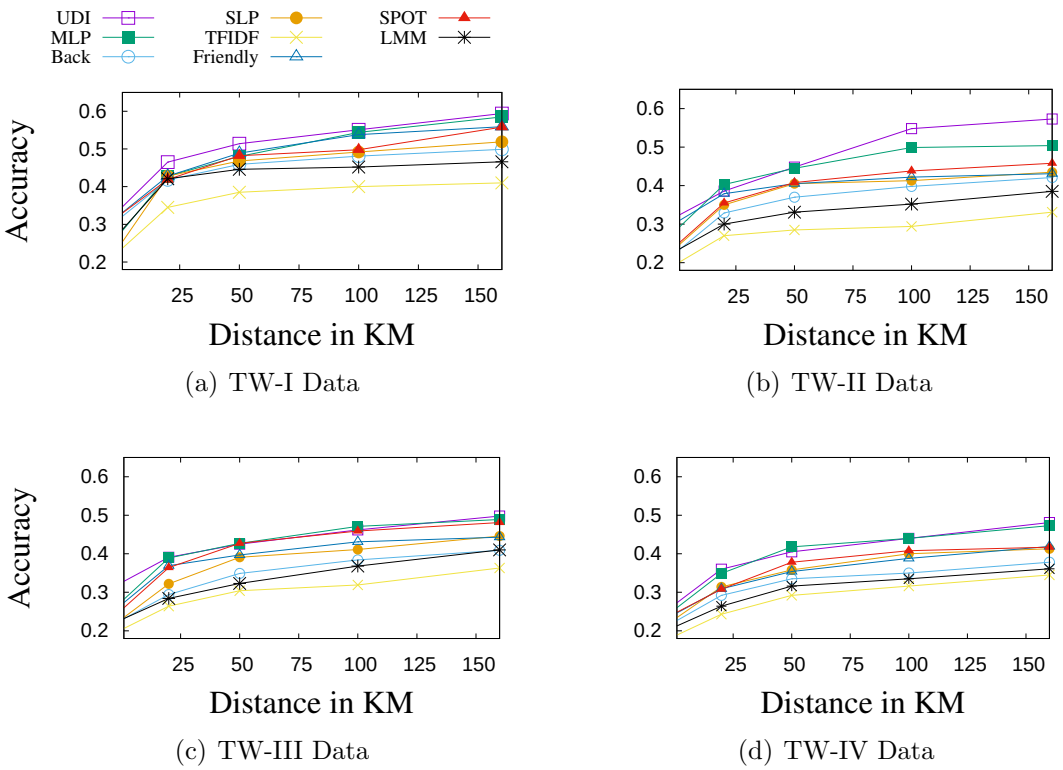
Figure 3.17: Local Prediction Accuracy of the Location Prediction Models using Twitter Dataset



(a) FS-I Data (b) FS-II Data

(c) FS-III Data (d) FS-IV Data

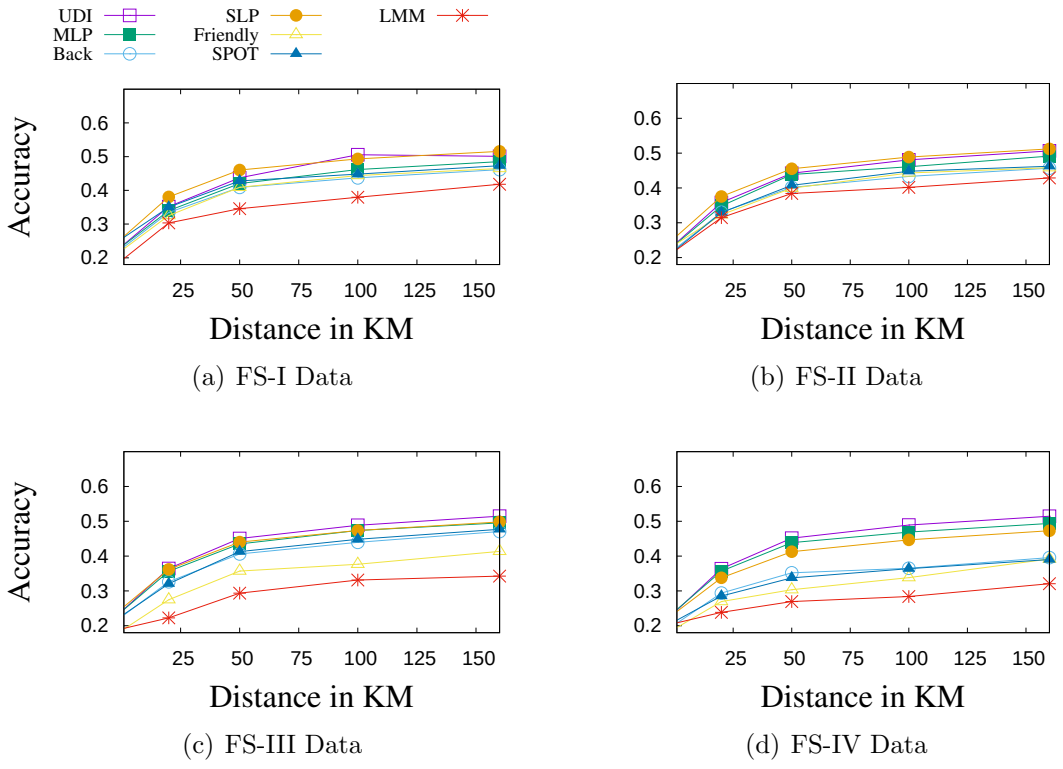
Figure 3.18: Local Prediction Accuracy of the Location Prediction Models using Foursquare Dataset



(a) TW-I Data (b) TW-II Data

(c) TW-III Data (d) TW-IV Data

Figure 3.19: Global Prediction Accuracy of the Location Prediction Models using Twitter Dataset



(a) FS-I Data (b) FS-II Data

(c) FS-III Data (d) FS-IV Data

Figure 3.20: Global Prediction Accuracy of the Location Prediction Models using Foursquare Dataset

3.5.9 Effectiveness on Local vs. Global Inference

Local Inference Technique

Local Inference (Local prediction) technique use one- or two-hop friendship information to infer users' locations. We compare the model performances using Twitter microblog (see Figure 3.17) and Foursquare (see Figure 3.18), the largest among the three LBSNs datasets for the comparison of the model performance using first four data settings (e.g. I - IV).

Considering the one-hop neighbor information, the *UDI* and *MLP* models produce higher accuracy than the other models in Twitter and Foursquare datasets. The accuracy of *SPOT* has declined by 4% (in TW-I dataset) compare to its default configuration with four iterations. In Twitter, there is no major differences in local inference accuracy of *Friendly*, *LMM* models with their default configuration. The local prediction accuracy is stable in *TFIDF* model, as the number of iteration is ineffective to the performance of this model. In both datasets, the accuracy of *Backstrom*, *Friendly* and *SPOT* decline faster with the increases of location sparsity.

Global Inference Technique

Global Inference (Global prediction) technique is used to overcome the location sparsity problem where a newly predicted location can be used further and updated iteratively to predict the locations of other users in the network. We set the number of iteration as 4 and show the accuracy of the models in Figure 3.19 and Figure 3.20 using data setting I - IV in Twitter and Foursquare data respectively.

The *Backstrom*, *Friendly*, and *LMM* models have significant improvements in accuracy compared to the local inference. For example, in TW-IV data settings the accuracy *Acc@160* increases in these three models by 5%, 5%, 9% respectively. In Foursquare FS-IV, the accuracy of these three models improves between 4% - 11%. The relative performance of the *UDI* model is quite stable in both of the datasets w.r.t the default settings. This is because, the difference in number of iterations in global inference and default setting is only one in *UDI* and hence, no significant new predictions occur. A large number of iterations may not always improve the performance of the models. We have identified *SLP* as the most sensitive model to 'number of iterations'. This model performs best in Foursquare with four and in Twitter with three iterations.

3.5.10 Effectiveness on Different Types of Users

Users with Different Node Degree

Figure 3.21 shows the performance of the location prediction models on users with different number of neighbors (i.e. node degree). Here, we are reporting the result using 'less' sparse (i.e. data setting I with 80% labeled users) and 'high' sparse (i.e. data setting IV with 20%

labeled users) data. The location inference technique of *TFIDF* model is independent on the network information, hence we exclude this model from the discussions. We make the following observations:

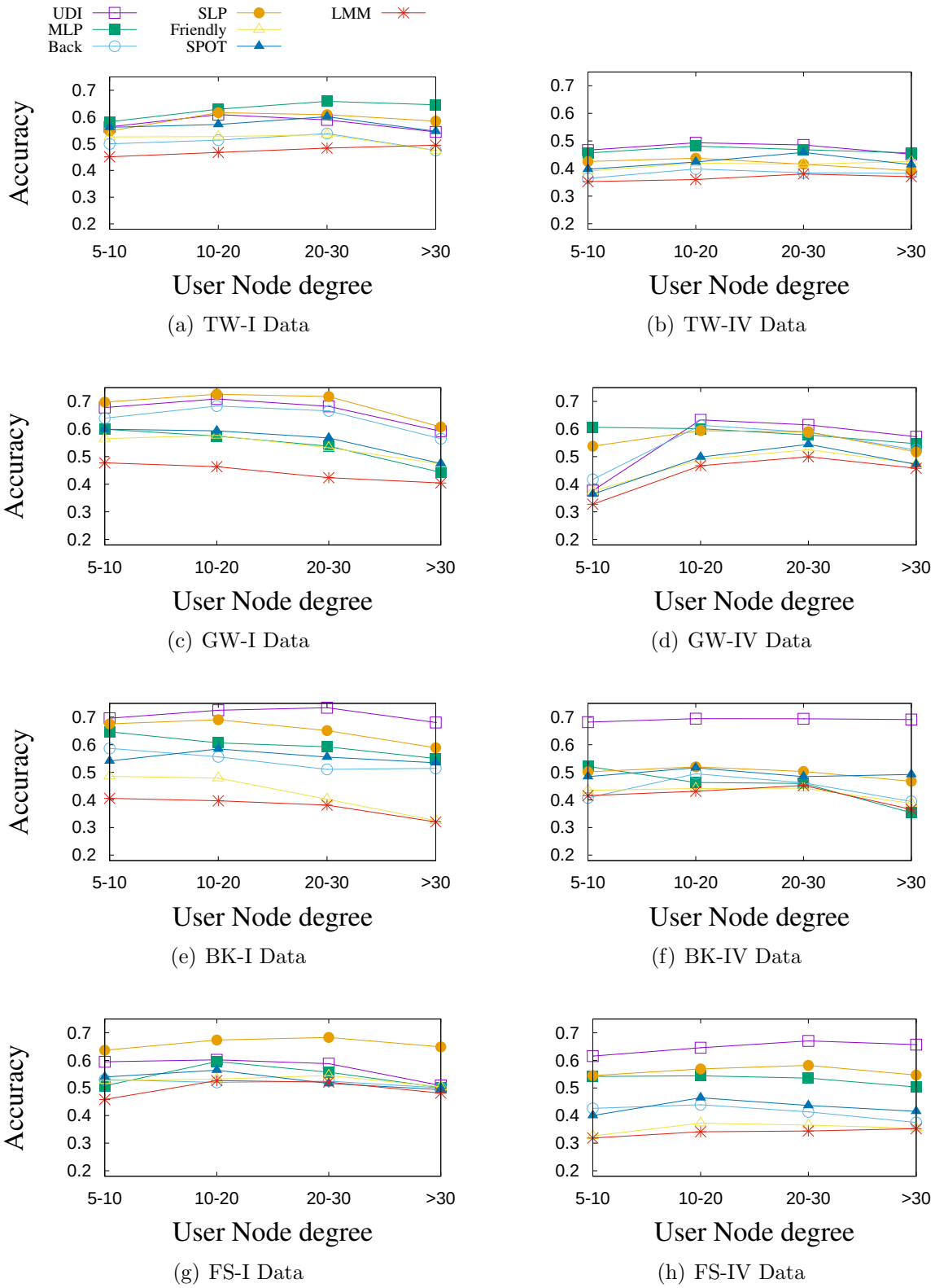
In Twitter, the relative accuracy of the models in different node degree are quite similar. In TW-IV data setting, the average accuracy decreases linearly in each model when node degree increases beyond 20. Similar pattern occurs in majority of the models in the remaining three datasets. In Gowalla, the users with node degree between 10-20 have higher accuracy in *UDI*, *MLP*, and *SLP*. In Brightkite BK-IV, the accuracy of *UDI* is constant w.r.t. different node degree ranges. In Foursquare dataset with FS-I setting, the relative accuracy of users in different range of node degree are similar with Gowalla GW-I data settings. The majority of the models obtain higher accuracy for the users who have node degree between 10 and 30. Users with a very large node degree fail to infer better locations.

Users with Different Node Locality

Figure 3.22 shows the proportion of the users predicted by each model within 160 km from the actual users' location. Each model has predicted a very small proportion of the users who have smaller node locality (i.e. less than 0.2), while a large amount of users with node locality more than 0.8 have been predicted precisely within 160 km. In Twitter, there is a steady increase in proportion of predicted users with the increase of node locality score. We notice that, there is a sudden growth in the predicted user proportion in Brightkite and Foursquare when node locality scores of the users are more than 0.8. Among the four datasets, more than 80% of users in Foursquare with node locality scores 0.8-1.0 have been predicted precisely within 160 km of error distance in *UDI*, *MLP*, and *SLP* models. In general, large proportion of the users with higher node locality are predicted precisely by the location prediction models in each dataset. This means, the predicted users are more concentrated in some locations closer to the actual locations co-shared by the neighbors.

3.5.11 User Prediction Coverage

Figure 3.23 shows the user prediction coverage of the models using default number of iterations. The relative coverage of the models in Gowalla and Brightkite are quite similar. *UDI*, *MLP*, and *SLP* models have higher prediction coverage (Figure 3.23(a)) in Twitter and the remaining models decline significantly with location sparsity. For example, in *Friendly* and *LMM*, the user coverage decline from 97% to 88% and 89% to 75% respectively when the sparsity level changes from I to IV. This is because, the default settings of these models do not consider multiple iterations. The *TFIDF* model has the lowest prediction coverage among the eight representative models. In Figure 3.24, we show the prediction coverage of the models when they are allowed to iterate only once. All the network-based models has lower prediction coverage than the default



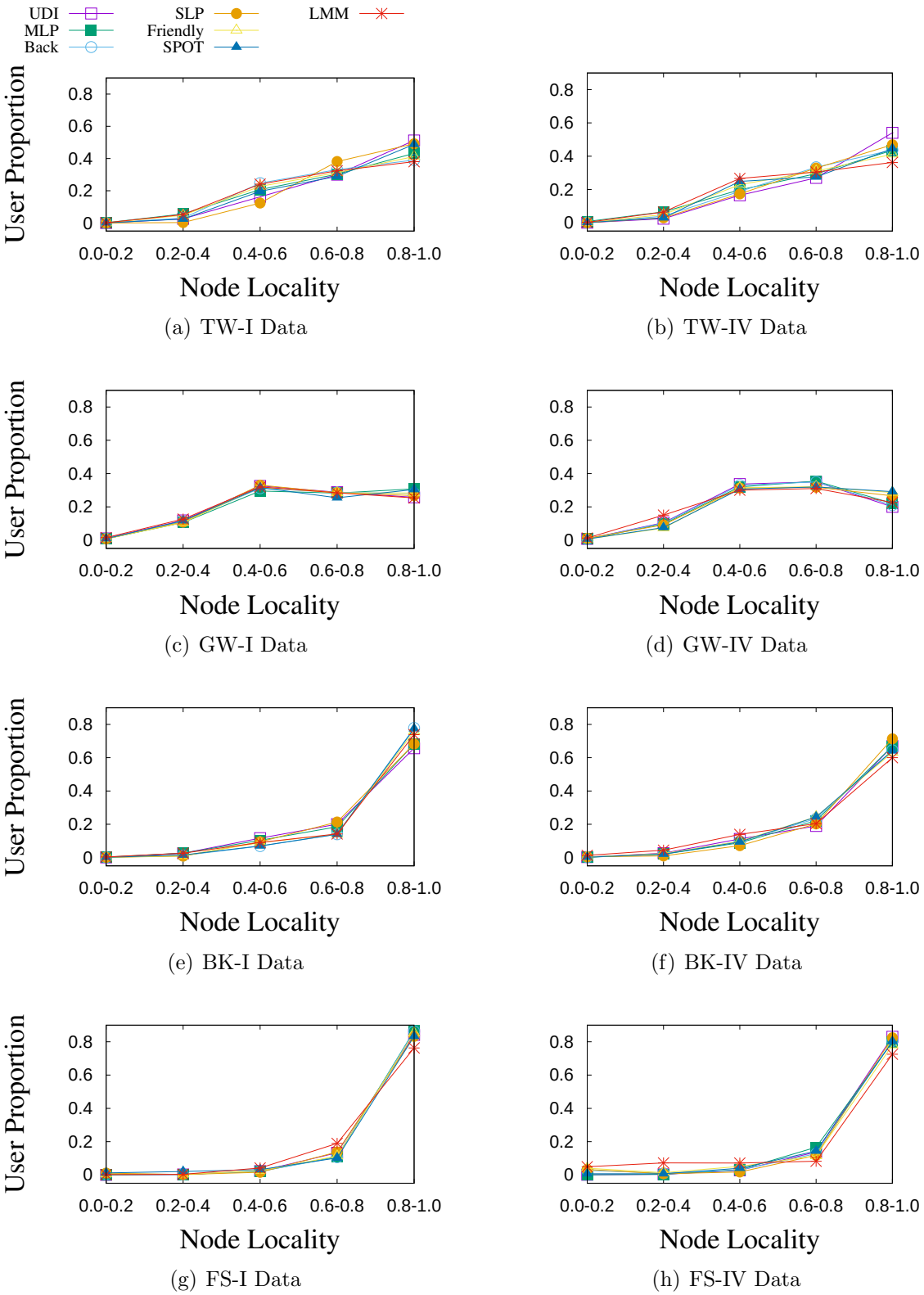
(a) TW-I Data (b) TW-IV Data

(c) GW-I Data (d) GW-IV Data

(e) BK-I Data (f) BK-IV Data

(g) FS-I Data (h) FS-IV Data

Figure 3.21: Performance of models with different node degrees



(a) TW-I Data (b) TW-IV Data

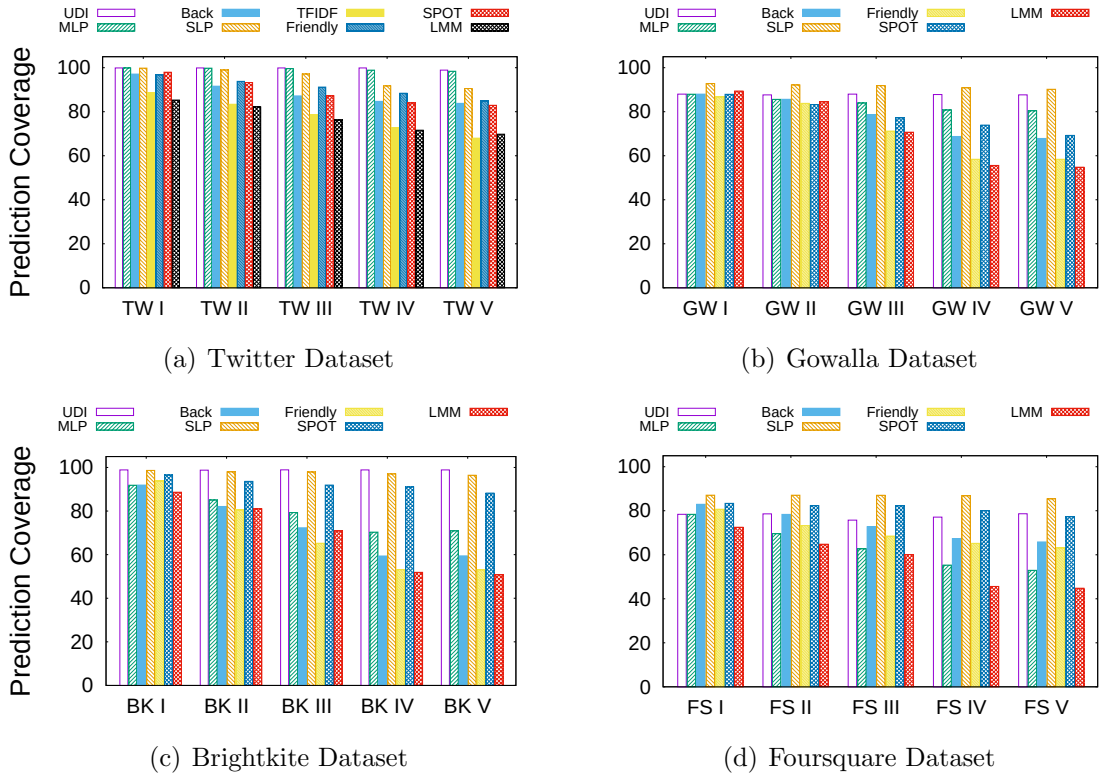
(c) GW-I Data (d) GW-IV Data

(e) BK-I Data (f) BK-IV Data

(g) FS-I Data (h) FS-IV Data

Figure 3.22: Predicted users proportion (with error distance less than 160 km) with different Node Locality

configuration. However, the prediction coverage in *SLP* model drops significantly from 'less' to 'extreme' sparsity level, e.g. it drops 21% in Foursquare. The majority of the models in default configuration execute multiple iteration and have similar coverage with the Global prediction. Hence, we do not include the Global prediction coverage here.



(a) Twitter Dataset

(b) Gowalla Dataset

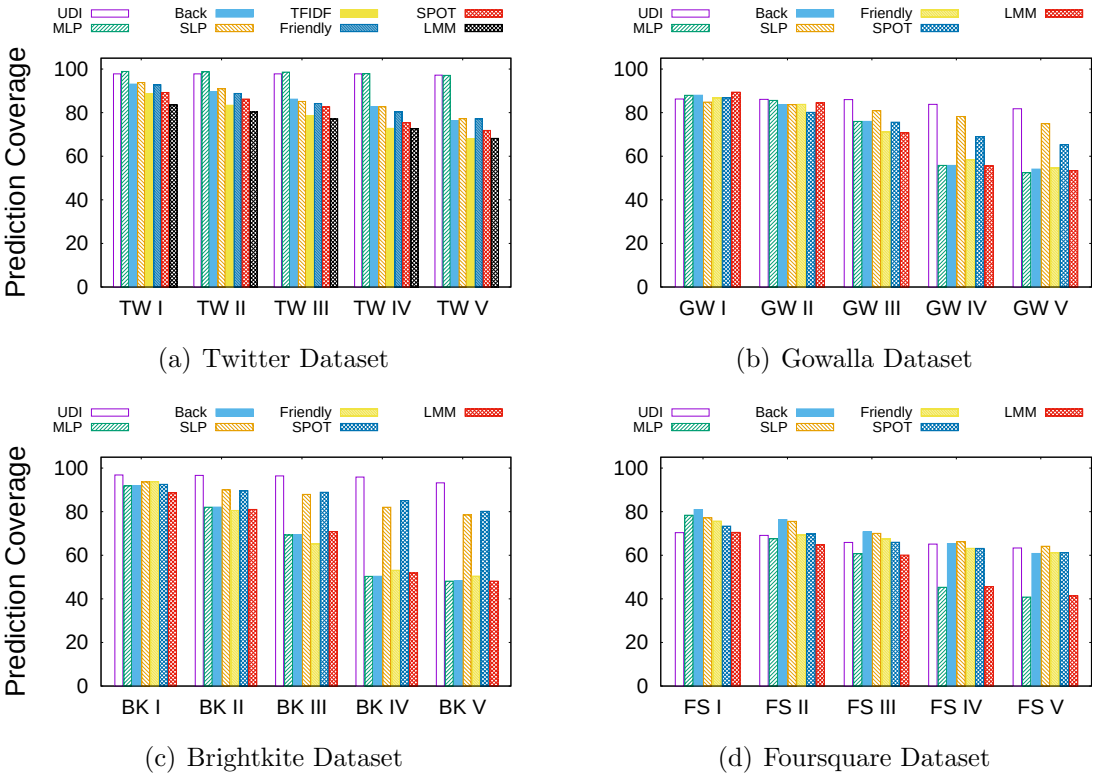
(c) Brightkite Dataset

(d) Foursquare Dataset

Figure 3.23: Prediction coverage of models in different datasets with default configuration

3.5.12 Running Time and Memory Consumption

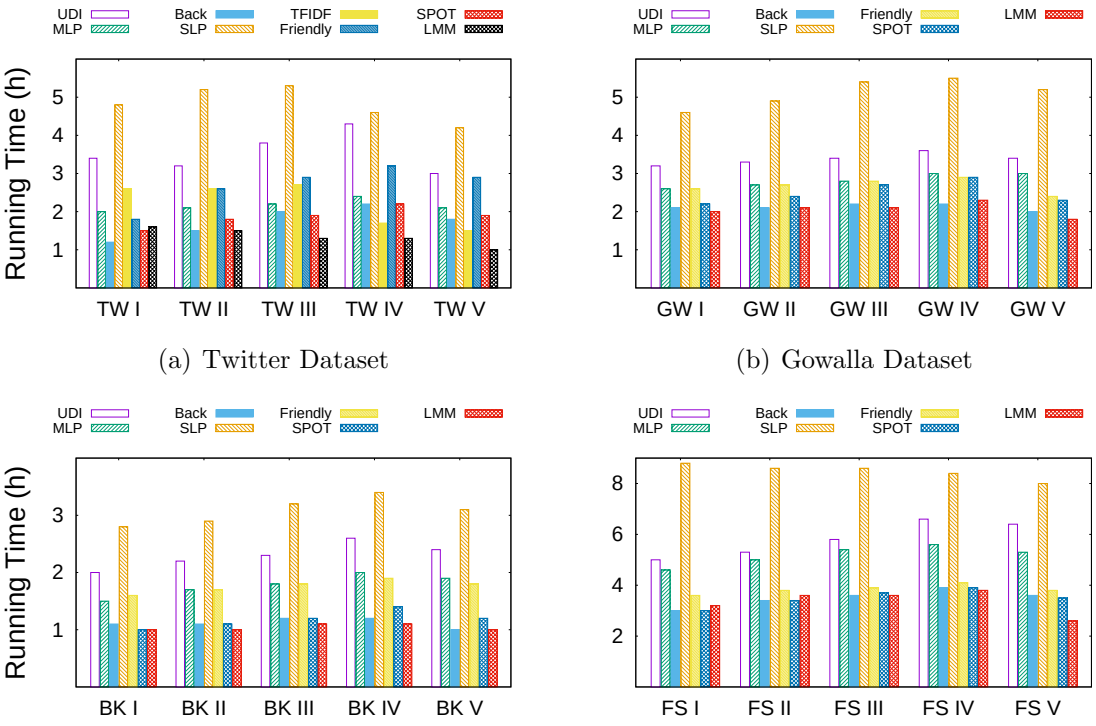
The time costs (in hour) of the model-driven prediction process are shown in Figure 3.25. The *Backstrom* model is the most time efficient among the other models, whereas *SLP* consumes the maximum time to process the four large-scale datasets. Meanwhile, the memory consumption does not vary much in different data settings, it depends on the size of dataset. We report the memory consumption in Twitter as reference to the other datasets. The *Backstrom* model consumes a lower memory of 810 MB, as it only stores the neighbor information while processing each dataset. The memory consumption of *MLP* is higher (e.g., 1725 MB), because it integrates various generative modules and each of the modules store the *following* and *messaging* information throughout the program. The remaining models have memory costs between 850 MB to 1380 MB.



(a) Twitter Dataset (b) Gowalla Dataset

(c) Brightkite Dataset (d) Foursquare Dataset

Figure 3.24: Prediction coverage of models when Local prediction is considered



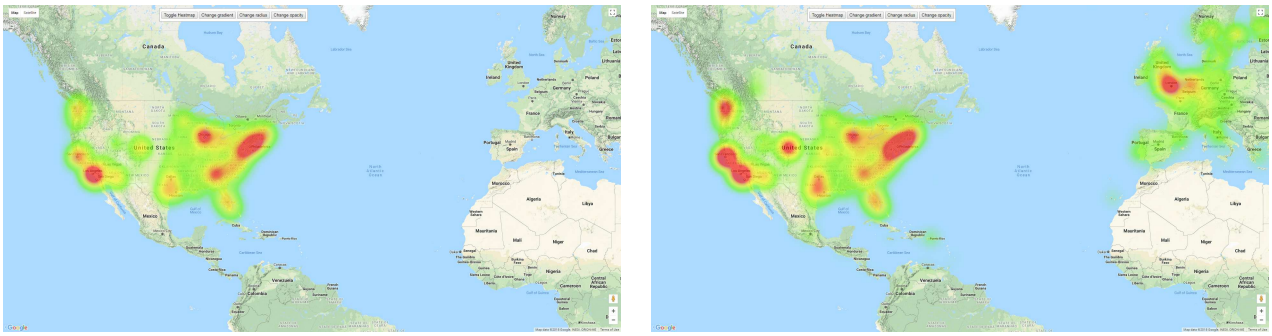
(a) Twitter Dataset (b) Gowalla Dataset

(c) Brightkite Dataset (d) Foursquare Dataset

Figure 3.25: Running Time of the different models

3.5.13 Region-specific Comparison of Overall Prediction Performance of the Models

Different social networks have different region-specific characteristics. Some models may have better performance in predicting users' locations from a specific region. Here, we compare and visualize the proportion of the users predicted within 160km of error distance with actual locations using Google maps. We choose *UDI*, *MLP*, *Backstrom*, and *SLP* models to compare the region-specific predictions in Twitter and Brightkite datasets. In Twitter, the majority of the users are distributed in New York and Los Angeles region, whereas, the users in Brightkite are spanning over New York, Los Angeles, San Fransisco, and London. In Figure 3.26, the dark red regions show relatively higher population density in the original Twitter and Brightkite datasets.



(a) Twitter (b) Brightkite

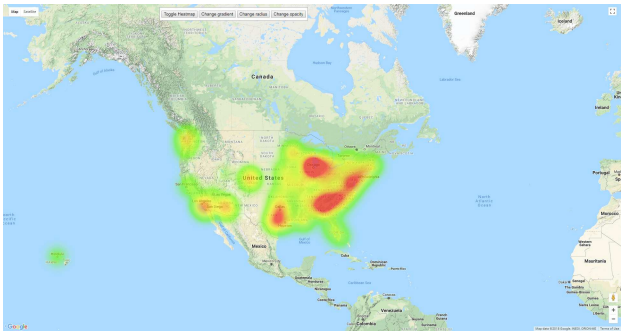
Figure 3.26: User location distribution in original datasets

In Twitter, the *UDI* model has relatively higher prediction in Chicago and Atlanta region whereas a lower prediction in Los Angeles area (Figure 3.27(a)). The prediction proportion of the other three models are identical with the original datasets. In Brightkite, the relative prediction of *UDI* and *MLP* in New York, Los Angeles, and San Fransisco area are similar with the ground-truth location distribution. However, the predicted location density using *Backstrom* and *SLP* models are slightly sparse in these three regions (refer Figure 3.27).

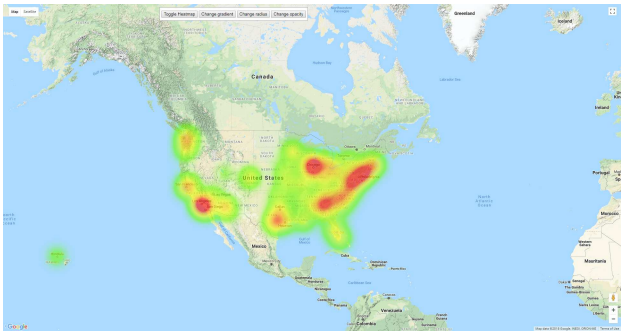
3.6 Discussions and the Findings

Our comprehensive evaluations have brought up many interesting insights that are useful for better understanding of the location prediction models. These insights are helpful in designing and optimizing models for different scenarios such as location sparsity, data type and neighbor types. We summarize a list of findings below:

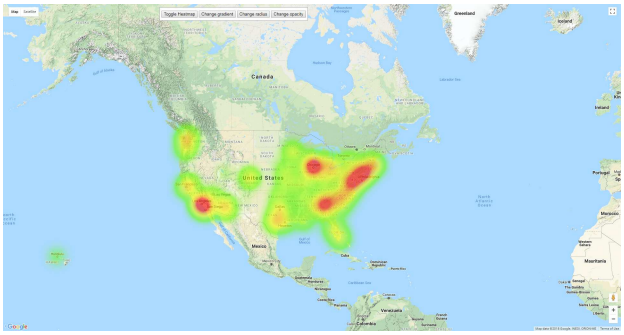
- In a dataset that has both network information and social content, the *UDI* model achieves better accuracy (Fig. 3.13).



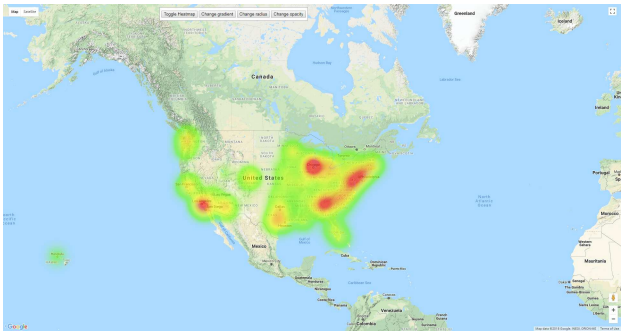
(a) UDI in TW-I data



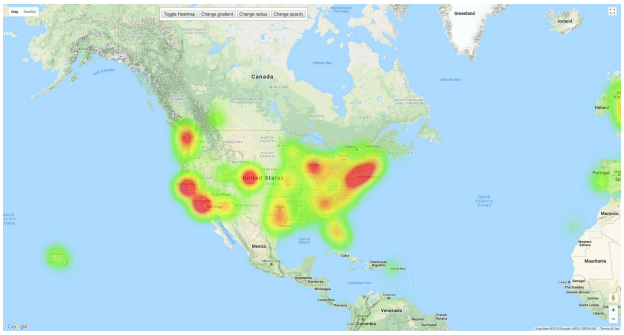
(b) MLP in TW-I data



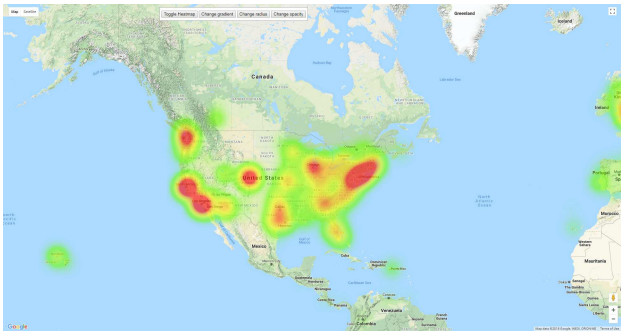
(c) BACK in TW-I data



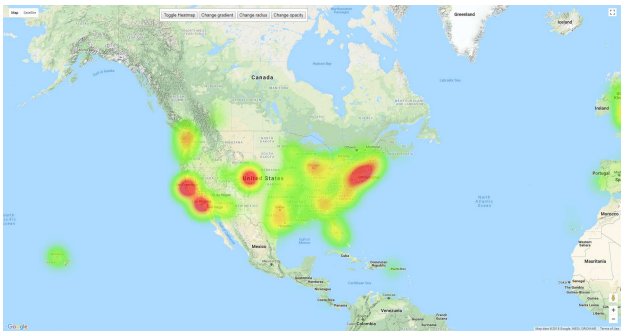
(d) SLP in TW-I data



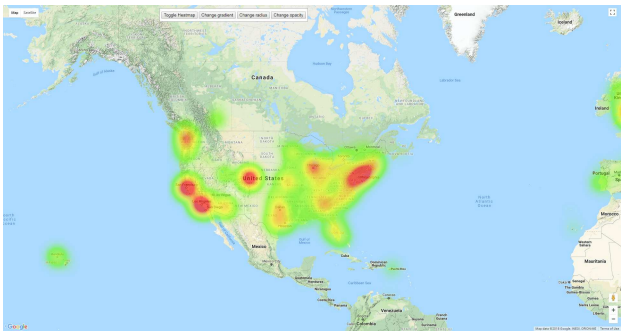
(e) UDI in BK-I data



(f) MLP in BK-I data



(g) BACK in BK-I data



(h) SLP in BK-I data

Figure 3.27: Heatmap of user prediction in different models using Twitter and Brightkite data setting I

- *SLP* model is highly sensitive to location sparsity. The prediction performance of this model drops significantly when the location sparsity increases. On the other hand, *UDI* model is less sensitive to sparsity and can predict precise users' locations in sparse data also. This is because, the inner iterations of *UDI* model execute to find the best location by updating the influence scope of each user from minimal location information (Fig. 3.13 - 3.16).
- The performance of *MLP*, *Backstrom* and *Friendly* models heavily relies on the type of data. Different social networks capture different kind of users and hence, they have different probability of friendships w.r.t. distances. Therefore, the friendship coefficient parameters must be calculated each time for a new social network dataset.
- With respect to execution speed, *UDI* model performs the best on all datasets as it uses local inference (i.e. one iteration). *SLP* model performance (accuracy and prediction coverage) drops significantly with the number of iterations (Fig. 3.13 - 3.16, Fig. 3.17 - 3.18, Fig. 3.23 - 3.24)
- In terms of training scalability, *SPOT* and *Backstrom* are the most scalable models as their preprocessing time is constant w.r.t. the number of labeled users. However, the preprocessing time of *Friendly* model increases linearly with the increase in number of labeled users.
- *Backstrom* and *LMM* are the most cost effective models, while *SLP* is the least efficient (Fig. 3.25).
- Users with moderate node degree (i.e. 10-30) have higher probabilities to be predicted precisely. A large number of neighbors may not substantiate better accuracy to users with high node degree (Fig. 3.21).
- In datasets with high declination in following probability with distace as in the case of Gowalla and Brightkite datasets, the *Backstrom* and *MLP* models perform better (Fig. 3.3, Fig. 3.14 - 3.15).
- *SLP* model has higher accuracy in datasets with properties similar to Foursquare. However, it suffers from high execution time. If execution time is not an issue, *SLP* model can be the best option to choose. Otherwise, *UDI* is a better option as it strikes a good balance between efficiency and effectiveness (Fig. 3.16, Fig. 3.25).

3.7 Summary

In this chapter, we study a comprehensive evaluation of eight representative location prediction models on four large scale real world datasets. This benchmarking study can also advance

research on social computing problems such as uncovering meaningful spatial communities, visit location recommendation, and location based event planing. We compared the prediction accuracy of the models using network properties such as friendships and interactions, neighbor proximity, location sparsity, node locality and degree etc. We have summarized our key findings of the models with different parameter settings. Our analysis shows that the effectiveness of location prediction is heavily dependent on the richness of neighbor information. The key findings of this study strongly suggest that service providers can greatly improve the quality of their services by selecting suitable location prediction models based on their application need.

Chapter 4

Geolocating Activity Location in Location-based Social Network

In the previous chapter, we have investigated several location prediction models and evaluated them on four real-world datasets using different metrics. Our all-inclusive evaluations have disclosed various interesting insights about the approaches. Among the eight models studied in the previous chapter, the location inference approaches of SLP[74], Backstrom [9], Friendly [101], LMM [159], and SPOT [78] are fully dependent on the network related information. Comparing these five network-based models that use relationship information only, we notice that SLP performs quite better in most of the dataset. However, the model suffers from high execution time. In this chapter, we propose a label propagation based algorithm to enhance the performance of SLP model while inferring the activity locations of users in location-based social network.

The users in social networks often form relationships with other users they participate in various activities nearby. The frequent check-in locations of social users in a region with most number of check-ins is important to understand the precise spatial space of the users. However, the locations of individuals in social networks are often unknown. Identifying the top activity location of a user in higher granularity level will improve various community based applications like Meetup, Groupon. In this chapter, we propose a method to infer a top activity location of social users using the implicit information of other socially connected users in the network. Our proposed approach can estimate location of a user by propagating the location information through the friendship edges by maintaining an inference sequence. The inference sequence establishes an priority to predict location for the users first which have a higher chance with lower error distances. We find that the proposed method significantly improve the state-of-the-art network-based location inference techniques in terms of accuracy and efficiency.

Chapter map. In Section 4.1, we give an overall introduction of activity location prediction problem. We formalize the problem in Section 4.2. After that, we propose our methodology in Section 4.3 and the check-in characteristics of two LBSN datasets are discussed in Section 4.4. The experimental results are reported in Section 4.5. Finally, we concluded this chapter in

Section 4.6.

4.1 Introduction

Due to the proliferation of mobile devices, a large collection of location related information is available in social networks. Some of these social networks are dedicated to location sharing, while others provide location-based features together with other social networking services, e.g., "check-in" services. Such services allow users to annotate their activities with granular location information derived from GPS enabled devices. Thousands of users have adopted the location services which helps to search for the social friends who are active in a particular spatial region. The Location-based Social Networks (LBSNs) facilitate users to share their location easily, and connect with other individuals in their activity regions.

A user may be associated with a large number of locations (e.g., Point-of-Interests (POIs), home location, work places) in her timeline. However, not all such locations are equally important to their social connections as they may reside far from their activity areas. Among the multiple check-ins, a user has the maximum chance to be located at their top activity location. Additionally, it is not easy to obtain such locations in the network as they do not allow the LBSN applications to disclose their check-in information in the public. Hence, it is useful to identify the best activity location of the users that can be inferred from using the activity locations of neighbors. Inferring such locations of each user can effectively reduce the spatial search space without affecting the socio-spatial relationships in a network. For example, in case of emergency, an application may smartly targeted to the activity locations of an affected region where users have higher chance to be co-located.

Due to the importance of location information in online applications [15, 87], a significant amount of efforts have been carried out to infer the user location in traditional the social networks. However, majority of the works analyze the contents produced by the users in the network to identify their home location. The other popular location estimation model is network-based where a user's location is derived from the known location information of the nearby users in the social network. One of the well-known network based approaches, FIND [9], selects a location for a user that maximizes the probability of friendships given the distance between the location candidates and friends' home locations. However, the performance of such model is highly dependent on the ground-truth information available in the network [63]. Another network-based location inference model SLP [74] propagates location label to their neighbors. This model iteratively assigns the location with the updated geometric median using neighbors location. However, the SLP [74] model suffers from several issues: (1) an incorrect location estimation of a friend may lead to increase error distance. (2) if some noisy social connections exists in the network, the estimated locations may shift far from the original locations. (3) the process consumes much time to converge the location of each user as SLP considers 'all' the

neighbors in the process irrespective of their importance to the user.

To address these challenges, we propose a location inference approach that considers the social relationships in a network and can incorporate the location granularity. In this work, we are particularly interested to predict the "activity locations" for social users using the network information only. We define a user's *activity location* as the place among the highly checked-in region where most of her activities occur. Our intuition is that the activity location of a user should be co-located with one of her friends' most frequent check-in location. The proposed model can effectively propagate the location information using small amount of ground truth location. We also define an inference sequence to effectively converge the location propagation process. In order to improve the location inference accuracy, we discard the friendship information that may generate some noisy estimations.

4.2 Problem Definition

In this section, we define the terminologies and briefly discuss the basic concept of location estimation in online social network.

Definition 4 (Activity Location)**.** *An activity location of a user is the location point where the user has higher chance to be available at the location.*

In this research problem, we will select the activity location of the social users. The other types of locations in social media are found as home location, work location, post (e.g., tweet) locations.

Definition 5 (Location based Social Network)**.** *A Location based Social Network is a graph consists of set users and locations entities, and the relationships between the entities. It can be defined as social graph $G(V, E, L, E')$, where V is the set of social users, E is the relationship edges between the users in V, L denotes the location set, and the check-in edges between users and locations are denoted by E'.*

When a user follows another user, we call the users as neighbor or friend to each other. The vertex set in G is composed of two types of user sets $V = V^* \cup V^N$, where V^* is a set of labeled users whose activity locations are known (in the form of latitude, longitude pair) and V^N be the set of unlabeled users whose activity locations are not known.

Definition 6 (Location Inference)**.** *Given a social network $G(V^* \cup V^N, E, L, E')$, the location inference problem annotates the unlabeled user set $\hat{V}$ with the locations to be selected from L such that the estimated top activity location $\hat{l}_u$ of $u \in \hat{V}$ is close to the actual location l_u.*

4.3 Methodology

Given a social network, identifying the nearest friends can provide strong evidence of an individual's location. However, the key problem is that, it is not easy to choose which friends to be selected as the reference to infer user location. Selecting all the friends for the location inference task may lead to consume high time cost. Again, inclusion of a noisy relationship will decrease the overall accuracy of a model. Therefore, for better prediction performance by a model, we need to ignore the noisy relationship in the datasets. Besides, the check-in location information of users may be sparse. Hence, to reduce the data sparsity problem and to enhance the location estimation coverage, it is essential to propagate the newly estimated locations of neighbors to the network. On the other hand, a proper inference sequence is important to improve the overall accuracy of the inference task. Therefore, we propose a new method for activity location estimation task using sequential label propagation where a series of heuristics is used to define the propagation order.

4.3.1 Selecting Neighbors for Location Inference

Friends having activity locations near to a user's activity location are useful to infer the location of a user more accurately. Therefore, we will first identify such neighbors of the social user which have higher chance to be co-located at some activity locations. At the same time, we can ignore those friends who may not provide better suggestions on estimating the locations of the unlabeled users.

Neighbor Validation. In our proposed algorithm, we first validate the neighbors of the unlabeled users whose locations can be used to further propagate in the network. To do so, we first validate the activity location of labeled social users based on the location information of their neighbors. This step is useful to filter some neighbors (e.g., celebrity user) who may not be a local friend to the user. Among the check-ins of a user, we select the most frequent location point from dense check-in area as the activity location of a labeled user. The detailed description about the ground-truth information related to the activity location is provided at 'Ground-truth Information' paragraph in Section 4.4.

In our proposed method, to filter some noisy relationships, we ignore the friends whose majority neighbors do not have activity locations nearby. We first verify the location of a user based on the relationship between distance and the probability of being friends by modeling the relationships between distances and the friendship probability. To do this, we simultaneously select each labeled user to validate whether the location information of the user can be used to propagate to predict location of their neighbors. Therefore, after selecting a labeled user, we mast her location first and based on the location distribution of her labeled neighbors, a location is assigned among the neighbors' activity location which has maximum likelihood [9].

If the newly assigned location of the use resides far from her original activity location, we will ignore that user to infer location for her friend. We provide a detailed description of the process below.

According to Backstorm et al. [9], the probability of friendship is roughly inversely proportional to the physical distance between the social friends. Given a distance $|l_u - l_v|$ between two users u and v, the probability of having an following relationship between u, v is measured as $p(|l_u - l_v|) = a(b + |l_u - l_v|)^{-c}$. The value of the constants can be empirically determined using the population distribution of the datasets as mentioned in [9]. Therefore, a location l_u co-located with u's friends is considered as the location of user u if, Equation 4.1 is maximized.

$$\gamma(l_u) = \prod_{e(u,v_j) \in E_F} \frac{p(|l_u - l_v|)}{1 - p(|l_u - l_v|)}, \qquad l_u \neq l_v \tag{4.1}$$

Let us assume, the above equation is maximized for the location point l_{u_est} and the original location of u be l_{u_org}. We heuristically ignore the user in estimating her neighbors' locations, if $|l_{u\_est} - l_{u\_org}| > 160KM$. This practice will filter the long distance relationships and hence, will enhance both the accuracy and efficiency of our proposed model.

Social Closeness. In an LBSN, the socially close friends can provide strong evidence about users' locations [78]. By studying the relationship between social closeness and geo-distance, our proposed model can consider those neighbors which have strong social relationship with the query user. In our proposed approach, we will discard some neighbors' location information if they do not have strong social bonding with the target user. The intuition is that, some active and strong relationship edges may carry substantial evidence about friend's activity location. For example, the edges between a user and co-workers can contribute strongly to the likelihood that the activity locations of the user are nearby to her co-workers. Therefore, by selecting the edges based on the social closeness, our model can significantly improve the estimation accuracy, particularly when the social users have few friends in their social connection.

In a study by Kong et al [78], it is mentioned that a pair of social friends has 83% of chance to live within 10KM, if they have common friends more than 50% of the total friends of the users. They also indicate that the friendship probability decreases to 2.4% when the ratio of the common friend declined to 10%. This development holds our hypothesis that social distance can help us to identify the best suitable friends in the location estimation. Thus we use social closeness to identify the important neighbors that can enhance the location estimation accuracy. The social closeness [78] between two socially connected users u and v are calculated as $S_{sc}(u, v) = |nbr(u) \cap nbr(v)| \sqrt{|nbr(u)||nbr(v)|}$, s.t., $nbr(u) \neq \emptyset$, $nbr(v) \neq \emptyset$. Lets, assume u be the target user (whose location should be inferred) and v be one of the social friends of u. If the social closeness score is $S_{sc}(u, v)$ (w.r.t. friend v) is more than a threshold τ, e.g., $S_{sc}(u, v) > \tau$, we will select the friend v for estimating the location of the user u. The value of

τ is dataset dependent and a higher score of τ may consider a few friends will be considered for the location inference process.

4.3.2 Creating User Sequence for Location Inference

In the above section, we discuss the importance of selecting the 'meaningful' neighbors in the process of activity location selection in a social network. The key problem in label propagation algorithms is how to set the priority of inferring locations of the unlabeled users. Here, we will discuss the process of generating a proper inference sequence such that the newly estimated location of neighbors can be allowed effectively propagate to infer the location of other users in the network. The intuition is that, if the location of a user is estimated correctly, it will be useful to be used the location information to further propagate in the network. This strategy will help to increase the effectiveness of our label propagation based model. To do so, we define the following criteria to sort the unlabeled user nodes in the priority queue:

- CR 1. users having number of labeled neighbors closer to their mean location point μ, where $\mu = \frac{1}{|\widehat{nbr}(u)|} \sum_{v \in \widehat{nbr}(u)} l(v)$ and $\widehat{nbr}(u)$ are the set of u's labeled neighbors.
- CR 2. users with number of labeled neighbors
- CR 3. users with number of total neighbors (both labeled and unlabeled)

The first criterion is to give higher priority to the unlabeled user nodes having large number of labeled neighbors' locations near to their mean location point μ. This criterion can increase the location estimation efficiency as neighbors concentrated over a particular region can be quickly converged to a location point. Such process will improve the accuracy as friends are usually co-located in a concentrated region [159] and the mean of the neighbors' locations will be the center of the region. Further, the ties are broken using the criteria 2 and 3. The second criterion is to give higher priority to the unlabeled users that have more labeled neighbors. This is because estimating locations of such users with higher priority have higher chance to have lower error distance, and hence the estimated location can be useful to propagate in the network. The third criterion is to give higher priority to the users with the number of neighbors, as such user nodes have higher chance to be inferred using multiple iterations.

4.3.3 Algorithm

We develop an iterative algorithm, Sequential Spatial Label Propagation (SSLP), based on the above mentioned intuitions and criteria. The pseudo-code of SSLP is shown in Algorithm 2. First, we initialize the parameter $\tau = 0.1$ to select those neighbors having social closeness score more than 0.1. The value of τ can be set according to the population distribution. A higher τ will discard more neighbors in the process of location propagation to the network. In

lines 2-6, the labeled neighbors are validated as mentioned in Section 4.3.1 and the filtered labeled neighbors are selected corresponding to each unlabeled user u_i. Next, at line 7, we create the unlabeled user sequence, e.g., SeqV, following the criteria mentioned in Section 4.3.2. These steps are part of the pre-processing.

The steps at line 8 to line 17 of Algorithm 2 do the iterative computation. The outer iteration continues until the convergence criteria are satisfied. Further, based on the priority mentioned in $SeqV$, in each iteration, we estimate location of each unlabeled user (line 15) and update the locations of the users at the same time. At the end of each iteration, we further update the inference sequence $SeqV$ at line 17 using the criteria mentioned in Section 4.3.2. The whole inference process continues until the convergence conditions are satisfied.

Algorithm 2: Sequential Spatial Label Propagation

Input: LBSN graph $G = (V^* \cup V^N, E, L, E')$
Output: $\widehat{V}^N$

1 Initialize: $\tau = 0.1$
2 **for** *each user* u_i *in* V^N **do**
3 **for** *each neighbor* v_i *in nbr*(u_i) **do**
4 **if** $S_{sc}(u_i, v_i) \geq \tau$ **then**
5 $N(u_i) \leftarrow N(u_i) \cup v_i$
6 $\hat{N}(u_i) \leftarrow \text{ngbrValidation}(u_i, N(u_i), G)$
7 SeqV $\leftarrow \text{createSequence}(V^N, G)$
8 **while** *convergence criteria not met* **do**
9 **for** *each user* u_i *in SeqV* **do**
10 $L(u_i) \leftarrow \emptyset$
11 **for** $nbr \in \hat{N}(u_i)$ **do**
12 **if** $l(nbr) \neq \emptyset$ **then**
13 $L(u_i) \leftarrow L(u_i) \cup l(nbr)$
14 **if** $L(u_i) \neq \emptyset$ **then**
15 $l'(u_i) \leftarrow \arg\min_{l_i \in L(u_i)} \sum_{l_j \in L(u_i)} dist(l_i, l_j)$
16 update $\widehat{V}^N \leftarrow \{l'(u_i)\}$
17 SeqV $\leftarrow \text{updateSequence}(\widehat{V}^N, G)$)

4.4 Location-based Social Network Datasets and Characteristics

The study of geographical properties of the users in Online Social Network (OSN) has drawn intensive attention in recent years [101]. In this section, we discuss various socio-spatial characteristics of the location-based social network datasets. This discussions will help us to

understand the relationships between the social users and their checked-in locations in the network.

Characterizing network properties to discover the relationships between activity region and social users have been studied in [72, 136, 157, 162]. Some studies [104, 146] have argued that location influence is one of the important factors to build social relationship with users. However, Scellato et al. [136] investigate the relationship between distance and location, and they showed that the social connections in location-based social networks are not exclusively determined by the geographical or social factors. They also showed that, if a user has more connections, their friends have higher chance to be distributed in larger spatial space.

In this study, we use two real-world datasets, e.g., Brightkite and Gowalla. The datasets have social network information, and the social users in these datasets are associated with check-in locations (latitude, longitude). We denote the users as vertex and the relationship between two users are depicted by an undirected edge if one user follows another. The descriptions about the datasets are given below:

Brightkite Dataset. The Brightkite dataset is collected from SNAP[1] repository and the original data was gathered over the period Apr. 2008 - Oct. 2010. The dataset contains 51,406 social users. We have removed some users from the dataset which have few check-ins and friends less than 5. Finally, we have selected 50,686 users who have sufficient check-in information. We build the social network using the selected users. The average user degree of the dataset is 7.67.

Gowalla Dataset. The Gowalla dataset is also collected from the SNAP repository. It contains the check-in information of 107,092 users with a total 6,442,892 check-ins in 1,280,969 places. The dataset was collected over the period of Feb. 2009 to Oct. 2010. We create the social network using the following information available in the dataset. The summary of the two datasets used in this work are is shown at Table 4.1. The average degree of the dataset is calculated as 8.53.

Ground-truth Information. The check-in data of a user can be seen in multiple locations. Therefore, we notice that majority of the users are associated with multiple locations with less than 100 check-in in each dataset. To select the ground-truth activity location, we follow a similar approach as [26]. We first discretize the spherical earth surface into 0.2 degree by 0.2 degree cells. For a given user, we find the cell with the dense check-ins [136] and within this cell, we select the co-ordinate of most frequent check-in location of the user as the ground-truth activity location.

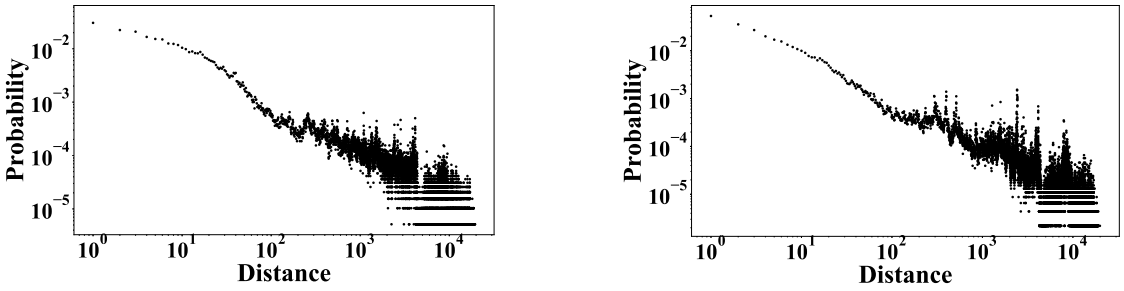
Check-in Characteristics. In order to understand the dynamic between the users' friendship and geography, we first investigate the probability of friendships as a function of distance. Naturally, we expect that the probability of following social friends to go down with the distance.

[1]http://snap.stanford.edu/data/index.html

Table 4.1: Statistics of the datasets.

Dataset Name	#Users	#Edges	#Checkins	#Places	Avg. User Degree	Avg. Neighbor Distance (KM)	Average Node Locality
Brightkite	51,406	197,167	4,491,143	772,783	7.67	1,819	0.69
Gowalla	107,092	456,830	6,442,892	1,280,969	8.53	1,722	0.52

In Figure 4.1, we observe that the probability of a user is likely to follow some friends who live near. This demonstrates that the nearest neighbors are highly predictive of the individual locations. However, the users may also follow some users who are living away.



(a) Brightkite (b) Gowalla

Figure 4.1: Probability of Friendship as a function of distance

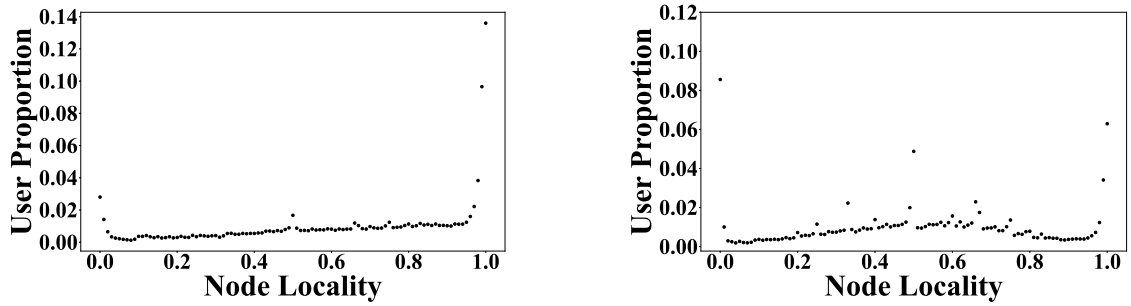
To generate the curve in Figure 4.1 on friendship distribution as a function of distance, we first count the number of friend pairs w.r.t. distance between their activity location. Then, we plot the proportion of the number of such pairs w.r.t. the distances. The distance-probability plot can be fit into power law distribution curve $f(x) = \beta \cdot x^{-\alpha}$. We found the value of (α, β) as (-1.14, 0.20) and (-1.52, 0.612) in Brightkite and Gowalla datasets respectively. The value of the exponent α near to -1 suggests that the probability of friendship is roughly inversely proportional to the distance.

Node Locality. The *node locality* metric is useful to quantify the geographic closeness of the neighbors to a certain user node. This metric determines how close a user is with her neighbors. The *node locality* of a user u_i with 1-hop neighbors are calculated as [134]:

$$NL(u_i) = \frac{1}{|ngbr(u_i)|} \times \sum_{v_j \in ngbr(u_i)} e^{-\frac{d(u_i,v_j)}{\beta}},$$

where β is a scaling factor which avoids extremely small values of *node locality* when the neighbor distance is very large. β is calculated as follow:

$$\beta = \frac{1}{|E|} \times \sum_{u,v \in V, e(u,v) \in E} d(u,v)$$



(a) Brightkite (b) Gowalla

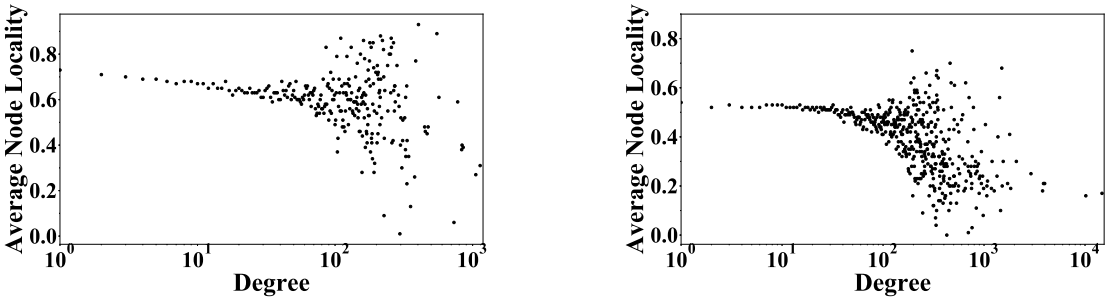
Figure 4.2: Cumulative Distribution on Node Locality

In Table 4.1, the average neighbor distance and the average node locality of Gowalla and Brightkite datasets are given. The average neighbour distance may not give a fair idea about the neighbour distribution on a spatial space, as a few neighbors residing far will generate a higher average neighbor distance. However, the average node locality of a dataset can provide a better assumption about the relative neighbor distributions. A dataset with higher average node locality have large number of social connection within a close geographic distance [134]. Gowalla dataset has a lower node locality score (0.52), which means users in Gowalla are engaged with a geographically spread set of individuals.

In Figure 4.2, the cumulative distribution on Node Locality of the datasets are shown. In Brightkite, a significant fraction of the users have node locality score close to 1. This means, there exist a large portion of the users in Brightkite dataset who have social connections with neighbors within a close geographic region. The average node locality score in Brightkite dataset is reported as 0.69. In Gowalla, the effect of node locality is weaker; a non-negligible fraction of users have node locality value close to 0, which means the users have social connections in far distances. We measure the average Node Locality score 0.52 in Gowalla dataset.

We also determine the correlation between the node degree and the node locality using the two datasets. Measuring such correlation is useful to understand the spatio-social properties of users with different node degree. In Figure 4.3, the average node locality as a function of the node degree is shown for each dataset. The trends show for the lower degree nodes (e.g. less than 100) in each of the four dataset are fairly constant. However the users with more than 100 node degree have wider node locality scores between 0.2 to 0.8. Since, it is statically more likely that the nodes with higher degree are more likely to be connected to the distant users, which

may exhibit a smaller locality values. Such behavior is observed in Gowalla dataset, i.e., the average locality drops significantly with the increase of the node degree. Interestingly, some of the users with higher node degree have also higher node locality score in Brightkite dataset. Such users have rich set of neighbors available within a close spatial region.



(a) Brightkite (b) Gowalla

Figure 4.3: Average Node Locality as function of degree

4.5 Experiments and Results

In this section, we conduct the experiments on two large scale datasets (e.g., Brightkite and Gowalla) and show the effectiveness of our methods from different aspects. The detailed description of the datasets is given in the previous section. For the experiment purpose, we divide each dataset in two different data settings. The first data setting contains 20% of the total users as unlabeled and the second data setting has 80% unlabeled user information. Our target is to annotate the unlabeled users in each data setting using the labeled user information. Nevertheless, the first data setting has large number of labeled users than the second data setting. We denote the two data settings with ID as BK20, BK80 and GW20, GW80 of the Brightkite and Gowalla datasets, respectively. The labeled-unlabeled user ratio in the second data settings (e.g., in BK80 and GW80) are more close to the real-world scenario where majority of the users do not disclose their location in the network [89, 159]. However, some of the existing location prediction models [74, 90] consider the labeled-unlabeled user ratio similar as first data setting. Therefore, we consider both the data settings to compare the models in different sparsity levels. The summary of the labeled and unlabeled user information is given in Table 4.2. All the experiments were conducted on a Windows environment with Intel i7 CPU and 40 GB memory. We report our results based on the average of five executions of the algorithms.

Methods. To fully evaluate our proposed algorithm, we compare with three state-of-the-art methods FIND [9], SLP [74], and Friendly [101]. FIND is one of the primitive models for the location inference task which studied the interplay between geographic distance and social relationships of the users. SLP infers a user's location by propagating the location labels among

Table 4.2: Summary of the Labeled and Unlabeled User Information.

Dataset ID	# Unlabeled users	#Labeled users	Average Labeled neighbors
BK20	20,274	30,412	4.93
BK80	40,549	10,137	2.02
GW20	42,830	64,262	8.85
GW80	85,674	21,418	5.43

their neighbors. The SPOT model is based on the hypothesis that social distance can identify the closest friends in location estimation [63].

Metrics. To compare the effectiveness and efficiency of the methods, we use the following measures.

Average Error Distance (AED). Let, $Err(u_i)$ be the error distance in KM unit between user u_i's activity location and the estimated location. For a set of users U, the AED is measured as $AED(U) = \frac{1}{|U|} \cdot (\sum_{u_i \in U} Err(u_i))$. A model with a lower value in AED confirms that the majority of the unlabeled users are assigned with a location that are near to the activity location of the corresponding users.

Accuracy. The metric *Accuracy* is the proportion of the correctly predicted users within a certain error distance (e.g., 160KM). For a set of users U, the accuracy is defined as, $ACC(U) = \frac{\{u_i | u_i \in U \wedge Err(u_i) \leq d\}}{|U|}$. Here d is the error distance threshold set for the experiment purposes.

Inference Coverage. Among the set of users U, if a model assigns $\widehat{U}$ users with a location, the inference coverage is calculated as $\frac{|\widehat{U}|}{|U|} \times 100$.

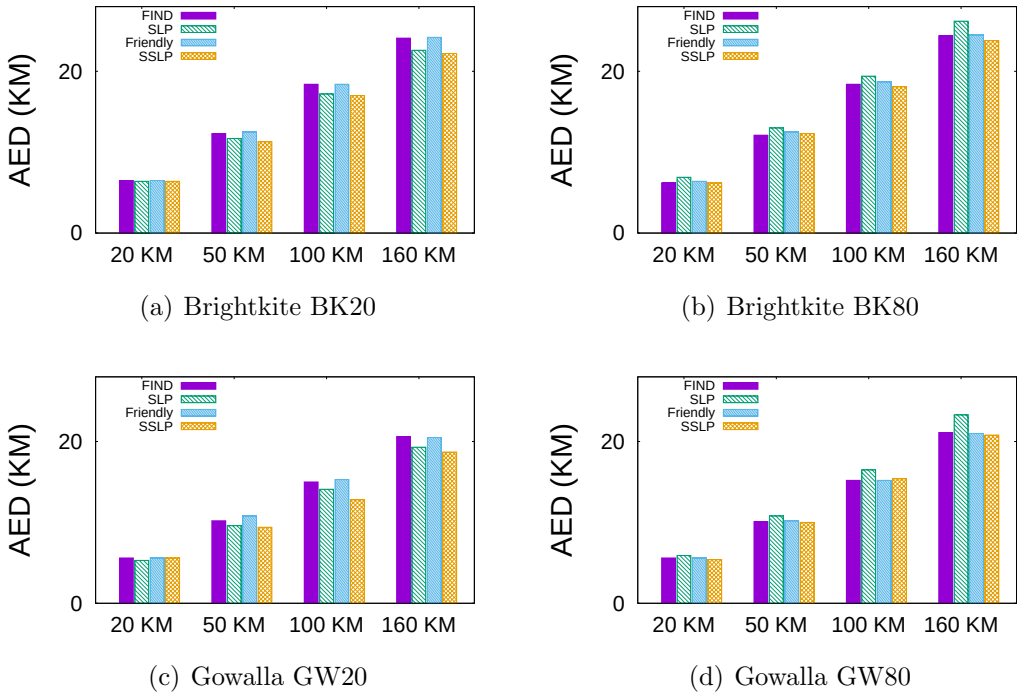
Running Time. The total execution time consumed by the algorithms are measured in hours.

4.5.1 Effectiveness

We notice that in both the datasets, e.g., Brightkite and Gowalla, our proposed algorithm performs better when the number of outer iteration is set as 4. Additional iterations do not significantly increase the user inference coverage in the datasets. Meanwhile, extensive iterations may shift the locations, estimated with lower error distance, far away if incorrectly predicted locations are included in each iteration [63]. Figure 4.4 compares the Average Error Distance (AED) of SSLP with other three state-of-the-art models FIND [9], SLP [74], and Friendly [101]. In this experiment, we set the error distances as 20KM, 50KM, 100KM, 160KM.

We also compare the Average Error Distance (AED) of our proposed algorithm with FIND, SLP, Friendly methods. We found that SSLP has always lower AED in both the datasets. For example, in Brightkite BK80 data setting, the AED results 23.8KM when error distance is considered 160KM. In the same data setting, the remaining models FIND [9], SLP [74], Friendly [101] report AED as 24.4KM, 26.2KM, 24.5KM, respectively. Comparing with Brightkite, the Gowalla dataset always produces lower AED in each algorithm. For example, in Gowalla dataset, SSLP reports 18.7KM and 20.8KM in AED within 160KM of error distance

in GW20 and GW80 data settings, respectively. On the other hand, SSLP reports 22.2KM and 24.6KM in AED within 160KM of error distance in BK20 and BK80 data settings of Brightkite dataset, respectively.



(a) Brightkite BK20

(b) Brightkite BK80

(c) Gowalla GW20

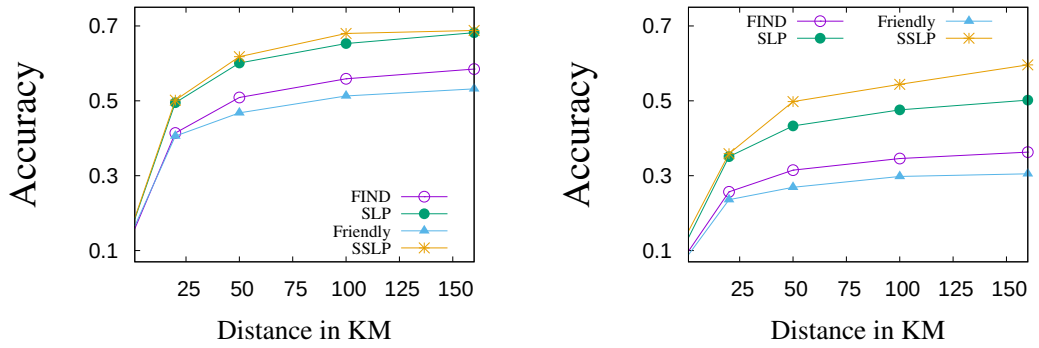
(d) Gowalla GW80

Figure 4.4: Average Error Distance within 20KM, 50KM, 100KM, 160KM

In Figure 4.5, we compare the accuracy of the models using both the datasets using different error distances settings ranging from 20KM to 160KM. The SLP and SSLP models have similar performances in terms of accuracy in both the datasets with lower location sparsity (e.g., in BK20 and GW20 data settings). However, in sparse data setting in BK80 and GW80, the SSLP approach has noticeable higher accuracy than SLP. For example, in BK80, SLP reports 50.2%, where SSLP reports 9.4% better accuracy within 160KM of error distance in BK80 data settings. In each dataset, the accuracy of the SLP, FIND, and SSLP follow a similar trend. In both the datasets, Friendly model has lower accuracy comparing with the remaining models. It reports 27.4% accuracy in Gowalla GW80 dataset within 160KM of error distance. This is because, the Friendly model trains a decision tree using several social factors [63], e.g., following relationship, conversation between users, etc. Due to unavailability of rich information of social features in LBSN dataset, the Friendly model can not build a better classifier to distinguish nearby users.

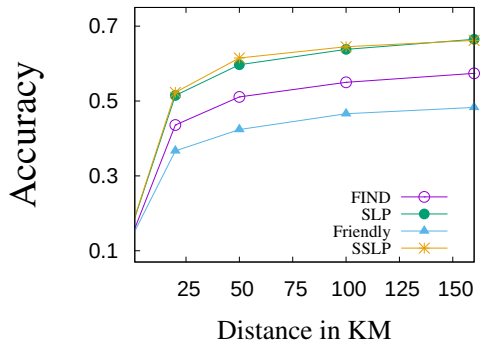
4.5.2 User Inference Coverage

In this section, we report the inference coverage of the models in both the datasets. This metric is useful to identify the models which can assign a location to the maximum number of unlabeled

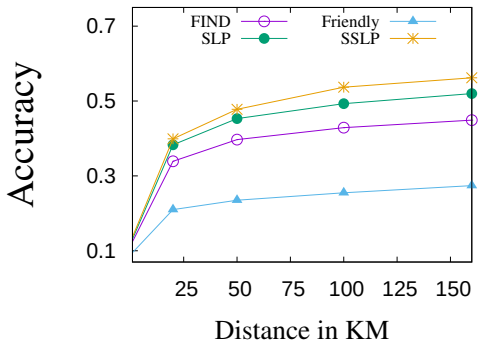


(a) Brightkite BK20

(b) Brightkite BK80



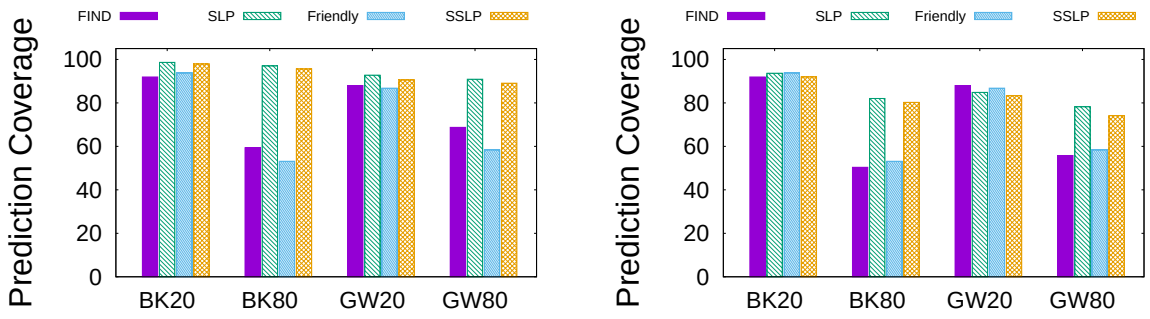
(c) Gowalla GW20



(d) Gowalla GW80

Figure 4.5: Accuracy comparison

users. We compare the inference coverage of the models using default setting (e.g., the number of iterations are set as 4) and Local setting (only one iteration). The percentage of the inference coverage using default number of iterations is shown in Figure 4.6(a). In both the datasets, SLP and SSLP models have similar inference coverage using default setting. The FIND and Friendly models follow similar trends in inference coverage in both the datasets. For example, the inference coverage is about 6-10% higher in FIND comparing to Friendly model in both the datasets with sparse labeled users (e.g., in BK80 and GW80 data settings). Figure 4.6(b) compares the coverage in local settings where the models are allowed to iterate only once. The local inference coverage is always lower than the coverage obtained from default settings. This is because, in local inference, the inferred location of the users are not allowed to propagate further in the network. We also notice that the prediction coverage of SSLP is little lower than SLP in local inference setting. This is because, the neighbor validation steps mentioned in Algorithm 2 (line 2-6) remove the irrelevant neighbor information for some users. Therefore, such users may not be estimated any location at the first iteration. However, the overall accuracy of SSLP is much higher than SLP (ref. Figure 4.5).



(a) Default Setting

(b) Local Inference

Figure 4.6: User Inference Coverage

4.5.3 Running Time

The time costs of the models have been shown in Figure 4.7. The trends of the running time of the four models are similar in both the datasets, where SLP consumes the maximum time to process the datasets. Meanwhile, the FIND model consumes the minimum time to process the datasets. This is because, the FIND model is based on likelihood estimation based on the friendship co-efficients that are already pre-computed in each dataset. In Brightkite BK80, the running time of FIND model is about 1 hour where in SLP it takes more than 3 hours for the location inference task. The SSLP model has always lower running time than SLP. This is because, SSLP ignores few neighbors corresponding to some users that do not have social closeness score more that the defined threshold (e.g., τ). Comparing Brightkite and Gowalla

datasets, the running time in Gowalla is always higher. This is because the number of user nodes and average user degree are higher in Gowalla dataset.

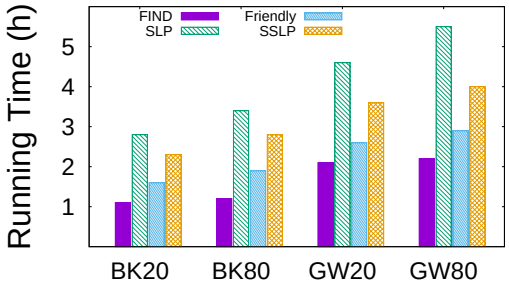


Figure 4.7: Running Time of the models in Gowalla and Brightkite datasets

4.6 Summary

In this chapter, we have presented a label propagation based algorithm that leverages the geographic distribution of the network to estimate their activity location. The social relationships in location-based social media platform provide significant evidence of users' locations. We first define a proper inference sequence that allows us to infer locations of the users who have a higher chance to be estimated locations with lower error distance. We ignore those neighbors in propagating their location label if they are not socially closer. Such practices help us to infer the user location with higher accuracy and the run-time also improved by two-folds. The problem of estimating the activity location of each social user in location-based social media will allow us to create more clear picture to help support meaningful spatial communities, location recommendation, event planning.

Chapter 5

Top-k Socio-spatial co-engaged Location Selection in Social Network

In the previous chapters, we have carried out an in-depth analysis of various location prediction models using a similar set of metrics and various datasets in a unified framework. We also propose a location inference model based on label propagation approach. The proposed model is guided by an inference sequence that allows us to infer the locations of the users first who have higher chance to be estimated a location with a lower error distance. In this chapter, we will exploit the social and spatial properties of the locations associated with social users and their connections.

With the advent of location-based social networks, users can tag their daily activities in different locations through check-ins. These check-in locations signify user preferences for various socio-spatial activities and can be used to build their profiles to improve the quality of services in some applications such as recommendation systems, advertising, and group formation. To support such applications, in this chapter, we formulate a new problem of *identifying top-k Socio-Spatial co-engaged Location Selection* (*SSLS*) for users in a social graph, that selects the best set of k locations from a large number of location candidates relating to the user and her friends. The selected locations should be (i) *spatially and socially relevant* to the user and her friends, and (ii) *diversified in both spatially and socially* to maximize the coverage of friends in the spatial space. This problem has been proved as NP-hard.

To address the challenging problem, we first develop a branch-and-bound based `Exact` solution by designing some pruning strategies based on the derived bounds on diversity. To make the solution scalable for large datasets, we also develop an approximate solution by deriving the relaxed bounds and advanced termination rules to filter out insignificant intermediate results. To further accelerate the efficiency, we present one fast exact approach and a meta-heuristic approximate approach by avoiding the repeated computation of diversity at the running time. Finally, we have performed the extensive experiments to evaluate the performance of our proposed models and algorithms against the adapted existing methods using four real-world

large datasets.

Chapter Map. In Section 5.1, we provide an overall introduction to the problem of identifying *top-k Socio-Spatial co-engaged Location Selection*. Section 5.2 formally defines the top-k *SSLS* problem. The `Exact` approach of top-k *SSLS* query is presented in Section 5.3. In Section 5.4, an approximate approach is discussed to speed up the `Exact` approach. Further, we develop a fast exact solution a.k.a. `Exact`$^+$, and a fast approximate solution to accelerate the efficiency in Section 5.5. Finally, we report the experimental results and a case study in Section 5.6, and conclude the chapter in Section 5.7.

5.1 Introduction

Nowadays, most social network platforms have enabled the location features for users, with which their activities would be tagged at different check-in locations. Therefore, the Location-based Social Networks (LBSN) that capture both the social and spatial information, are becoming popular. Moreover, even the conventional social network platforms, such as Facebook, have enabled the location check-in features to allow the social users to tag their daily activities at different places. Such location information along with social factors can be used to improve the quality of services in many applications such as recommendation systems, marketing, advertising, and group formation [7, 79, 105, 163]. However, given a user, the number of candidate locations might be quite large, and different locations may represent different aspects of the user's behavior under the consideration of various socio-spatial factors. Such factors can be, for example, the popularity of locations among the user's social friends, or how close geographically are the places of their interests; in essence, locations that are popular or close to the popular ones of the friends would have higher relevance.

The social and spatial factors have a strong correlation in LBSNs where the check-in locations are established through social activities and spatial influences [164]. Given a user and a large number of her visited historical locations, it should model both the factors in a meaningful way in order to discover a small set of locations that can engage the user and her friends. More specifically, the social factors are significant to distinguish the preferences of locations to a friend. Similarly, the spatial factors can influence the user and her friends' interest in different spatial proximity. At the same time, we want to avoid bias, and if possible spread these locations across the possibilities - allows for more diverse recommendations. For example, a user may give high preference to a location where most of her friends have visited the place, than the other locations with fewer visits by her friends. Therefore, this work is to exploit both the social and spatial characteristics of relationships among the social network users and their locations to better support location dependent applications. To that end, in this chapter we propose the problem of *identifying top-k Socio-Spatial co-engaged Location Selection for users*, denoted as *SSLS*. A co-engaged location can be easily accessible by a user and her selected friends that can

be covered by the location. More specifically, given a user, *SSLS* will return a set of selected locations that satisfy the following two conditions:

i. (***relevance***:) The selected locations should be both *spatially and socially relevant* to the user and her social friends. Two users are called social friends to each other if they have a social connection, e.g., one user is following the other.

ii. (***diversity***:) The selected locations should also be *diversified both spatially and socially* in order to maximize the spatial and social coverage of the user's social friends.

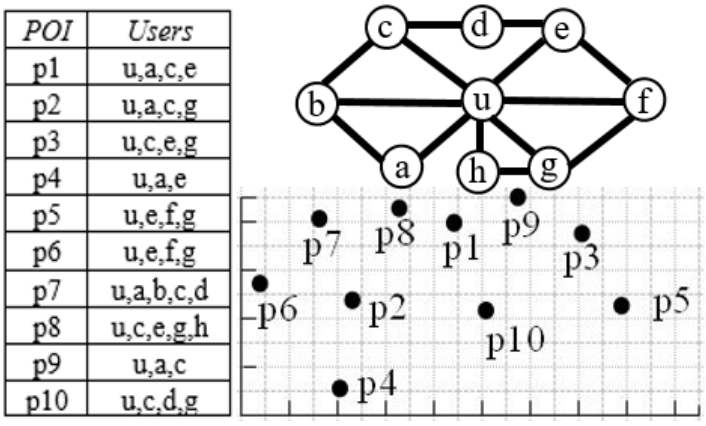


Figure 5.1: An example of the *SSLS* query

Applications. *SSLS* has a wide range of applications. Here, we discuss one application to explain the *SSLS* problem better:

Event Scheduling. Let us consider a toy example of an event scheduling application in Figure 5.1. There are ten points-of-interests (POIs) $\{p_1, p_2, ..., p_{10}\}$ and a set of users who checked-in the places are given. In the example, an enterprise social network user u wants to schedule a series of social events in multiple locations (say, in two locations), which will be preferable and convenient for both the user and her linked customers (e.g., friends). More specifically, the user u wants to select the two locations such that they are (i) related: locations are the user's favorite ones where she visited earlier; (ii) socially and spatially relevant: locations where many of the user's friends also visited these places or some nearby places; (iii) spatial diversified: the selected locations are spatially distant, e.g., in different cities; and (iv) social diversified: each selected location should cover a set of friends such that the selected locations together can cover a maximum number of friends, and any two selected locations have a minimum overlap of friends to be covered.

In this application scenario, the *SSLS* model will return p_5 and p_7 as the best two locations for scheduling events for u. This is because, (i) p_5 and p_7 are previously checked-in by u; (ii) all the friends (e.g., customers) of u except h, have checked-in p_5 and p_7 (socially relevant); (iii) although the friend h of u did not have exact check-ins at either of the two locations, she has

one check-in location p_8 near to p_7 (spatially relevant); (iv) u's friends who checked-in p_5, p_7 are disjoint, i.e., $\{e, f, g\}$ with p_5, $\{a, b, c\}$ with p_7 (socially diverse); and (v) p_5 and p_7 are itself spatially distant.

We have proved that the *SSLS* problem is NP-hard. To solve this problem, one may consider to directly use the existing greedy solutions on top-k diversified spatial object selection, such as DisC [37] and SOS [61]. However, there exist some stringent gaps that make them inapplicable, including (i) Both DisC and SOS define diversity based on spatial distance only. Thus, they do not account for the important aspect of diversity in geo-social networks, which we refer to as *social diversity*. We argue that both the spatial and social aspects need to be considered for selecting diversified objects in a geo-social network to get the best *SSLS* set. (ii) Both the approaches depend on a user-defined distance threshold to select k diversified objects. Nevertheless, it is hard for an end-user to define the best distance thresholds for different k without knowing the underlying data distribution. Also, their selection processes cannot be personalized towards individual users with particular preferences.

The main contributions of this work are as below:

• **SSLS Formulation.** We formally define the problem top-k Socio-Spatial co-engaged Location Selection problem. We provide detailed algorithms and metrics for using social and spatial relevance, and diversity in order to maximize the spatial and social coverage of the search space.

• **Exact and Approximate approaches.** We first propose an `Exact` approach by developing some pruning strategies based on the *derived lower bounds on the diversity* of an already explored feasible set. Such an approach avoids the exploration of a large number of locations that are irrelevant to users and their social connections. We also devise an efficient exact method (`Exact`$^+$), a variation that derives bounds based on the *relevance* of candidate locations, and hence avoids repetitive complex diversity computation of groups of locations as in the `Exact` approach. In addition, we present an approximate approach, in which we derive relaxed bounds and propose advanced termination criteria based on the score of the best feasible set, and the diversity of remaining locations. We also introduce a greedy-based Fast Approximate approach that uses the bounds of `Exact`$^+$ and greedily selects the best locations.

• **Extensive Experimental Evaluation.** Finally, we have conducted extensive experiments to evaluate the effectiveness and efficiency of our proposed approaches using four real-world datasets. We have compared the proposed algorithms with two adaptive greedy-based approaches namely, *GMC* [149], and *GNE* [149] that consider relevance and diversity. We also have compared our approaches with an adapted version of Spatial Object Selection (SOS) [61]. Our experimental results show that `Exact`$^+$ outperforms `Exact` and the Approximate approach by 3 to 6, and 2 to 3 times in default data settings, respectively. Moreover, we show that our approaches result in better social and spatial coverage, and diversity of the selected location set for a user when compared with the adapted algorithms.

Table 5.1: Basic Notations

Symbols	Descriptions
$R_{ss}\{D_{ss}\}$	Socio-spatial Relevance {Diversity} Score
$L_u\{V_u\}$	Set of check-in locations {social friends} of u
α	Trade-off between spatial and social importance
ω	Trade-off between relevance and diversity

5.2 Problem Formulation

Let $G(V, E', L, E'')$ be a socio-spatial graph, where V is the set of users, E' is the set $\{(u, v)|u, v \in V\}$ of edges representing the social connections among users, L is the set of locations associated with users, E'' is the set $\{(u, l)|u \in V, l \in L\}$ of edges representing the spatial connections between users and locations. Let $V_u \in V$ and $L_u \in L$ be the social friends and check-ins of user u, respectively. The goal of socio-spatial location selection (*SSLS*) query is to find the best k *socio-spatial relevant and diversified* locations from L_u for user u where $|L_u| >> k$. We have modeled the *SSLS* problem as bi-criteria optimization problem that considers relevance, diversity and their socio-spatial aspects as a whole. In Section 5.1, we have demonstrated the *SSLS* problem using the Event Scheduling application. In this case, the *SSLS* query will select the locations p_5 and p_7 (refer Figure 5.1) as the top two *socio-spatial relevant and diversified* locations from L_u to schedule events for user u. Table 5.1 presents the list of notations used in this chapter.

In this section, we first discuss the intuition and metrics of socio-spatial relevance and socio-spatial diversity, and then formalize the top-k *SSLS* problem with the proof of NP-hardness.

5.2.1 Socio-spatial Relevance

The study in [151] showed that social interest is the type of check-in incentive that stimulates interactions or influences among the friends. Therefore, a location may have higher social importance to a user if a large number of her friends have checked-in the location. Hence, a location $l_i \in L_u$ is socially important to user u that are already known (e.g. previously checked-in) to her social friends as they are easy to communicate in participating an event. Based on this intuition, we define the social relevance score of a location $l_i \in L_u$ w.r.t. a user u.

Definition 7 (Social Relevance)**.** *A check-in location $l_i \in L_u$ of user u is socially relevant to u if her friends also have check-ins at l_i. Formally, the social relevance score S_{sc} of location l_i can be computed as follows.*

$$S_{sc}(l_i) = \frac{|v \in V_u : e(v, l_i) \in E''|}{|V_u|}$$

To define the spatial relevance score, the study of [164] has revealed that the geographical proximities of POIs have a significant influence on social users' check-in behavior. In addition,

[163] also remarked that friends with nearby check-ins would have higher probability for them to share common locations. This is because it is easy for friends to participate in activities at the same locations. Therefore, if some of the friends of user u do not have exact check-in at location l_i, but visit nearby POIs, the user can influence the friends to visit nearby check-ins [163, 164].

Definition 8 (Spatial Relevance)**.** *Given a location $l_i \in L_u$ checked-in by user u, the spatial relevance score, S_{sp} of l_i is computed as follows.*

$$S_{sp}(l_i) = 1 - \frac{\sum_{v \in V_u} \min dist(l_i, L_v)}{d_m * |V_u|}$$

Here, $\min dist(l_i, L_v)$ is the minimum Euclidian distance between the location l_i and the check-in location set L_v of the friend v. d_m is a constant to adjust the spatial relevance score within the range $(0, 1]$. d_m can be assigned as the maximum value among the minimum distances between L_v and l_i for $v \in V_u$.

Based on the above definitions, the socio-spatial relevance score of a location for a user w.r.t. her friends can be measured by a linear combination of the location's social and spatial relevancy.

Definition 9 (Socio-spatial Relevance)**.** *The socio-spatial relevance score R_{ss} of location $l_i \in L_u$ is defined as the weighted sum of the social (S_{sc}) and spatial (S_{sp}) relevance scores.*

$$R_{ss}(l_i) = \alpha \cdot S_{sc}(l_i) + (1 - \alpha) \cdot S_{sp}(l_i) \tag{5.1}$$

Here, $\alpha \in [0, 1]$ is a parameter to specify the relative importance of social and spatial factors based on the applications and $R_{ss}(l_i) \in [0, 1]$. As such, a set S of locations can have its total socio-spatial score as $R_{ss}(S) = \sum_{l \in S} R_{ss}(l)$.

5.2.2 Socio-spatial diversity

Intuitively, diversity requires measuring the dissimilarity (or distances) among the objects in a set. In the spatial domain, a spatially diverse location pair should reside far from each other [37, 139]. Similar to the spatial distance function [139], we calculate the spatial diversity as below,

Definition 10 (Spatial Diversity)**.** *The spatial diversity $D_{sp}(l_i, l_j)$ w.r.t. a user u between a pair of locations $l_i, l_j \in L_u$ is the normalized Euclidean distance between l_i and l_j,*

$$D_{sp}(l_i, l_j) = \frac{dist(l_i, l_j)}{maxD}$$

Here, $maxD$ is used to normalize the spatial diversity score in [0,1]. The value of $maxD$ can be assigned as the maximum distance of all pairs of locations in L_u.

Similarly, the social diversity between a pair of locations w.r.t. a user depends on the social relationships of her friends who visited the locations [145]. Two locations will be considered to be socially diverse w.r.t. a user if the locations have been visited by the distinct sets of the user's friends. Therefore, as defined in [139], we calculate the social diversity using Jaccard distance as below.

Definition 11 (Social Diversity)**.** *The social diversity $D_{sc}(l_i, l_j)$ w.r.t. a user u between a pair of locations $l_i, l_j \in L_u$ is measured as,*

$$D_{sc}(l_i, l_j) = 1 - \frac{|V_u(l_i) \cap V_u(l_j)|}{|V_u(l_i) \cup V_u(l_j)|}$$

where $V_u(l_i)$ and $V_u(l_j)$ are the set of u's friends who checked-ins at l_i and l_j, respectively.

Similar as [139], we define socio-spatial diversity of a location pair as the weighted sum of $D_{sc}(l_i, l_j)$ and $D_{sp}(l_i, l_j)$ below:

Definition 12 (Socio-spatial Diversity)**.** *The Socio-spatial Diversity $D_{ss}(l_i, l_j)$ of a pair of locations l_i, l_j w.r.t. a user u is calculated by the linear combination of spatial and social diversity scores.*

$$D_{ss}(l_i, l_j) = \alpha \cdot D_{sc}(l_i, l_j) + (1 - \alpha) \cdot D_{sp}(l_i, l_j) \tag{5.2}$$

The parameter $\alpha \in [0, 1]$ specifies the relative importance of social and spatial costs. As such, given a location set L', the socio-spatial diversity score of each location $l \in L'$ w.r.t. L' is calculated as $D_{ss}(l, L') = \min\{D_{ss}(l, l_i) | l_i \in L' \setminus l\} = D_{ss}(l, \bar{l})$ s.t., $\bar{l} = \arg\min_{l_i \in L' \setminus l} D_{ss}(l, l_i)$. We further calculate the socio-spatial diversity of a set L' as $D_{ss}(L') = \sum_{l \in L'} D_{ss}(l, L' \setminus l) = \sum_{l \in L'} D_{ss}(l, \bar{l}) = D_{ss}(l_i, l_j) = \alpha \cdot \sum_{l \in L'} D_{sc}(l, \bar{l}) + (1 - \alpha) \cdot \sum_{l \in L'} D_{sp}(l, \bar{l})$, where, $\bar{l} = \arg\min_{l_i \in L' \setminus l} D_{ss}(l, l_i)$ is the location among the set L' that has minimum diversity with l.

We follow the existing works [7, 139] to derive a ranking function as the weighted linear combination of socio-spatial relevance and diversity.

Definition 13 (Socio-spatial Score of a Location Set)**.** *Given a location set $S \subseteq L_u$ w.r.t. a user u, the socio-spatial score $F(S)$ of S w.r.t. u is measured as the weighted linear combination of the socio-spatial relevance and its socio-spatial diversity of S w.r.t. u.*

$$F(S) = \omega \cdot R_{ss}(S) + (1 - \omega) \cdot D_{ss}(S)$$

Here, $\omega \in (0, 1)$ is the trade-off parameter between relevance and diversity and can be adjusted accordingly as per application requirement. When the value of ω is $\omega > 0.5$, the socio-spatial relevance of the set S w.r.t. the user and her friends is more important than the socio-spatial diversity (D_{ss}).

Problem Statement of Top-k *SSLS* Query. Given a social graph G, a positive integer k, a query user u with check-in locations L_u, trade-off parameters ω, α, β, and the socio-spatial score function F, the top-k *SSLS* query returns a set L' of k locations from L_u, s.t., $\forall L^* \subseteq L_u, F(L') > F(L^*)$, where $|L^*| = k$ and $L' \neq L^*$.

Significance of α, ω in *SSLS* Query. The trade-off parameters have significant importance in the quality of the selection considering both the social and spatial aspects. For example, if an application prefers social factors including social relevance and social diversity, then we can set $\alpha = 1$ and ω as default (e.g., $\omega = 0.5$). Thus, it means that the selected locations should be checked-in by a diverse set of friends, and the locations are socially relevant to the user and her friends. Given such setting, $S = \{p_6, p_7\}$ will be selected as the answer for the example in Figure 5.1. If we increase ω to 0.6, i.e., the socio-spatial relevance is preferred, then the *SSLS* query will return $S = \{p_7, p_8\}$ as the answer. Similarly, an end-user can tailor the result by varying different values for α and ω.

Theorem 1. *The top-k SSLS problem is NP-hard.*

Proof: We consider a special case of the problem, assuming the socio-spatial relevance score of each location of user u is 1, i.e., $\forall l \in L_u$, $R_{ss}(l) = 1$, and each location pair is connected with an edge where the edge-weight is represented by socio-spatial distances. We remove the edges between the location pairs where social diversity is 0, and present the location set L_u as vertices of a graph G. Based on this setting, our top-k *SSLS* problem can be transformed into the problem of top-k diverse vertices search in a graph. Additionally, we know that finding top-k diverse set of vertices from G is equivalent to finding maximum weight independent set (MWIS) of size k [120]. Further, in [51], the problem of MWIS has been proved as NP-hard. Hence, we can conclude the proof.

5.3 An Exact Approach

A top-k *SSLS* query searches for the best set of k locations which have the highest socio-spatial score. A straightforward approach to find the best k diverse locations from a large number of candidate locations requires enumerating all possible combinations of k groups of locations [149]. Such an approach is intractable even for a moderate number of candidate locations. Thus, the main challenge is to devise an effective strategy to prune the unnecessary locations that cannot be part of the answer set. The diversity of a group of locations depends on all the participating members (e.g. locations) of the group, and adding a new location to the existing group invalidates the previous diversity score, where total diversity of the group may decrease due to inclusion of a new location. This adds more challenges in developing pruning strategies while exploring the data space for the answer.

For an exact solution to answer the Top-k *SSLS* query, we resort to an incremental Branch-and-Bound (BnB) strategy that progressively adds locations to build the answer set. The key

idea is to develop pruning strategies based on the derived lower bound on socio-spatial diversity of an intermediate set that can avoid exploring a large number of location sets.

5.3.1 Computing Bounds on Diversity of an Intermediate Set

In this subsection, we use the concept of score gain to decide whether a location should be included in an intermediate result set S_I in the process of finding top-k *SSLS* set. Initially, S_I is initialized as empty and $|S_I| < k$ holds always. We use S_R to denote the set of remaining locations to be explored. To make the exact approach efficient, we derive some bounds that help us to avoid exploring a large portion of the data space and terminate the algorithm early. Therefore, if we add a location $l' \in S_R$ to S_I, the socio-spatial score $F(S'_I)$ of the new set $S'_I = \{S_I \cup l'\}$ becomes,

$$F(S'_I) = \omega \cdot R_{ss}(S'_I) + (1-\omega) \cdot D_{ss}(S'_I)$$

Consequently the socio-spatial score gain δ of S'_I w.r.t. the previous set S_I can be computed as follows.

$$\begin{aligned}
\delta &= F(S'_I) - F(S_I) \\
&= \omega \cdot R_{ss}(S'_I) + (1-\omega) \cdot D_{ss}(S'_I) - \omega \cdot R_{ss}(S_I) - (1-\omega) \cdot D_{ss}(S_I) \\
&= \omega \cdot \big(R_{ss}(S'_I) - R_{ss}(S_I)\big) + (1-\omega) \cdot \big(D_{ss}(S'_I) - D_{ss}(S_I)\big) \\
&= \omega \cdot \delta_r + (1-\omega) \cdot \delta_d
\end{aligned} \tag{5.3}$$

Here, we consider δ_r and δ_d as the Relevance Gain and Diversity Gain of S'_I w.r.t. the previous set S_I, respectively:

Relevance Gain (δ_r). The relevance gain can be simplified as: $\delta_r = R_{ss}(S'_I) - R_{ss}(S_I) = R_{ss}(S_I \cup l') - R_{ss}(S_I) = R_{ss}(l')$. $\delta_r \in [0,1]$ can not be negative for any $l' \in L_u$.

Diversity Gain (δ_d). The diversity gain $\delta_d = D_{ss}(S'_I) - D_{ss}(S_I)$ is the difference in socio-spatial diversity of $S'_I = S_I \cup l'$ to S_I. δ_d can be negative when $D_{ss}(S'_I) < D_{ss}(S_I)$.

Here, the value of $D_{ss}(S'_I)$ is dependent on the diversity of the added location $l' \in S_R$ w.r.t. S_I, and the updated aggregated diversity of the locations of S_I, such as,

$$\begin{aligned}
D_{ss}(S'_I) &= D_{ss}(l', S_I) + \sum_{l \in S_I} \min\{D_{ss}(l, S_I \setminus l), D_{ss}(l, l')\} \\
&= \widehat{d} + \widehat{D},
\end{aligned} \tag{5.4}$$

where the first part, $\widehat{d} = D_{ss}(l', S_I) = \min_{l \in S_I}\{D_{ss}(l', l)\}$, is the diversity of the newly added location l' w.r.t. the intermediate set S_I, and the remaining part $\widehat{D} = \widehat{D}(S_I, l') = \sum_{l \in S_I} \min\{D_{ss}(l, S_I \setminus l), D_{ss}(l, l')\}$ is the updated total diversity of the existing set S_I, when $l' \in S_R$ is added to S_I. Using Equation 5.4, we derive the diversity gain δ_d as,

$$\delta_d = D_{ss}(S'_I) - D_{ss}(S_I) = \widehat{d} + \widehat{D} - D_{ss}(S_I) \tag{5.5}$$

Further, we obtain the socio-spatial score gain of the intermediate set S_I using Equation 5.3 and Equation 5.5 as follows,

$$\delta = \omega \cdot \delta_r + (1-\omega) \cdot (\widehat{d} + \widehat{D} - D_{ss}(S_I)) \tag{5.6}$$

Now, we will identify the *eligible* locations from S_R that can generate a positive socio-spatial score gain w.r.t. S_I.

Definition 14 (Eligible Location)**.** *Given a current intermediate set S_I, and a location $l' \in S_R$, l' will be considered as an eligible location if $\delta > 0$.*

Next, we will define some lemmas using the socio-spatial diversity of a set of locations to deduce a lower bound on $\widehat{D}$.

Lemma 1. *Given an intermediate set S_I, an eligible location $l' \in S_R$ w.r.t. S_I, the updated aggregated socio-spatial diversity $\widehat{D}$ of set S_I w.r.t. the eligible location l' will never exceed the total socio-spatial diversity $D_{ss}(S_I)$ of the intermediate set S_I, e.g., $\widehat{D} \leq D_{ss}(S_I)$ is always true.*

Proof: We know, $\sum_{l \in S_I} \min\{D_{ss}(l, S_I \setminus l), D_{ss}(l, l')\} \leq \sum_{l \in S_I} D_{ss}(l, S_I \setminus l)$ is true, as for any $l \in S_I$, $\min\{D_{ss}(l, S_I \setminus l), D_{ss}(l, l')\}$ is no larger than $D_{ss}(l, S_I \setminus l)$. Therefore, $\widehat{D} \leq D_{ss}(S_I)$ holds, as $\widehat{D} = \sum_{l \in S_I} \min\{D_{ss}(l, S_I \setminus l), D_{ss}(l, l')\}$, and $D_{ss}(S_I) = \sum_{l \in S_I} D_{ss}(l, S_I \setminus l)$ (refer Section 5.2.2).

An *eligible* location l' may produce a negative gain in diversity δ_d that can lessen the socio-spatial score of an updated set $S'_I = S_I \cup l'$ comparing with S_I. The below lemma derives the condition when instead of having a negative diversity gain, the socio-spatial score of an intermediate set can generate a positive socio-spatial gain (e.g., $\delta > 0$) for S'_I.

Lemma 2. *Given an intermediate set S_I, an eligible location $l' \in S_R$, s.t., $S'_I = S_I \cup l'$; if S'_I has a negative gain in diversity but $\delta_r > \frac{(1-\omega)}{\omega} \cdot |\delta_d|$ w.r.t. S_I, then the socio-spatial score of S'_I will be larger than that of S_I, e.g., $F(S'_I) > F(S_I)$.*

Proof: Let, $F(S'_I)$ and $F(S_I)$ be the socio-spatial scores of $S'_I = S_I \cup l'$ and S_I respectively. Hence, the socio-spatial gain of S'_I is $\delta = F(S'_I) - F(S_I)$. If $\delta_d < 0$, but $\delta_r > \frac{(1-\omega)}{\omega} \cdot |\delta_d|$, then $\delta = \omega \cdot \delta_r + (1-\omega) \cdot \delta_d > 0$ is always true. Therefore, $\delta = F(S'_I) - F(S_I) > 0$. Hence, $F(S'_I) > F(S_I)$.

Now, we will derive a lower bound on the updated diversity ($\widehat{D}$) of an intermediate set S_I, which will help us to discard a large number of locations from S_R that can not generate a better solution w.r.t. the current intermediate set S_I.

Lower Bound of $\widehat{D}$

Maximum relevance gain of an intermediate set S_I w.r.t. S_R is $\delta_r^{\uparrow} = \max_{l' \in S_R} R_{ss}(l')$. Similarly, the maximum possible diversity of locations in S_R w.r.t. S_I can be calculated as $\widehat{d}_{max} = \max_{l' \in S_R} D_{ss}(l', S_I)$.

An *eligible* location $l' \in S_R$ always derives positive gain, e.g., $\delta > 0$, to an intermediate set S_I (Definition 14). Thus, we get, $\delta = \omega \cdot \delta_r + (1-\omega) \cdot \big(\widehat{d} + \widehat{D} - D_{ss}(S_I)\big) > 0$. Therefore, $\widehat{D} > D_{ss}(S_I) - \widehat{d} - \frac{\omega}{1-\omega} \cdot \delta_r$. Now, we derive lower bound of $\widehat{D}$ by replacing $\widehat{d}$ and δ_r with their maximum possible values,

$$\widehat{D}{\downarrow} = D_{ss}(S_I) - \widehat{d}_{max} - \frac{\omega}{1-\omega} \cdot \delta_r^{\uparrow} \tag{5.7}$$

Early Pruning based on $\widehat{D}$

Using Equation 5.7, we derive that a location $l' \in S_R$ cannot be included into an intermediate set S_I, if $\widehat{D} \leq \widehat{D}{\downarrow}$ is true (refer Equation 5.7). We formalize this pruning condition in Property 1 assuming that we are yet to find a feasible solution of size k.

Property 1. *Given an intermediate set S_I, s.t., $|S_I| < k$, we can prune a location $l' \in S_R$ w.r.t. S_I, if $\widehat{D} \leq \widehat{D}{\downarrow}$ satisfies.*

Further, we derive an advanced termination strategy based on the score of already explored best feasible set and the expected contributions of the remaining locations in the overall score.

Advanced Pruning

First, we derive a pruning condition for an intermediate set S_I of size $(k-1)$, then generalize to any sets of size less than k. Let, S_b be the previously identified best feasible set. Also, let $l' \in S_R$ be an arbitrary location with relevance score $R_{ss}(l')$, and $D_{ss}(l', S_I)$ be the diversity of l' w.r.t. S_I. We denote the updated diversity of $S_I' = S_I \cup l'$ as, $\sum_{l \in S_I' \setminus l'} \min\{D_{ss}(l, S_I \setminus l), D_{ss}(l, l')\} = \widehat{D}$. The set S_I' of size k can replace an earlier identified best feasible set S_b, if $F(S_I') = \omega \cdot R_{ss}(S_I \cup l') + (1-\omega) \cdot D_{ss}(S_I \cup l') > F(S_b)$,

$$\begin{aligned} &\Rightarrow \omega \cdot (R_{ss}(S_I) + \delta_r) + (1-\omega) \cdot \big(\widehat{d} + \widehat{D}\big) > F(S_b) \\ &\Rightarrow \widehat{D} > \frac{1}{1-\omega} \cdot \big(F(S_b) - \omega \cdot (R_{ss}(S_I) + \delta_r)\big) - \widehat{d} \end{aligned} \tag{5.8}$$

The lower bound of $\widehat{D}$ for termination (when $|S_I| = (k-1)$) can be obtained by replacing $\widehat{d}$ and δ_r with their corresponding maximum possible values, e.g., $\widehat{d}_{max} = \max_{l' \in S_R} D_{ss}(l', S_I)$ and $\delta_r^{\uparrow} = \max_{l' \in S_R} R_{ss}(l')$ respectively. Therefore, $\widehat{D}{\Downarrow} = \frac{1}{1-\omega} \cdot \big(F(S_b) - \omega \cdot (R_{ss}(S_I) + \delta_r^{\uparrow})\big) - \widehat{d}_{max}$.

Adopting the above procedure, we add an arbitrary subset of locations $S_R' \subseteq S_R$ to the intermediate set S_I, such that (i) $|S_R'| = (k - |S_I|)$, (ii) the socio-spatial score of new set $S' = S_I \cup S_R'$ surpasses $F(S_b)$, e.g., $F(S') > F(S_b)$. Therefore,

$$\omega \cdot (R_{ss}(S_I) + R_{ss}(S_R')) + (1-\omega) \cdot D_{ss}(S_I \cup S_R') > F(S_b) \tag{5.9}$$

Now, we define the below lemma on socio-spatial diversity of a set $S' = S_I \cup S'_R$ of size k, using the diversity scores of the locations $l' \in S'_R$ w.r.t. current intermediate set S_I.

Lemma 3. *Given an intermediate set S_I, a subset $S'_R \subseteq S_R$ of locations, the socio-spatial diversity $S' = S_I \cup S'_R$ satisfies $D_{SS}(S_I \cup S'_R) \leq \widehat{D} + \sum_{l' \in S'_R} D_{ss}(l', S_I)$, where $\widehat{D}$ is the updated diversity of S_I w.r.t. an arbitrary location $l' \in S'_R$.*

Proof: Let us assume the arbitrary location $l' \in S'_R$ is added to the intermediate set S_I. Therefore, we get, $D_{SS}(S_I \cup l') = \widehat{D} + D_{SS}(l', S_I)$. Now, if we add another arbitrary location $l_i \in S'_R \backslash l'$ to the current intermediate set $S_I \cup l'$, we will get $D_{SS}(S_I \cup l' \cup l_i) = \widehat{D_i} + D_{SS}(l_i, S_I \cup l') = \widehat{D} + D_{SS}(l', S_I) + D_{SS}(l_i, S_I \cup l')$. We know, $D_{SS}(l_i, S_I \cup l') \leq D_{SS}(l_i, S_I)$. Therefore recursively adding all the location from S'_R to set S_I will satisfy $D_{SS}(S_I \cup S'_R) \leq \widehat{D} + \sum_{l' \in S'_R} D_{ss}(l', S_I)$.

Applying Lemma 3 in Equation 5.9 can be derived it as,

$$\omega \cdot (R_{ss}(S_I) + R_{ss}(S'_R)) + (1 - \omega) \cdot (\widehat{D} + \sum_{l' \in S'_R} D_{ss}(l', S_I)) > F(S_b)$$

$$\Rightarrow \widehat{D} > \frac{F(S_b) - \omega \cdot \big(R_{ss}(S_I) + R_{ss}(S'_R)\big)}{(1 - \omega)} - \sum_{l' \in S'_R} D_{ss}(l', S_I)$$

Now, we will derive the lower bound $\widehat{D}\Downarrow$ by replacing $R_{ss}(S'_R)$ and $\sum_{l' \in S'_R} D_{ss}(l', S_I)$ with their maximum values,

$$\widehat{D}\Downarrow = \frac{F(S_b) - \omega \cdot \big(R_{ss}(S_I) + R^{Max}_{ss}(S'_R)\big)}{(1 - \omega)} - D^{Max}_{ss} \tag{5.10}$$

Here, $R^{Max}_{ss}(S'_R) = \max_{l' \in S_R}(\sum_{k-|S_I|} R_{ss}(l'))$ is the aggregated top $(k - |S_I|)$ relevance scores among the locations in S_R, and $D^{Max}_{ss} = \max_{l' \in S_R}(\sum_{k-|S_I|} D_{ss}(l', S_I))$ is the sum of the top $(k - |S_I|)$ diversity scores of the locations $l' \in S_R$ w.r.t. S_I. Finally, we formalize the pruning condition in Property 2 when a feasible set has been retrieved already.

Property 2 (Location Pruning)**.** *Let S_I be an intermediate set s.t. $|S_I| < k$, $|S_I| + |S_R| \geq k$, and S_b be the best feasible set of size k that has been identified already. Using Equation 5.10, we can prune location $l' \in S_R$ w.r.t. S_I, if $\widehat{D} \leq \widehat{D}\Downarrow$ satisfies.*

The `Exact` algorithm progressively adds locations, and checks whether the locations can generate a positive gain in the socio-spatial score. Further, it prunes a large number of locations using the lower bound of an intermediate set.

5.3.2 Algorithm

Algorithm 3 summarizes the `Exact` approach for answering the *SSLS* query. It takes socio-spatial graph G, query user u, an integer k as inputs, and returns a set S of k locations that maximizes socio-spatial score $F(S)$. We initialize an intermediate set S_I as empty, and S_R contains the

remaining locations $l \in L_u \setminus S_I$ arranged in descending order of socio-spatial relevance scores R_{ss}. A priority queue, Q, maintains a tuple of intermediate set S_I, set S_R of remaining locations, and socio-spatial score of S_I. An inner loop fetches next location l from S_R (Line 9), and an entity $(S_I - \{l\}, S_R)$ is pushed to Q. If no feasible set is retrieved yet, the process further prunes S_R using Property 1 (Line 13). Otherwise, Property 2 (Line 15) is used to prune. Finally, an entity (S_I, S_R) is pushed into Q when $|S_R| > 0$. The process continues until Q is empty. Finally, the final result set S of size k is returned.

Algorithm 3: SSLS: `Exact`

```
   Input: Socio-spatial graph G, set size k, query user u
   Output: Location set S of size k
1  Initialize: S_I ← ∅, S ← ∅, F(S_b) ← 0, flagFS ← false,
2  Append ⟨l, R_ss(l, u)⟩ into S_R in non-increasing R_ss(l, u)
3  Q.push(S_I, S_R, 0)
4  while Q is not empty do
5      S_I, S_R ← Q.pop()
6      if |S_I| = k or |S_R| = ∅ then
7          continue
8      while |S_I| < k and |S_I| + |S_R| ≥ k do
9          l ← nextLocation(S_R)
10         S_I.append(l); S_R.remove(l)
11         Q.push(S_I − {l}, S_R, F(S_I − {l}))
12         if flagFS == false then
13             S_R ← pruneE(S_I, S_R) *** Property 1
14         else
15             S_R ← pruneT(S_I, S_R, F(S_b)) *** Property 2
16         if |S_R| > 0 then
17             Q.push(S_I, S_R, F(S_I))
18         if |S_I| == k and F(S_I) > F(S_b) then
19             S ← S_I; F(S_b) ← F(S_I)
20             flagFS ← true; break;
```

Time Complexity. Time complexity of `Exact` is $O({}^nC_k)$, as in worst case the `Exact` needs to check all combinations of k from n number of locations. However, in practice, the actual running time is much less as large number of locations can be pruned using the developed pruning strategies.

Steps of `Exact` with an example. We use the example in Figure 5.1 to demonstrate the steps of `Exact` algorithm. First, we will show the steps to compute the socio-spatial relevance score of a location (say, p_6), and socio-spatial diversity of a location pair (say, $\{p_6, p_2\}$) of u using the check-in information available in Figure 5.1.

The user u has seven friends, among them three friends checked-in the location p_6 that results social relevance score $S_{sc}(p_6, u) = \frac{3}{7} = 0.43$. Now, we will calculate the spatial diversity

score $D_{sp}(p_6, p_2)$. The locations l_6 and l_2 are checked-in by u's friends $\{e, f, g\}$ and $\{a, c, g\}$ respectively. Therefore, the spatial diversity score $S_{sp}(p_6, p_2)$ is calculated as, $D_{sp}(p_6, p_2) = 1 - \frac{|\{e,f,g\} \cap \{a,c,g\}|}{|\{e,f,g\} \cup \{a,c,g\}|} = 1 - \frac{1}{5} = 0.80$. The calculated social diversity and social relevance of u's locations are shown in Figure 5.2.

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	S_{sc}
p1	0	0.5	0.5	0.33	0.8	0.8	0.5	0.6	0.33	0.75	0.43
p2	0.5	0	0.5	0.75	0.8	0.8	0.5	0.6	0.33	0.33	0.43
p3	0.5	0.5	0	0.75	0.5	0.5	0.8	0.25	0.75	0.33	0.43
p4	0.33	0.75	0.75	0	0.75	0.75	0.75	0.8	0.67	1	0.29
p5	0.8	0.8	0.5	0.75	0	0	1	0.6	1	0.75	0.43
p6	0.8	0.8	0.5	0.75	0	0	1	0.6	1	0.75	0.43
p7	0.5	0.5	0.8	0.75	1	1	0	0.83	0.33	0.75	0.43
p8	0.6	0.6	0.25	0.8	0.6	0.6	0.83	0	0.8	0.5	0.57
p9	0.33	0.33	0.75	0.67	1	1	0.33	0.8	0	0.67	0.29
p10	0.75	0.33	0.33	1	0.75	0.75	0.75	0.5	0.67	0	0.29

Figure 5.2: Social Diversity and Social Relevance Scores of u's locations (refer Figure 5.1)

Now, to calculate the spatial relevance score of the location of $p_6 \in L_u$ of u, we need to calculate $d_m = \max\{\min_{v \in V_u} dist(l_i, L_v)\}$ as the maximum value among the smallest distances between the location $p_6 \in L_u$ and the location set L_v of each friend $v \in V_u$, e.g., $v = \{a, b, c, e, f, g, h\}$. We will demonstrate first to calculate the value $\min_{v \in V_u} dist(l_i, L_v)$ using the check-in information of one friend $a \in V_u$ as reference. The check-ins of friend $a \in L_u$ are $\{p_1, p_2, p_4, p_7, p_9\}$. Therefore, among the location set $\{p_1, p_2, p_4, p_7, p_9\}$, we get $dist\{p_6, p_7\} = 3.5$ as the minimum spatial distance among the location p_6 and the check-ins by the friend $a \in L_u$ (see Figure 5.1 for the relative distances between the points). Following this, we get $d_m = \max\{\min_{v \in V_u} dist(l_i, L_v)\} = \max\{3.5, 3.5, 3.5, 0, 0, 0, 9.5\} = 9.5$ for the location p_6 where, $V_u = \{a, b, c, e, f, g, h\}$. Therefore, we calculate the spatial relevance score of location p_6 as, $S_{sp}(p_6, u) = 1 - \frac{\sum_{v \in V_u} \min dist(l_i, L_v)}{d_m * |V_u|} = 1 - \frac{(3.5+3.5+3.5+0+0+0+9.5)}{9.5*7} = 0.699$. Now, we will calculate the social diversity between the locations p_6 and p_2 as an example. Among the locations of u, we get $maxD = 15$ as the maximum distance among the location pairs checked-in by user u (see Figure 5.1 where distance between the pair (p_6, p_5) is maximum as $dist(p_6, p_5) = 15$). Therefore, we calculate the spatial diversity between the location pair (p_6, p_2) as $D_{sp}(p_6, p_2) = \frac{4}{15} = 0.27$. The spatial diversity and the spatial relevance scores of the locations are shown in Figure 5.3.

Now, we will calculate the socio-spatial relevance score and socio-spatial diversity of the locations considering equal weight in social and spatial factors, e.g., $\alpha = 0.5$. Therefore, we calculate the socio-spatial relevance score $R_{ss}(p_6, u)$ of p_6 as $R_{ss}(p_6, u) = 0.5 * 0.43 + 0.5 * 0.699 = 0.564$. Similarly, the socio-spatial diversity $D_{ss}(p_6, p_2)$ is calculated as $D_{ss}(p_6, p_2) = (0.5 * 0.80 + 0.5 * 0.27) = 0.53$. The Table in Figure 5.4 shows the socio-spatial relevance scores (R_{ss}) and diversity of u's locations calculated using $\alpha = 0.5$. Note, we only need to pre-compute the socio-spatial relevance R_{ss} of the locations. We consider $\omega = 0.5$ in the *SSLS* query.

Figure 5.5 illustrates the node exploration towards searching for the top-2 *SSLS* locations. Each state (node in tree) is marked with a number denoting the node exploration sequence.

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	S_{sp}
p1	0	0.36	0.33	0.57	0.52	0.56	0.37	0.17	0.18	0.27	0.665
p2	0.36	0	0.69	0.23	0.73	0.27	0.25	0.3	0.53	0.37	0.498
p3	0.33	0.69	0	0.79	0.23	0.87	0.73	0.51	0.22	0.33	0.671
p4	0.57	0.23	0.79	0	0.8	0.36	0.47	0.53	0.73	0.46	0.505
p5	0.52	0.73	0.23	0.8	0	1	0.87	0.65	0.42	0.37	0.526
p6	0.56	0.27	0.87	0.36	1	0	0.23	0.45	0.77	0.63	0.699
p7	0.37	0.25	0.73	0.47	0.87	0.23	0	0.17	0.5	0.52	0.701
p8	0.17	0.3	0.51	0.53	0.65	0.45	0.17	0	0.33	0.43	0.751
p9	0.18	0.53	0.22	0.73	0.42	0.77	0.5	0.33	0	0.34	0.526
p10	0.27	0.37	0.33	0.46	0.37	0.63	0.52	0.43	0.34	0	0.489

Figure 5.3: Spatial Diversity and Spatial Relevance Scores of u's locations (refer Figure 5.1)

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	R_{ss}
p1	0	0.43	0.42	0.45	0.66	0.68	0.43	0.38	0.26	0.51	0.547
p2	0.43	0	0.59	0.49	0.77	0.53	0.38	0.45	0.43	0.35	0.463
p3	0.42	0.59	0	0.77	0.36	0.69	0.77	0.38	0.49	0.33	0.55
p4	0.45	0.49	0.77	0	0.78	0.56	0.61	0.66	0.7	0.73	0.395
p5	0.66	0.77	0.36	0.78	0	0.5	0.93	0.63	0.71	0.56	0.477
p6	0.68	0.53	0.69	0.56	0.5	0	0.62	0.52	0.88	0.69	0.564
p7	0.43	0.38	0.77	0.61	0.93	0.62	0	0.5	0.42	0.64	0.565
p8	0.38	0.45	0.38	0.66	0.63	0.52	0.5	0	0.57	0.46	0.661
p9	0.26	0.43	0.49	0.7	0.71	0.88	0.42	0.57	0	0.51	0.406
p10	0.51	0.35	0.33	0.73	0.56	0.69	0.64	0.46	0.51	0	0.387

Figure 5.4: Socio-spatial Diversity and Socio-spatial Relevance Scores of u's locations (refer Figure 5.1)

A priority queue, Q is initialized with $S_I = \emptyset$ and $S_R = \{p_8, p_7, p_6, p_3, p_1, p_5, p_2, p_9, p_4, p_{10}\}$, where S_R contains u's locations in non-increasing order of relevance scores (R_{ss}). The entries $(\emptyset, \{p_7, p_6, ..., p_4, p_{10}\}, 0)$ (Algorithm 3, Line 11) and $(p_8, \{p_7, p_6, ..., p_4, p_{10}\}, 0.331)$ (Line 17) are pushed to Q for further exploration. Next, $(\{p_8\}, \{p_7, p_6, ..., p_4, p_{10}\}, 0.331)$ is dequeued from Q. We begin exploring from p_8, and S_I becomes $\{p_8, p_7\}$ (step 2). In the meantime, $(\{p_8\}, \{p_6, ..., p_4, p_{10}\}, 0.331)$ is pushed to Q (Line 11), and we get the first feasible solution $S_b = \{p_8, p_7\}$ with $F(S_b) = 0.5 * (0.661 + 0.565) + 0.5 * (0.5 + 0.5) = 1.113$. Continuing the process (till step 10), the best feasible set $S_b = \{p_8, p_5\}$ with $F(S_b) = 1.199$ is obtained in this branch.

Further, we dequeue $(\emptyset, \{p_7, p_6, ..., p_4, p_{10}\}, 0)$ and explore the branch with node p_7 (Step 11). After processing lines 10 and 11 of Algorithm 3, we check pruning condition at Line 15 using Property 2, where $\widehat{D}\Downarrow = \frac{1.199 - 0.5*(0.565+0.564)}{0.5} - 0.93 = 0.339$ is computed using Equation 5.10. As, $\widehat{D} > 0.339$ is true w.r.t. each location in current $S_R = \{p_7, p_6, ..., p_4, p_{10}\}$, we continue exploring the branch with node p_7 and update the best feasible set as $S_b = \{p_7, p_5\}$ with $F(S_b) = 1.451$. In the next iteration while exploring node p_6 (step20), we calculate $\widehat{D}\Downarrow = \frac{1.451 - 0.5*(0.564+0.55)}{0.5} - 0.88 = 0.908$ w.r.t. $S_I = \{p_6\}$ and $S_b = \{p_7, p_5\}$. All the locations in S_R satisfy Property 2, therefore, we terminate processing $S_I = \{p_6\}$. By exploring the remaining branches, we obtain $S = \{p_7, p_5\}$ as the top-2 *SSLS* set for the query user u.

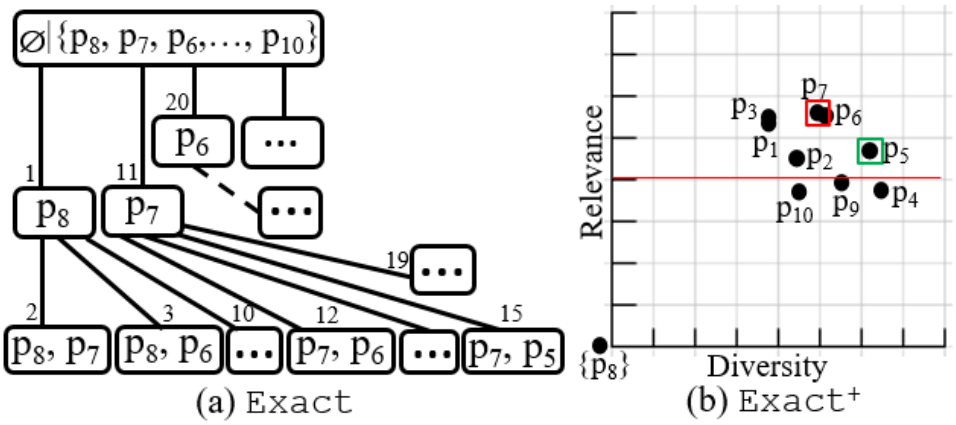


Figure 5.5: Node exploration steps of `Exact` and `Exact`$^+$

5.4 An Approximate Approach

One major limitation of `Exact` approach is the high computational cost, which makes it unrealistic for a large number of candidate locations. To validate the location pruning in `Exact`, it is required to calculate the updated diversity $\widehat{D}$ of the current intermediate set S_I w.r.t. each location $l' \in S_R$. Calculating $\widehat{D}$ for each intermediate set w.r.t. each location in remaining set is expensive when the size of S_R is large. Therefore, to improve pruning and advanced termination, we derive relaxed bounds on diversity. We first define the maximum possible score of $\widehat{D}$ for an intermediate set S_I when an eligible location $l' \in S_R$ is added to S_I. Lemma 1 deduces $\widehat{D} \leq D_{ss}(S_I)$ is true for any intermediate set S_I. Therefore, the maximum possible value of $\widehat{D}$ can be obtained as,

$$\widehat{D}_{max} = max(\widehat{D}) = D_{ss}(S_I) \tag{5.11}$$

Note that $\widehat{D}$ is dependent on S_R, and $\widehat{D}_{max} = D_{ss}(S_I)$ is true w.r.t. $l' \in S_R$ only when $\forall l \in S_I, D_{ss}(l, S_I \setminus l) < D_{ss}(l', l)$ strictly holds. Therefore, to make an efficient approximate approach, we design pruning and termination in the below subsection using the lower bound on $\widehat{d}$. We consider $\widehat{D} = \widehat{D}_{max} = D_{ss}(S_I)$ always true w.r.t. each location $l' \in S_R$.

Computing bounds on diversity of locations. Let us assume, S_I be an intermediate set of size $(k-1)$, $l' \in S_R$ be an *eligible* location, and S_b be the best feasible set identified already. For an arbitrary *eligible* location $l' \in S_R$ that can be added to S_I, we continue to derive

Equation 5.8,

$$\begin{aligned}
&\widehat{D} > \frac{1}{1-\omega} \cdot \big(F(S_b) - \omega \cdot (R_{ss}(S_I) + \delta_r)\big) - \widehat{d} \\
\Rightarrow\ &\widehat{D} > \frac{1}{1-\omega} \cdot \big(F(S_b) - \omega \cdot (R_{ss}(S_I) + \delta_r) - (1-\omega) \cdot D_{ss}(S_I) \\
&\qquad\qquad + (1-\omega) \cdot D_{ss}(S_I)\big) - \widehat{d} \\
\Rightarrow\ &\widehat{D} > \frac{1}{1-\omega} \cdot \big(F(S_b) - (\omega \cdot R_{ss}(S_I) + (1-\omega) \cdot D_{ss}(S_I)) \\
&\qquad\qquad + (1-\omega) \cdot D_{ss}(S_I) - \omega \cdot \delta_r\big) - \widehat{d} \\
\Rightarrow\ &\frac{F(S_b) - F(S_I)}{1-\omega} + D_{ss}(S_I) - \widehat{D} - \frac{\omega}{1-\omega} \cdot \delta_r < \widehat{d}
\end{aligned}$$

We will first derive a relaxed bound for termination using the above equation. Hence, we replace the upper bound of $\widehat{D}$ with its maximum possible value, e.g., $\widehat{D}_{max} = D_{ss}(S_I)$,

$$\begin{aligned}
&\Rightarrow \frac{F(S_b) - F(S_I)}{1-\omega} + D_{ss}(S_I) - \widehat{D}_{max} - \frac{\omega}{1-\omega} \cdot \delta_r < \widehat{d} \\
&\Rightarrow \frac{F(S_b) - F(S_I)}{1-\omega} < \widehat{d} + \frac{\omega}{1-\omega} \cdot \delta_r \text{ (putting } \widehat{D}_{max} = D_{ss}(S_I)) \\
&\Rightarrow F(S_b) < F(S_I) + (1-\omega) \cdot \widehat{d} + \omega \cdot \delta_r
\end{aligned}$$

Now, we will generalize the above condition for any intermediate set S_I of size $|S_I| < k$. So, we need to add an arbitrary subset $S'_R \subseteq S_R$ to S_I such that $|S'_R| = (k - |S_I|)$. Hence,

$$\begin{aligned}
&F(S_b) < F(S_I) + (k - |S_I|) \cdot ((1-\omega).\widehat{d} + \omega \cdot \delta_r) \\
&\Rightarrow \frac{F(S_b) - F(S_I) - \omega \cdot (k - |S_I|) \cdot \delta_r}{(1-\omega) \cdot (k - |S_I|)} < \widehat{d}
\end{aligned}$$

Next, we will derive the lower bound $\widehat{d\downarrow}$ by replacing the expression $(k - |S_I|) \cdot \delta_r$ with the total socio-spatial relevance score of top $(k - |S_I|)$ relevant locations from S_R. We calculate the total socio-spatial relevance score of the top $(k - |S_I|)$ locations as $\max_{l' \in S_R} \sum_{k-|S_I|} R_{ss}(l')$. Hence, we get the lower bound of $\widehat{d}$ as follows,

$$\frac{F(S_b) - F(S_I) - \omega \cdot \max_{l' \in S_R} \sum_{k-|S_I|} R_{ss}(l')}{(1-\omega) \cdot (k - |S_I|)} = \widehat{d\downarrow} \tag{5.12}$$

Pruning and Termination Rules. We terminate processing an intermediate set S_I when $\forall l' \in S_R$, $\widehat{d} \leq \widehat{d\downarrow}$ is true. This is because, there will exist no location in S_R that can form a better set containing S_I than the best feasible set S_b. Otherwise, we need to prune the particular locations $l' \in S_R$ that satisfy $D_{ss}(l', S_I) \leq \widehat{d\downarrow}$. As we consider, $\widehat{D}_{max} = D_{ss}(S_I)$ is always true for an intermediate set S_I regardless of S_R, the derived lower bound $\widehat{d\downarrow}$ may produce the answer set to miss some eligible locations. Nevertheless, the approach achieves high efficiency with the sacrifice of a certain precision.

Algorithm. For our Approximate (AP) solution, we modify the `Exact` algorithm to introduce the advanced termination and pruning as described above. Here, we only need to

replace the *pruneT* methods at Line 15 in Algorithm 3 using the above mentioned termination and pruning rules based on $\widehat{d}\downarrow$ when an intermediate set S_I contains more than one location. The time complexity of AP is similar to `Exact`, as both the algorithms execute same number of iterations in worst case.

Approximation Ratio. We derive a theoretical bound on the approximation ratio of our Approximate approach (AP). We define the ratio as the socio-spatial score of the *SSLS* set returned by `Exact` algorithm divided by the score of the AP. Let's assume, S' be the approximate set, and S^* be the exact solution of size k, where locations $l' \in S_R$ are added progressively to S', and $l^* \in S_R$ to S^*. To accelerate the Approximate approach, we had derived a relaxed lower bound $\widehat{d}\downarrow$ considering $\widehat{D}(S_I, l) = D_{ss}(S_I)$ is always true $\forall l \in S_R$ (see first paragraph of Section V). This means, AP will discard some *eligible* locations $l \in S_E \subseteq S_R$, whose diversities (w.r.t. an intermediate set S_I) lie between $\widehat{d}\downarrow - (D_{ss}(S_I) - \widehat{D}(S_I, l)) \leq D_{ss}(l, S_I) < \widehat{d}\downarrow$. Let, $\hat{l}^* \in S_E \subseteq S_R$ produces maximum socio-spatial score w.r.t. the intermediate set, therefore, $\hat{l}^*$ will be part of the exact solution S^*. Similarly, let $\hat{l}'$ produces the maximum socio-spatial score among the locations whose diversity w.r.t. S_I is more than $\widehat{d}\downarrow$, e.g., $D_{ss}(\hat{l}', S_I) \geq \widehat{d}\downarrow$. Hence, $D_{ss}(\hat{l}', S_I) > D_{ss}(\hat{l}^*, S_I)$, and $\hat{l}'$ will be part of the approximate solution. Therefore, $F(S_I \cup \hat{l}^*) > F(S_I \cup \hat{l}')$ is true to consider $\hat{l}^*$ in the exact result set. Hence, we can derive $R_{ss}(\hat{l}^*) > R_{ss}(\hat{l}') + \frac{1-\omega}{\omega} \cdot (\psi_i)$ satisfying the above condition, where $\psi_i = D_{ss}(\hat{l}', S_I) - D_{ss}(\hat{l}^*, S_I) > 0$ is the difference in the diversity of the locations $\hat{l}'$ and $\hat{l}^*$ w.r.t. corresponding intermediate set (e.g., S_I).

Following the above process, let's assume that we find the exact set $S^* = \{l_1^*, ..., l_k^*\}$ and the approximate solution $S = \{l_1', ..., l_k'\}$ of size k arranged in decreasing order of relevance score, and $\psi_k = D_{ss}(l_k', S') - D_{ss}(l_k^*, S^*)$. As we progressively add the locations, each time the lower bound $\widehat{d}\downarrow$ gets update. We assign $\widetilde{d} = \min\{\widehat{d}\downarrow\}$ as the minimum score among the lower bounds $\widehat{d}\downarrow$ we derived at each step. Therefore, the lowest total diversity of set S' will be $D_{ss} = k \cdot \widetilde{d}$, and the socio-spatial score of the lowest scoring approximate set S' is $F(S') = \omega \cdot \sum_{l_i' \in S'} R_{ss}(l_i') + (1-\omega) \cdot k \cdot \widetilde{d}$. Similarly, for the exact set S^*, we calculate the best total diversity score as $k \cdot (\widetilde{d} - \epsilon)$, where the diversity of each location should be slightly smaller by ϵ than $\widetilde{d}$. Also, we can calculate the best total relevance score $R_{ss}(S^*) = \sum_{l_i^* \in S^*} R_{ss}(l_i^*) = \sum_{l' \in S'} R_{ss}(l') + \frac{1-\omega}{\omega} \cdot (\sum_k \psi_k)$. Let $\psi = \frac{1}{k} \cdot \sum_k \psi_k$, therefore, $F(S^*) = \omega \cdot (\sum_{l' \in S'} R_{ss}(l') + \frac{(1-\omega)}{\omega} \cdot k \cdot \psi) + (1-\omega) \cdot k \cdot (\widetilde{d} - \epsilon) = \omega \cdot \sum_{l' \in S'} R_{ss}(l') + (1-\omega) \cdot (k \cdot \psi) + (1-\omega) \cdot k \cdot (\widetilde{d} - \epsilon) = \omega \cdot \sum_{l' \in S'} R_{ss}(l') + (1-\omega) \cdot k \cdot (\widetilde{d} + \psi - \epsilon)$. Hence, the approximation ratio will be bounded by:

$$\frac{F(S^*)}{F(S')} = \frac{\omega \cdot \sum_{l' \in S'} R_{ss}(l') + (1-\omega) \cdot k \cdot (\widetilde{d} + \psi - \epsilon)}{\omega \cdot \sum_{l' \in S'} R_{ss}(l') + (1-\omega) \cdot k \cdot \widetilde{d}}$$

Let us assume, $\epsilon_A = \frac{\psi - \epsilon}{\widetilde{d}}$, s.t., $0 \leq \epsilon_A < 1$. Now, if we emphasize on higher diversity (e.g., $\omega = 0$), the approximation ratio will be $1 + \epsilon_A$, while, it returns 1 when emphasize on the relevance (e.g., $\omega = 1$).

5.5 A Fast Exact Algorithm

The socio-spatial diversity of a location is dependent on the other locations in a set. The pruning strategies based on the bound derived by diversity need to re-calculate the diversity scores of the locations whenever the intermediate set gets an update. Therefore, the algorithms based on the bound derived by diversity (e.g., `Exact`) consume more time to execute. In this section, we develop an efficient exact method (`Exact+`) that considers bounds on the relevance scores of the candidate locations. Such a practice will help to search the exact results by reducing the complex diversity computation of the intermediate sets (as performed in `Exact`). The key idea of `Exact+` is motivated by the following observations: (i) As the relevance score of each member in a set is independent of the other members; it will be computationally efficient to design pruning strategies on relevance scores. (ii) Lemma 2 suggests that a location with a relevance score more than $\frac{(1-\omega)}{\omega} \cdot |\delta_d|$ is eligible to be added to an intermediate set. Therefore, we can easily derive a lower bound on relevance score using the above observations to prune a large number of irrelevant locations.

5.5.1 Computing Bounds on Relevance

Here, we introduce some lemmas to derive bounds for pruning locations and early termination. The below lemma aims to compute the maximum possible socio-spatial diversity of a set $S'_I = S_I \cup l'$ when an arbitrary location $l' \in S_R$ is added to an intermediate result set S_I.

Lemma 4 (Maximum Socio-spatial Diversity of an Updated Intermediate Set)**.** *Given an intermediate set S_I, an arbitrary location $l' \in S_R$, the maximum Socio-spatial diversity D^M_{ss} of an updated set $S'_I = S_I \cup l'$ will be $D^M_{ss}(S'_I) = D_{ss}(S_I) + D_{max}$, where $D_{max} = \max_{l' \in S_R} D_{ss}(l', S_I)$ is the maximum diversity generated by an arbitrary location of S_R w.r.t. S_I.*

Proof: Socio-spatial diversity of $S'_I = S_I \cup l'$ is $D_{ss}(S'_I) = \widehat{d} + \widehat{D}$ (Equation 5.4). Therefore, we get maximum socio-spatial diversity of S'_I as, $D^M_{ss}(S'_I) = max(D_{ss}(S'_I)) = max(\widehat{D} + \widehat{d}) = max(\widehat{D}) + max(D_{ss}(l', S_I))$. Since $l' \in S_R$ is an arbitrary location, therefore, $D^M_{ss}(S'_I) = max(\widehat{D}) + \max_{l' \in S_R} D_{ss}(l', S_I) = max(\widehat{D}) + D_{max}$. Hence, $D^M_{ss}(S'_I) = D_{ss}(S_I) + D_{max}$, as $max(\widehat{D}) = D_{ss}(S_I)$ (Equation 5.11).

Now, we will derive the lower bound for the socio-spatial relevance score ($R^{\downarrow}_{ss}$). Such bound will identify the locations that can be added to the current intermediate set. Meanwhile, we label the *reference* location (l_{ref}) that has maximum socio-spatial relevance score among the remaining locations in S_R.

Lemma 5 (Lower Bound of Relevance Score)**.** *Given an intermediate set S_I, reference location l_{ref}, and the remaining location set S_R, the lower bound of Socio-Spatial Relevance Score is $R^{\downarrow}_{ss} = R_{ss}(l_{ref}) + \frac{(1-\omega)}{\omega} \cdot \big(D_{ss}(S_I \cup l_{ref}) - D_{ss}(S_I) - D_{max}\big)$, where $D_{max} = \max_{l' \in S_R} D_{ss}(l', S_I)$ is the maximum diversity of locations in S_R w.r.t. S_I.*

Proof: Suppose the *reference* location $l_{ref} \in S_R$ has been added to the intermediate set S_I, the socio-spatial score of the updated intermediate set $S'_I = S_I \cup l_{ref}$ can be computed as, $F(S'_I) = \omega \cdot R_{ss}(S_I \cup l_{ref}) + (1-\omega) \cdot D_{ss}(S'_I) = \omega \cdot (R_{ss}(S_I) + R_{ss}(l_{ref})) + (1-\omega) \cdot D_{ss}(S'_I)$

Given another location $l' \in S_R \setminus l_{ref}$ s.t. $S''_I = S_I \cup l'$, it needs to be probed only when $F(S_I \cup l') > F(S'_I)$ according to the selection criteria. Hence, we simplify the condition below.

$$\omega \cdot (R_{ss}(S_I) + R_{ss}(l')) + (1-\omega) \cdot D_{ss}(S''_I) > \omega \cdot (R_{ss}(S_I) + R_{ss}(l_{ref})) + (1-\omega) \cdot D_{ss}(S'_I)$$

$$\Rightarrow \omega \cdot R_{ss}(l') > \omega \cdot R_{ss}(l_{ref}) + (1-\omega) \cdot \big(D_{ss}(S'_I) - D_{ss}(S''_I)\big)$$

$$\Rightarrow R_{ss}(l') > R_{ss}(l_{ref}) + \frac{(1-\omega)}{\omega} \cdot \big(D_{ss}(S'_I) - D_{ss}(S_I \cup l')\big)$$

Now, we substitute $D_{ss}(S_I \cup l')$ with its maximum value $D_{ss}(S_I) + D_{max}$ using Lemma 4. Therefore, we get $R^{\downarrow}_{ss} = R_{ss}(l_{ref}) + \frac{(1-\omega)}{\omega} \cdot \big(D_{ss}(S'_I) - D_{ss}(S_I) - D_{max}\big)$.

If S_I contains single location, we compute the lower bound as $R^{\downarrow}_{ss} = R_{ss}(l_{ref}) + \frac{(1-\omega)}{\omega} \cdot \big(D_{ss}(l_{ref}, S_I) - D_{max}\big)$, as $D_{ss}(S'_I) - D_{ss}(S_I) = D_{ss}(l_{ref}, S_I)$ if $|S_I| = 1$. Now, using Lemma 5, we identify the potential locations that can be added to the current intermediate set.

Property 3 (Potential Locations)**.** *A location $l \in S_R$ is a potential candidate location w.r.t. S_I if $R_{ss}(l) \geq R^{\downarrow}_{ss}$.*

5.5.2 Advanced Termination

The `Exact`$^+$ algorithm needs to iteratively check the remaining locations until the best result set is determined. However, it is time-consuming to process all intermediate sets and checks for the feasible set at each iteration. Therefore, we need to introduce some lemmas to derive early termination criteria. First, similar to Lemma 4, we derive below lemma on maximum possible socio-spatial diversity of an answer set.

Lemma 6 (Maximum Socio-spatial Diversity of an Answer Set)**.** *Given set S_I, an arbitrary subset of locations $S'_R \subseteq S_R$ of size $(k - |S_I|)$ s.t., $S' = S_I \cup S'_R$ and $S_I \cap S'_R = \emptyset$; the maximum Socio-spatial diversity $D^M_{ss}(S')$ of the set $S' = S_I \cup S'_R$ is $D^M_{ss}(S') = D_{ss}(S_I) + D^{Max}_{ss}$, where $D^{Max}_{ss} = \max_{l' \in S_R}(\sum_{k-|S_I|} D_{ss}(l', S_I))$ is the sum of the top $(k - |S_I|)$ socio-spatial diversity scores of the locations $l' \in S_R$ w.r.t. S_I.*

Proof: Proof is omitted due to space limitations.

For any intermediate set S_I and a feasible solution S' of size k containing S_I, s.t. $S_I \subset S'$, we derive the below lemma on maximum possible socio-spatial score of S'.

Lemma 7 (Maximum Socio-spatial Score of an Answer Set)**.** *Given an intermediate set S_I, an arbitrary subset of locations $S'_R \subseteq S_R$ of size $(k - |S_I|)$ s.t., $S' = S_I \cup S'_R$, the maximum possible socio-spatial score of S' is $F_{max}(S') = F(S_I) + \omega.R^{Max}_{ss}(S'_R) + (1-\omega).D^{Max}_{ss}$, where, $R^{Max}_{ss}(S'_R) = \max_{l' \in S_R}(\sum_{k-|S_I|} R_{ss}(l'))$ is the sum of top $(k - |S_I|)$ socio-spatial relevance scores of the remaining set S_R, s.t., $S_R \supseteq S'_R$ and $D^{Max}_{ss} = \max_{l' \in S_R}(\sum_{k-|S_I|} D_{ss}(l', S_I))$.*

Proof: Let an arbitrary location set $S'_R \subseteq S_R$ of size $(k - |S_I|)$ is added to S_I s.t. $S' = S_I \cup S'_R$. The socio-spatial score $F(S')$ of S' is, $F(S') = \omega.R_{ss}(S_I \cup S'_R) + (1-\omega).D_{ss}(S')$

$$\Rightarrow F(S') = \omega \cdot R_{ss}(S_I) + \omega \cdot R_{ss}(S'_R) + (1-\omega) \cdot D_{ss}(S') \tag{5.13}$$

To achieve the maximum socio-spatial score $F_{max}(S')$ of S', we need to replace the two unknown variables $R_{ss}(S'_R)$ and $D_{ss}(S')$ in Equation 5.13 with their maximum possible scores.

As $S'_R \subseteq S_R$ is an arbitrary subset of S_R, the maximum possible socio-spatial relevance score $R_{ss}(S'_R)$ of S'_R can be calculated as $R^{Max}_{ss}(S'_R) = \max_{l' \in S_R} \sum_{k-|S_I|} R_{ss}(l')$. Similarly, from Lemma 6, we get the maximum possible socio-spatial diversity of S' as $D^M_{ss}(S') = D_{ss}(S_I) + D^{Max}_{ss}$. After substituting $R_{ss}(S'_R)$ with $R^M_{ss}(S'_R)$, and $D_{ss}(S')$ with $D^M_{ss}(S')$ in Equation 5.13, we get $F_{max}(S') = \omega \cdot R_{ss}(S_I) + \omega \cdot R^{Max}_{ss}(S'_R) + (1-\omega) \cdot (D_{ss}(S_I) + D^{Max}_{ss})$. Therefore, the lemma is proved as $\omega \cdot R_{ss}(S_I) + (1-\omega) \cdot D_{ss}(S_I) = F(S_I)$.

Property 4 (Advanced Termination)**.** *Given an intermediate set S_I, a k-sized answer set $S' \supset S_I$, and the best feasible set S_b, if $F(S_b) > F_{max}(S')$, we will terminate processing S_I.*

The `Exact`$^+$ algorithm incrementally adds locations and checks for a feasible set. It prunes a large number of locations using the lower bound on relevance score, and further terminates processing considerable number of intermediate sets using the advanced termination mentioned in Property 4.

5.5.3 Algorithm

Algorithm 4 summarizes the major steps of `Exact`$^+$, for processing the *SSLS* query. Given a socio-spatial graph G, query user u, the top-k *SSLS* query returns a set S of size k. Initially, the locations of user u are added to S_{Rel} in non-increasing order of their relevance scores and marked as unvisited. In each iteration, the unvisited locations of S_{Rel} are copied to S_R, and the top relevant location of S_R is added to S_I (Line 6). Further, the advanced termination of the current intermediate set is probed using Property 4 (Line 8). In Line 11, the lower bound on relevance score ($R^{\downarrow}_{ss}$) is calculated using Lemma 5, and the potential locations (V_P) are identified using Property 3 (Line 12). The intermediate set S_I is updated with the location $l_{top} \in V_P$ that generates maximum socio-spatial score (Line 14). The inner loop continues until a set of k locations is found, and finally, it returns the best set S.

Time Complexity. The worst case time complexity of `Exact`$^+$ algorithm is $O(n^3k)$, where n is the number of locations of a user. The outer and inner loop of `Exact`$^+$ take $O(n)$ and $O(k)$, respectively. The time complexity of other major parts are: *advTerm* process in $O(n^2)$, *topLocations* selection in $O(1)$, *relBound* computation in $O(n^2)$, *potentialLocs* selection in $O(n)$, and l_{top} selection in $O(n)$.

Steps of `Exact`$^+$. We use the example in Figure 5.1 to demonstrate the steps of `Exact`$^+$ (Algorithm 4) for selecting top-2 *SSLS* set for user u. We show the node exploration steps of

Algorithm 4: SSLS: `Exact`$^+$

Input: Socio-spatial graph G, set size k, query user u
Output: Location set S of size k

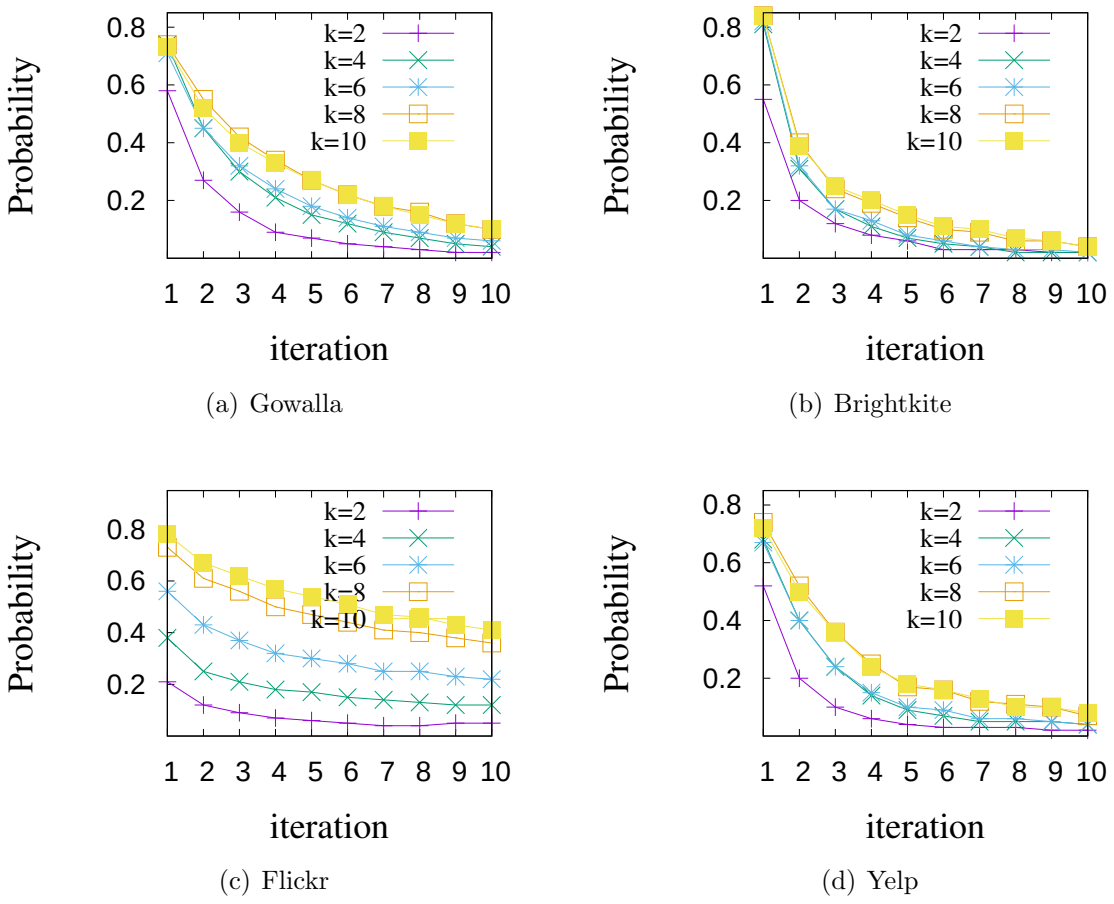
1 Initialize: $S_I \leftarrow \emptyset$, $S \leftarrow \emptyset$, $bestScore \leftarrow 0$,
2 append $\langle l, R_{ss}(l,u)\rangle$ into S_{Rel} in non-increasing R_{ss}
3 mark all locations of S_{Rel} unvisited
4 **while** *no unvisited location exist in* S_{Rel} **do**
5 $\quad S_R \leftarrow unvisited(S_{Rel})$
6 $\quad l \leftarrow S_R.pop(0); S_I.append(l)$
7 $\quad$ **while** $|S_I| < k$ *and* $|S_I| + |S_R| \geq k$ **do**
8 $\quad\quad$ **if** $advTerm(bestScore, S_I, S_R, k)$ **then**
9 $\quad\quad\quad$ break
10 $\quad\quad l_{ref} \leftarrow topLocation(S_R)$
11 $\quad\quad R^{\downarrow}_{ss} \leftarrow relBound(l_{ref}, S_I, S_R)$
12 $\quad\quad V_P \leftarrow potentialLocs(S_R, R^{\downarrow}_{ss})$ *** Property 3
13 $\quad\quad l_{top} \leftarrow \arg\max_{l_i \in V_P} F(S_I \cup l_i)$
14 $\quad\quad S_I.append(l_{top}); S_R.remove(l_{top})$
15 $\quad$ **if** $|S_I| == k$ **then**
16 $\quad\quad$ **if** $F(S_I) > bestScore$ **then**
17 $\quad\quad\quad bestScore \leftarrow F(S_I)$; $S \leftarrow S_I$
18 $\quad\quad S_I \leftarrow \emptyset$
19 $\quad$ mark l in S_{Rel} as visited

the first iteration of `Exact`$^+$ in Figure 5.5 (b). The calculated relevance and diversity scores of the locations are available in Figure 5.4. We set the trade-off parameters as $\alpha = 0.5$, $\omega = 0.5$.

First, we add u's locations in $S_{Rel} = \{p_8, p_7, .., p_{10}\}$ in non-increasing R_{ss} score. Next, the top relevant location p_8 (shown in left bottom corner of Figure 5.5 (b)) is added to intermediate set S_I. As the advanced termination is not satisfied at Line 8, we process to explore the remaining locations in S_R. First, we select the reference location as $l_{ref} = p_7$ (Line 10) shown within red box in Figure 5.5 (b), and the remaining locations in S_R are shown as black dots. The Y and X axes denote the relevance scores and diversity of the locations in S_R w.r.t. p_8, respectively. Now, the lower bound in relevance score w.r.t. $l_{ref} = p_7$ is calculated using Lemma 5, e.g., $R^{\downarrow}_{ss} = 0.565 + \frac{(1-0.5)}{0.5}(0.5 - 0 - 0.66) = 0.405$. The horizontal line in red depicts the lower bound in relevance score. The points $\{p_7, p_6, .., p_2, p_9\}$ on or above the line are labeled as potential locations (V_P), and $\{p_4, p_{10}\}$ are pruned w.r.t. intermediate set $S_I = \{p_8\}$. The location p_5 (marked in green box) among V_P produces the maximum score $F(S_I \cup p_5) = 1.199$ (Line 14). Therefore, in this iteration, we get the best set of 2 locations as $\{p_8, p_5\}$. The process continues until no unvisited locations exist in S_{Rel}. We finally get $S = \{p_7, p_5\}$ as the top-2 *SSLS* solution for u with socio-spatial score $F(S) = 1.451$.

5.5.4 Fast Approximate

Our key observation in Exact$^+$ approach is that feasible sets obtained in first few iterations have higher probabilities to have similar socio-spatial score as top-k SSLS set. We select a portion of users from each dataset (details of datasets are available in Section 5.6) who have 50-100 check-ins. In Figure 5.6, we show the proportion of exact score similarity of the feasible sets obtained by first 10 iterations using Exact$^+$ w.r.t. top-k SSLS set. We consider k as $k = 2, 4, 6, 8, 10$. It is noticed that a feasible set of size k obtained in first iteration, has maximum probability to have similar socio-spatial score as top-k SSLS set. Similarly, when we consider cumulative probabilities (see Figure 5.7), we find first two iterations together (e.g. $C12$) has noticeable increase in score, whereas first three cumulative iterations (e.g. $C123$) is almost similar as $C12$. For example, in Brightkite, when $k = 8$, $C12$ increases from 0.84 to 0.92, however, in $C123$, the cumulative probability score increases by only 0.02.

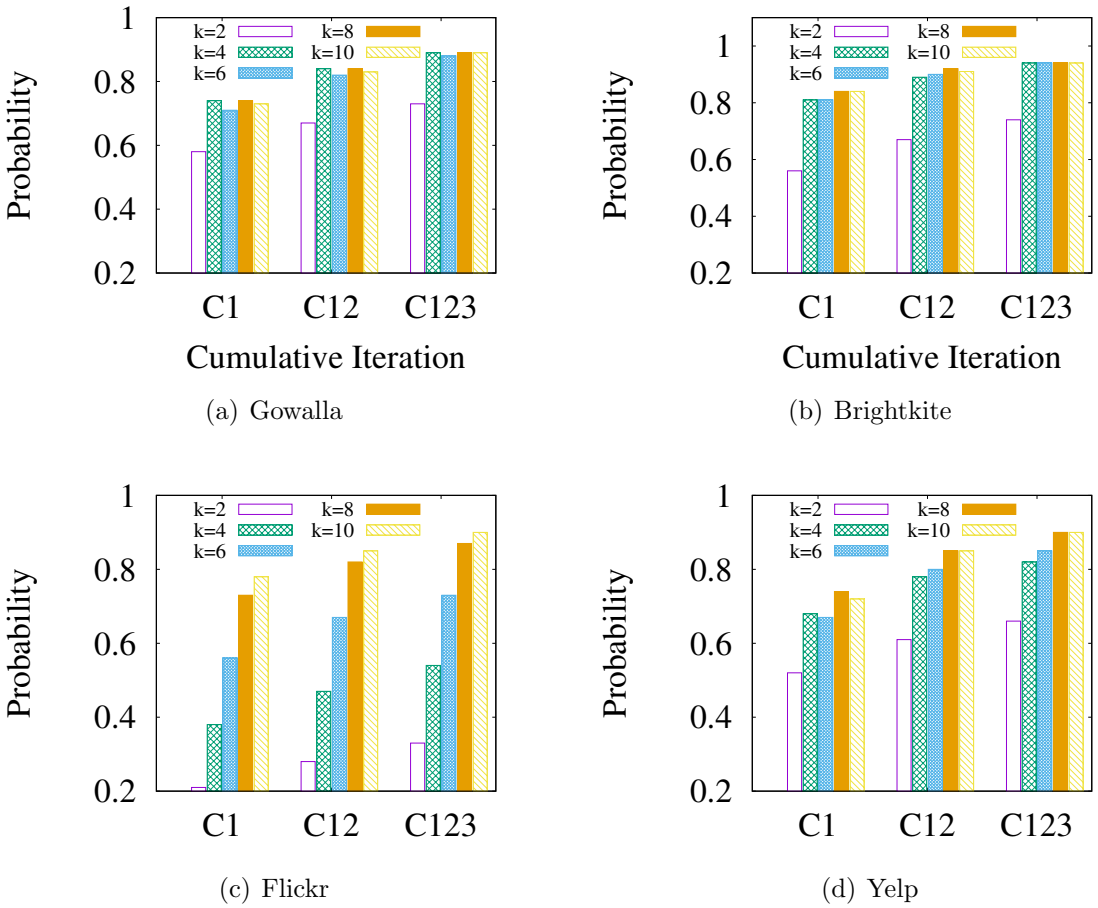


(a) Gowalla (b) Brightkite

(c) Flickr (d) Yelp

Figure 5.6: Proportion of having similar total score as optimal solution w.r.t. number of iterations using Exact$^+$

From the empirical evaluation, we find that if we greedily select the best locations using the procedure of Exact$^+$, the results rapidly converges towards optimal solution in the first



(a) Gowalla (b) Brightkite

(c) Flickr (d) Yelp

Figure 5.7: Cumulative probabilities of first three iteration

Table 5.2: Dataset Statistics

Data	Users	Edges	Checkins	Places	AC	AF	AFC
GW	107,092	456,830	6,442,892	1,280,969	60	8.5	4.6
BK	51,405	214,078	4,491,143	772,783	87	7.7	3.8
FL	189,537	2,028,873	12,592,819	4,896,634	66	21.4	0.3
YL	270,323	1,913,501	5,425,778	192,609	20	14.2	10.4

few iterations. Therefore, to make a reasonable trade-off between the performance and the accuracy, we consider an early termination of `Exact`$^+$ after the first two iterations in our Fast Approximate algorithm.

5.6 Experimental Evaluation

In this section, we present the experimental evaluation of our proposed approaches for *Top-k SSLS* queries: the `Exact` solution (*E*); the *Approximate* solution (*AP*); the `Exact`$^+$ solution (*EP*); and the *Fast Approximate* solution (*FA*). We implement the algorithms using Python 3.6 on Windows environment with 3.40GHz CPU and 64GB RAM. To further validate, we compared with three baselines adapted from existing works:

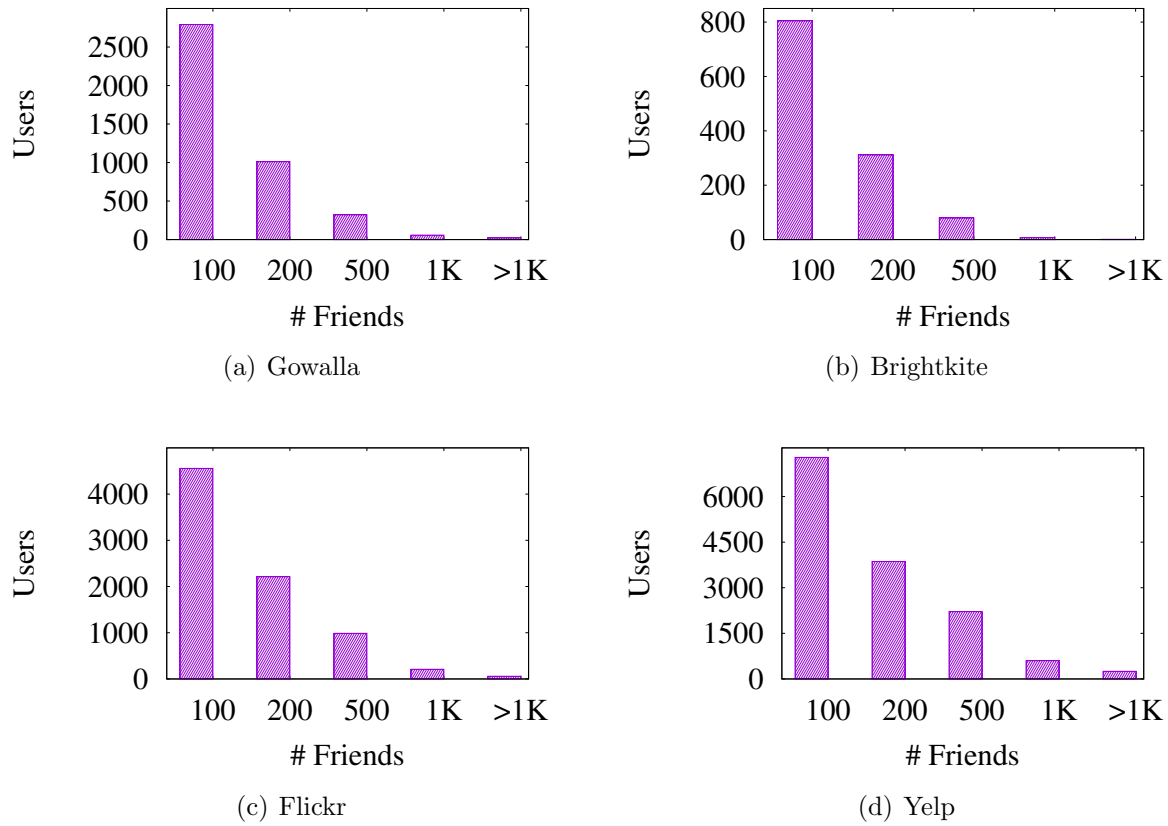
• **GMC [149].** It combines relevance and diversity, and greedily selects the elements based on their marginal contributions. Locations with the highest partial contributions will be selected.

• **Adaptive-SOS [61].** To make the adaption of *SSLS* to SOS [61], denoted as *AS*, we model the social similarity of a pair of locations by using their common users who checked in the location pair. Thus, an edge can be added between the two locations if the similarity is more than a threshold (e.g., 0.4).

• **GNE [149].** It randomly adds a location from the top ranked locations into a temporary result set. Then, it performs swaps between elements of the temporary result set and the most diverse elements of the candidate set.

Datasets. We conduct experiments using four real-world large datasets: *Gowalla* (GW), *Brightkite* (BK), *Flickr* (FL), and *Yelp* (YL). Gowalla [83] and Brightkite [83], each contains the social connections of the users, and the check-ins available over the period Feb. 2009 - Oct. 2010 and Apr. 2008 - Oct. 2010 respectively. One can refer the detailed description about these two datasets at [83]. *Flickr* data was collected using Flickr public API in 2017-18. We establish a social link between a user pair using the *following* information, and consider a check-in if a user has a photo geo-tagged the location. *Yelp* (collected from https://www.yelp.com/dataset/, Round 13, Year 2019) contains friendship network and POIs of users in the form of reviews, and location-tags in users' tips. Table 5.2 presents brief statistics of the four datasets, where the last three columns show the Average Check-ins (AC) by users, Average Friendships (AF), and Average number of Friends that users have at places they have Checked-in (AFC). In Figure 5.8, we show the number of users have friends in the given ranges, where the x-axis labels '100',

'200', '500', '1K' , '>1K' denote the number of friends in the ranges '10-100', '101-200', '201-500', '501-1K', and '>K' respectively.



(a) Gowalla (b) Brightkite

(c) Flickr (d) Yelp

Figure 5.8: Friendship Distribution

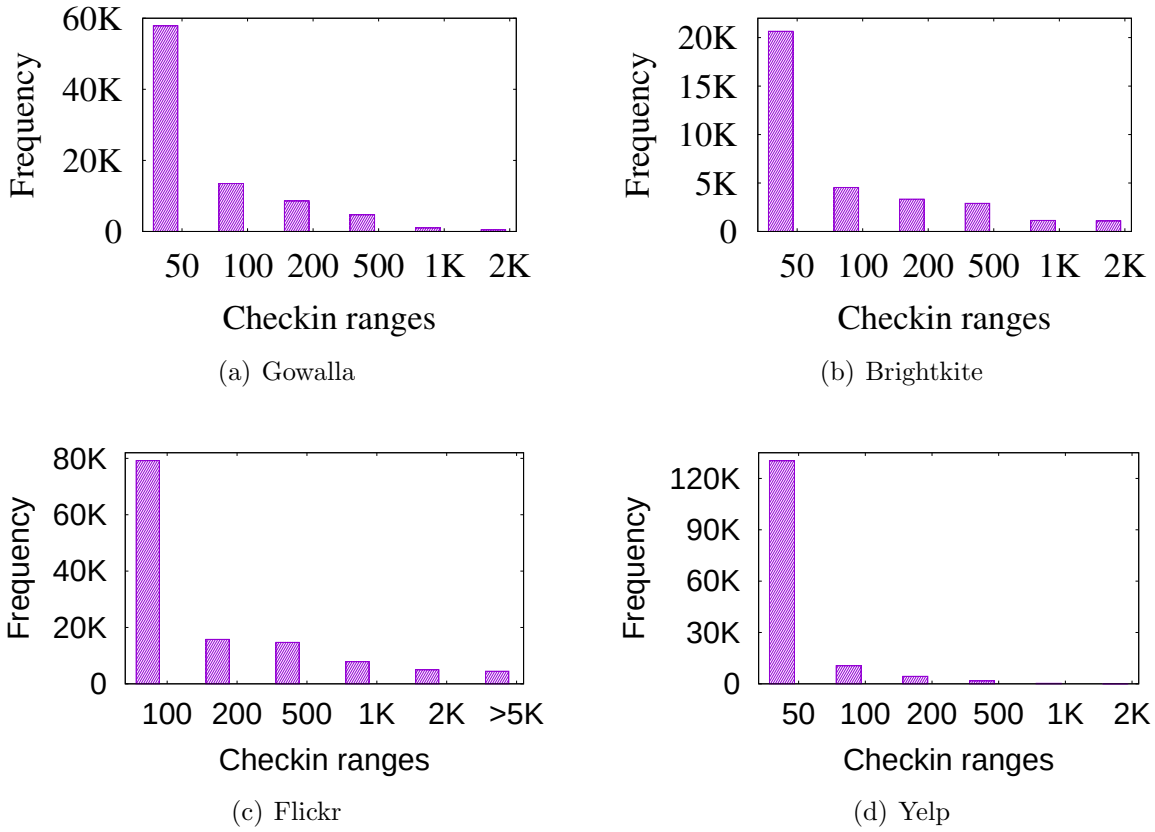
Figure 5.9 shows the check-in characteristics of the users in different check-in ranges.

Evaluation Metrics. *Precision.* It represents the percentage of the common elements (e.g., locations) between the result set returned by an approach and the exact results.

Mean of Minimum Diversity (MMD). Likewise, Minimization of the Mean of Shortest Distance (MMSD) [34, 152], we calculate the Mean of Minimum Diversity (MMD) for a query user u w.r.t. neighbors' locations, i.e., $MMD(u) = \frac{\sum_{v \in V_u} \min dist(L_v, S)}{|V_u|}$. This metric shows how well the selected set of locations S for user u can cover her friends $v \in V_u$. Note, in a socio-spatial domain, $dist$ is considered as socio-spatial distance (D_{ss}) between two locations.

Social Coverage (SC). To measure the social quality of the selected set, we compute social coverage using the percentage of friends who have at least one check-in within θ kilometer (KM) from the selected set S, i.e., $SC(u) = \frac{|v \in V_u \wedge dist(L_v, S) \leq \theta|}{|V_u|} * 100$.

Social Entropy (SE). Given a selected set of locations S of u, let $V_{u,l}$ be the set of u's friends who visits $l \in S$. The social entropy of the set S of u is, $SE = -\sum_{l \in S} p_l \log_2(p_l)$, where, $p_l = \frac{|V_{u,l}|}{\sum_{l_i \in S} |V_{u,l_i}|}$. SE measures the diversity of a location set w.r.t. the participation of its users across different other groups [139]. Here, for a selected location $l \in S \subset L_u$ of u, one of l's corresponding group is considered as the friends who visited l (e.g. $V_{u,l}$). A higher social entropy



(a) Gowalla (b) Brightkite

(c) Flickr (d) Yelp

Figure 5.9: Characteristics of user check-ins

of a set suggests that the selected locations can cover more socially diverse friends.

Table 5.3: Parameters and their values

Parameter	Values	Default
α, ω	$(0, 1)$	0.5
k	$2, 4, 6, 8, 10$	6
Check-in group id	$50, 100, 200, 500, 1000$	100

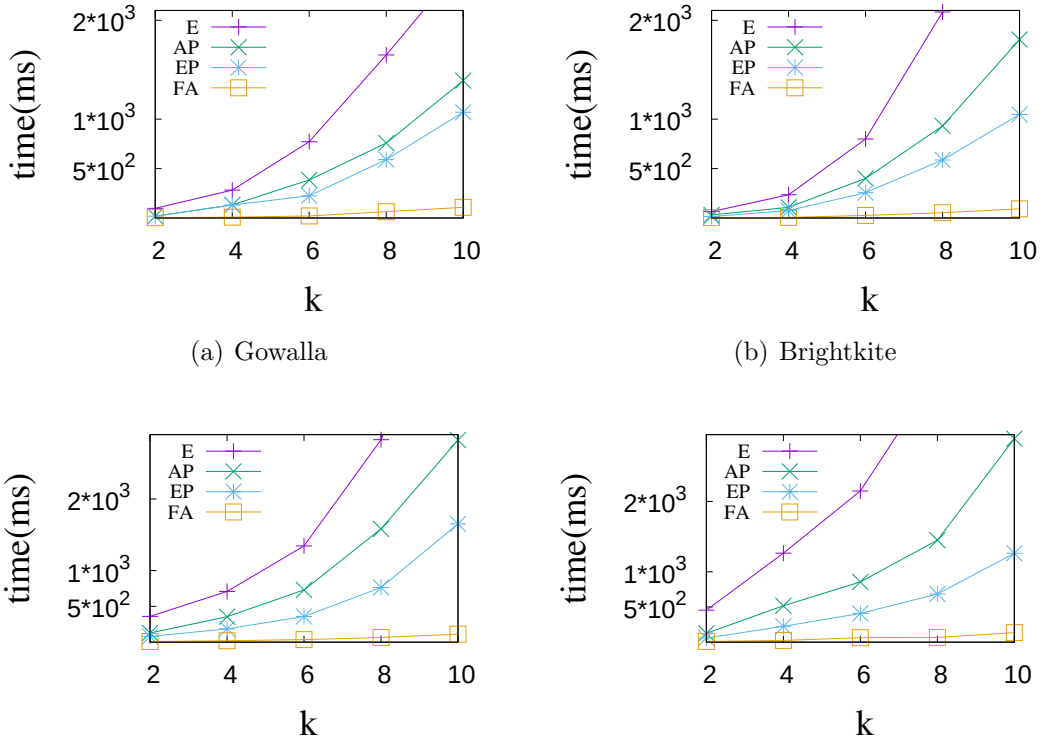
Parameter Configuration. Table 5.3 presents the varied range of the parameters with default values. If there is no specific declaration, then the default values of the parameters will be used when one parameter varies. We only consider users having at least ten check-ins and at least two friends have check-in information. To see the effect of varying the number of check-ins, we divide the users of each dataset into five groups based on the number of check-in locations they have. The check-in group ids 50, 100, 200, 500, and 1000 contain the users with check-in locations in the range 10-50, 51-100, 101-200, 201-500, and 501-1000, respectively.

5.6.1 Efficiency Evaluation

In this section, we compare the scalability of our proposed approaches.

Varying Answer Set Size, k

Figure 5.10 shows the average runtime of our proposed methods by varying k between 2 to 10. The runtime of the algorithms follow similar trends, where E consumes maximum time to process a query. On average, EP is 2 to 3 times faster than AP, and 3 to 6 times faster than E. We notice, AP performs efficiently than EP for those users who have candidate locations with similar relevance scores, and higher diversity. Also, AP is three times faster than E and FA is 9 to 15 times faster than EP in different datasets when k varies from 2 to 10.



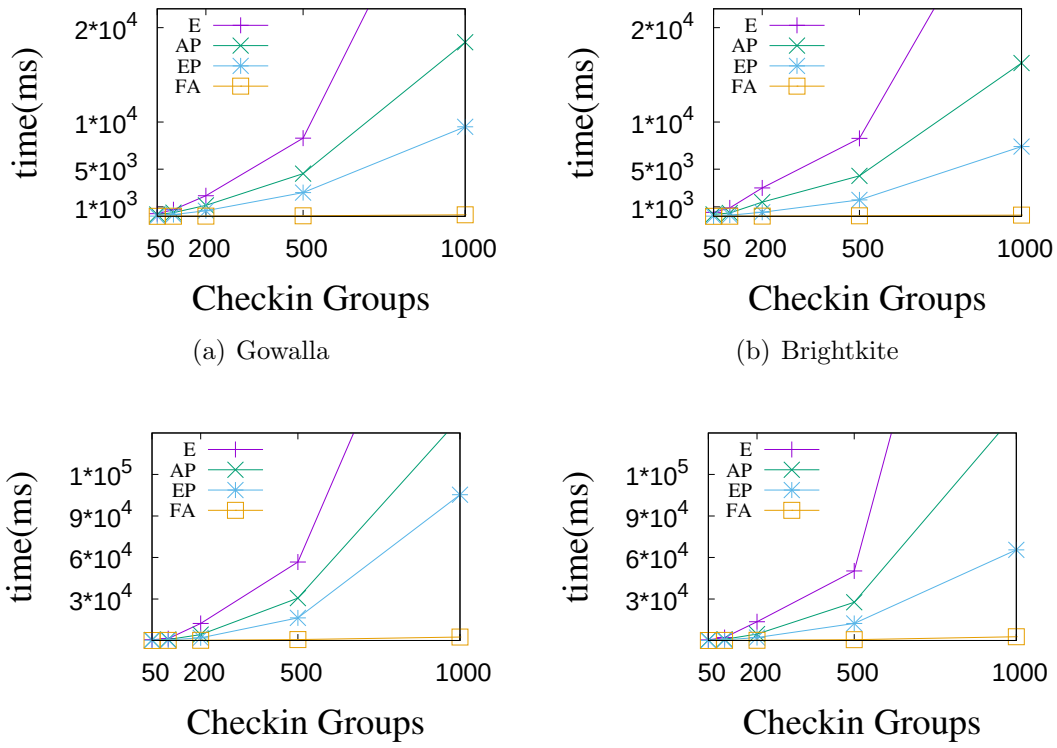
(a) Gowalla (b) Brightkite

(c) Flickr (d) Yelp

Figure 5.10: Varying k

Varying Check-in Group Size

In this experiment, we study the performance of the proposed approaches on the distribution of number of check-ins. Specifically, we show how the size of the check-in locations (e.g., candidate set) of users affects the performance on the approaches. From Figure 5.11, we find the runtime of the proposed approaches, except FA, increases fast with the check-in group size. This is because a considerable amount of possible groups of locations are needed to compare in E, AP, and EP when the check-in group size is large. We also notice EP is much efficient than E and AP. For example, in check-in group 500 in *Brightkite*, EP reports 2.5 and 4.7 times faster than AP and E, respectively. FA performs significantly efficient, even for the large candidate set. For example, in *Gowalla*, FA is 57 times faster than EP when the check-in group id is 1000.



(a) Gowalla (b) Brightkite

(c) Flickr (d) Yelp

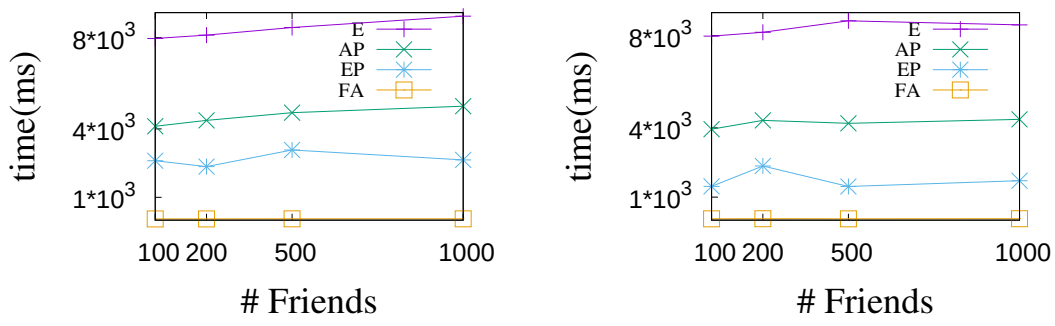
Figure 5.11: Varying check-in groups

Varying Number of Friends

We compare the efficiency of our proposed algorithms by varying the number of social connections users have. To balance user count with sufficient check-ins, we first select the check-in group 500, then divide the users into five groups with a medium to a higher number of social connections. The user group ids 100, 200, 500, 1000 contain the users with 50-100, 101-200, 201-501, 501-1000 friends, respectively. In Figure 5.12, we notice a similar trend in the proposed methods, where a higher number of friends do not affect the efficiency. This is because, the social and spatial relevance scores are pre-computed using the location information of the friends. The proposed algorithms only depend on the number of check-ins a query user has. Therefore, it merely gets affected by the number of social connections.

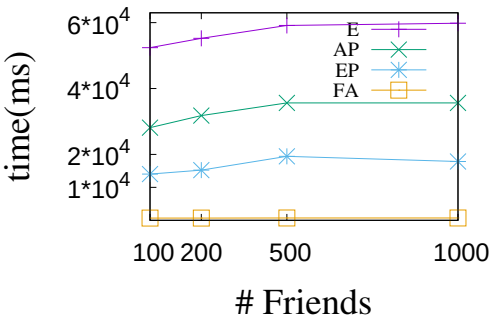
Varying α, and ω

We also test our proposed algorithms by varying the trade-off parameters α, ω. Figure 5.13 shows the average runtime of our proposed algorithms in *Gowalla* and *Yelp* datasets when the trade-off parameters α, ω vary from 0.1 to 0.9. As expected, we do not observe any noticeable change in the efficiency trends in each datasets, where the average execution time of individual algorithm almost remains constant. This is because, these trade-off parameters do not interfere on how a method operates, but only precepts in selecting locations in the result set.

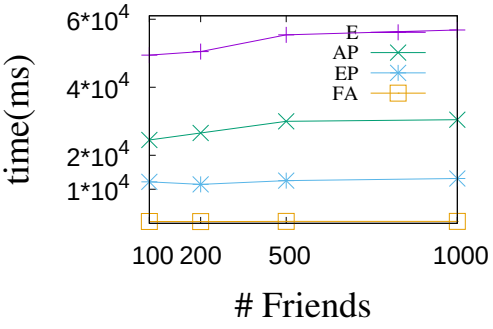


(a) Gowalla

(b) Brightkite

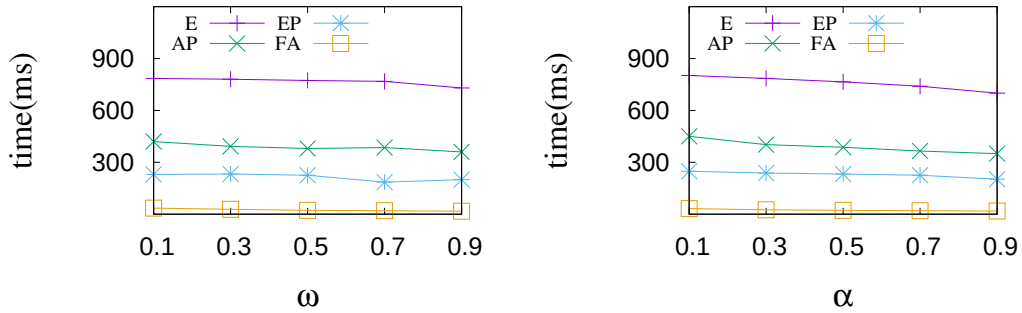


(c) Flickr

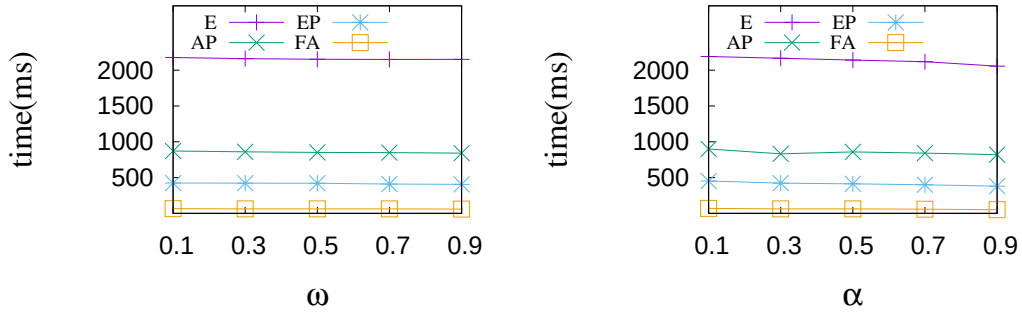


(d) Yelp

Figure 5.12: Varying Number of Friends



(a) Gowalla



(b) Yelp

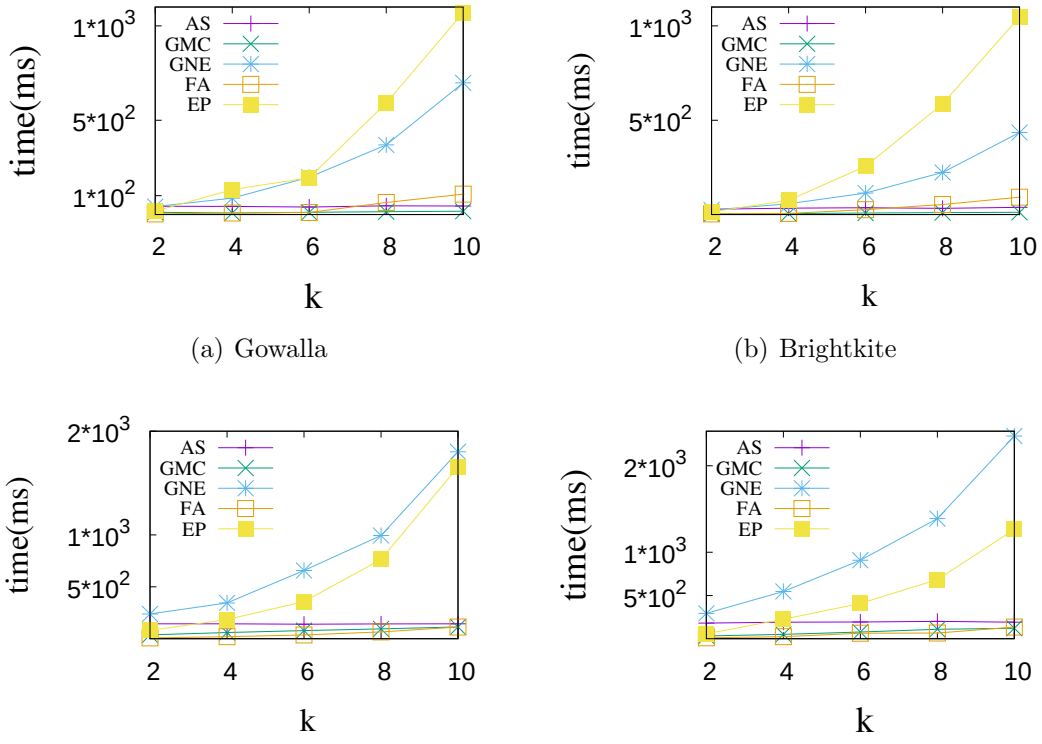
Figure 5.13: Varying ω, α

5.6.2 Comparison with Existing Models

We compare the performance of the existing greedy solutions, e.g., *GMC*, *AS*, *GNE*, with our proposed approaches. For brevity of the presentation, we only show the results using the medium-sized dataset *Gowalla* and the large dataset *Yelp*.

Efficiency

To make a fair comparison between the greedy based existing works and our proposed solutions, we consider our top two efficient algorithms, *EP* and *FA*, in this experiment. Figure 5.14 depicts the runtime of the approaches by varying the answer set size k in default check-in group. In *Gowalla* dataset, *GNE* has higher efficiency than *EP*, but in *Yelp*, it shows an opposite trend. This is because the candidate locations in check-in group 100 of *Yelp* is higher than *Gowalla*. *GNE* always performs slower than *FA*; e.g., in *Yelp*, *GNE* is two times slower than *FA*. In each dataset with moderate-sized candidate locations, *GMC* performs faster than the others.



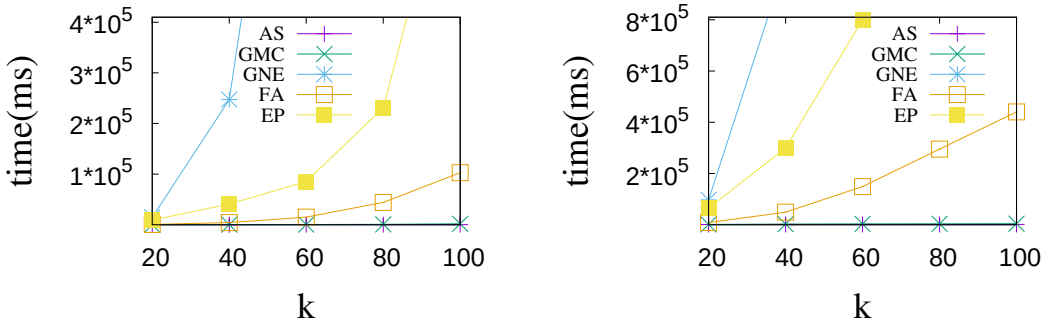
(a) Gowalla (b) Brightkite

(c) Flickr (d) Yelp

Figure 5.14: Varying k

We further evaluate the efficiency of the approaches using larger value of k (e.g., $k = 20, 40, ..., 100$), as shown in Figure 5.15 where we run the experiment on bin id 500 using *Gowalla* and *Yelp* datasets. It shows that both *EP* and *FA* are slower than *GMC* and *AS*. But *EP* generates the exact result set. Even for *FA*, its precision is also much higher than that of *GMC* and *AS* (see Figure 5.17). *GNE* is the most time consuming algorithm. In most cases, *EP* is

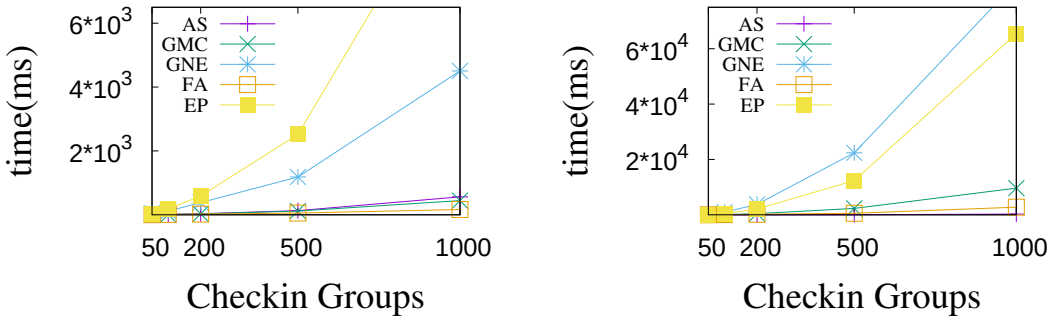
considerably efficient with large answer set size, e.g., in our limited experiment environment *EP* takes only 2.3*10^5ms for *Gowalla* when k is 80.



(a) Gowalla (b) Yelp

Figure 5.15: Varying Large k

Figure 5.16 compares the runtime of the approaches when check-in group size varies. We notice that *FA* is faster than *GMC* when the check-in group size is more than 100. This is because, *GMC* needs more time to calculate the marginal contribution of the locations in large candidate sets. In higher check-in groups, *GNE* takes considerable time to swap the locations in the current result set and the most diverse element among the remaining locations which results a lower efficiency.



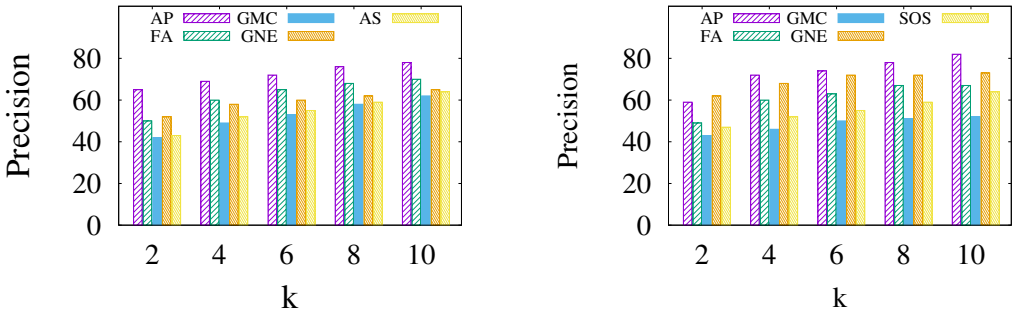
(a) Gowalla (b) Yelp

Figure 5.16: Varying check-in groups

Accuracy

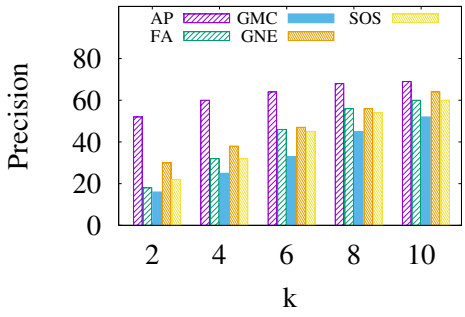
Figure 5.17 demonstrates the precision of the approaches w.r.t. the exact result when k is varied. *AP* has higher precision than the other approaches in each dataset. Although the precision of *FA* is lower than *AP*, *FA* is much efficient (e.g., 10-25 times faster, see Figure 5.10). For example, in *Yelp*, *FA*'s precision is lower than *AP* by 16% only, but its efficiency outperforms *AP* by about 20 times when $k = 6$.

Figure 5.18 shows the average precision of the models when α and ω vary. As the relative trends are similar on other datasets, we only show the effect of α and ω on *Gowalla*. The

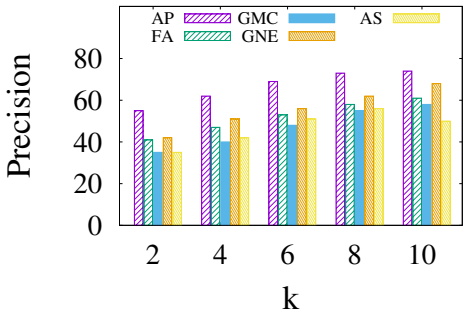


(a) Gowalla

(b) Brightkite



(c) Flickr



(d) Yelp

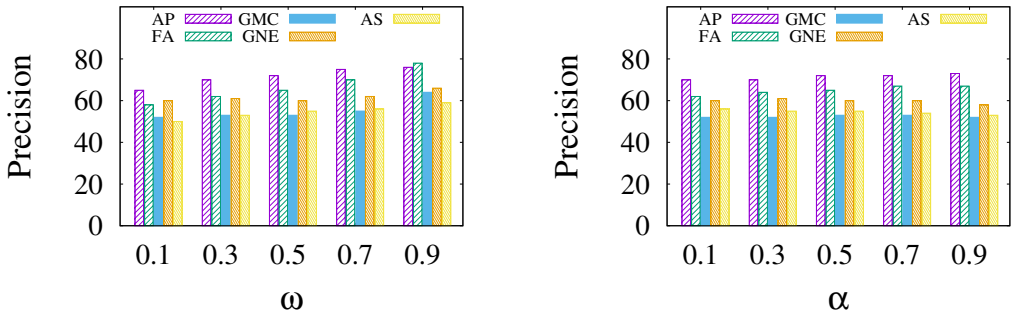
Figure 5.17: Precision

precision of the methods typically increases with ω (e.g., preference to relevance). The *FA* and *AS* methods are influenced by the selection of top relevant location in the result set, which affects the precision when diversity has higher importance than relevance. The precision of *AP*, *GMC*, and *GNE* remain almost constant when ω varies. The variation of α does not affect much in the precision of the approaches when k is set as default. For example, in *Gowalla*, the average precision of *AP* is 71% when $k = 6$ and α varies from 0.1 to 0.9.

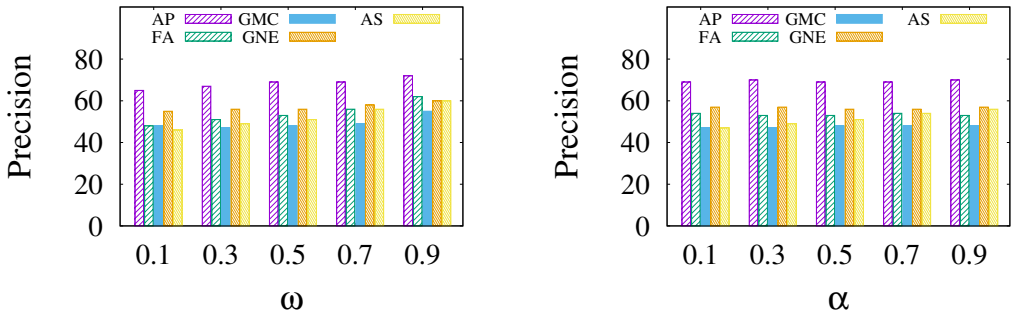
Effectiveness

We compare the socio-spatial qualities of the selected locations using the *MMD* metric. In *Gowalla* (Figure 5.19(a)), the *MMD* of *AS* remains almost constant, while for the other approaches, the *MMD* score decreases smoothly with the increase of k. This is because the *AS* model considers a fixed user-defined threshold to maintain a minimum diversity. In *Yelp*, all the approaches produce lower *MMD* (Figure 5.19(b)). This means the majority of user's friends in *Yelp* have closer check-ins to the selected locations.

Figure 5.20 compares the social coverage (*SC*) of the algorithms. In both the datasets, the relative trends are similar. The top-6 *SSLS* locations in *EP* are co-located with 64% and 74% neighbors in *Gowalla* and *Yelp* datasets, respectively. The *GMC* method has the lowest *SC*, i.e., it reports only 30% in *Yelp*. Interestingly, we find that the social coverage of *FA* is marginally

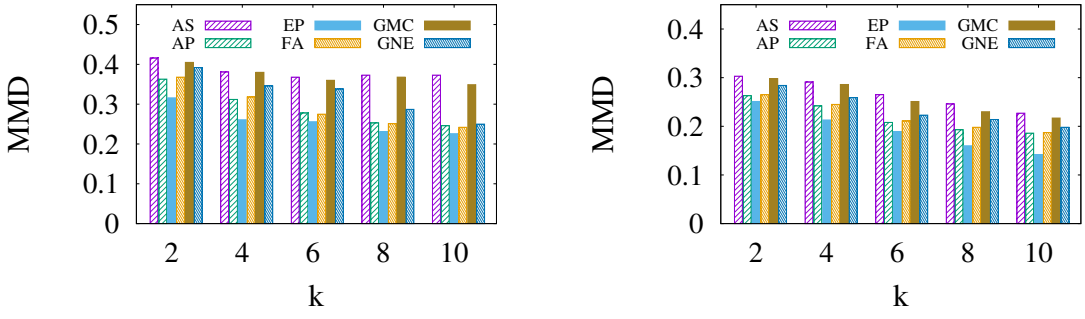


(a) Gowalla



(b) Yelp

Figure 5.18: Precision when varying ω, α

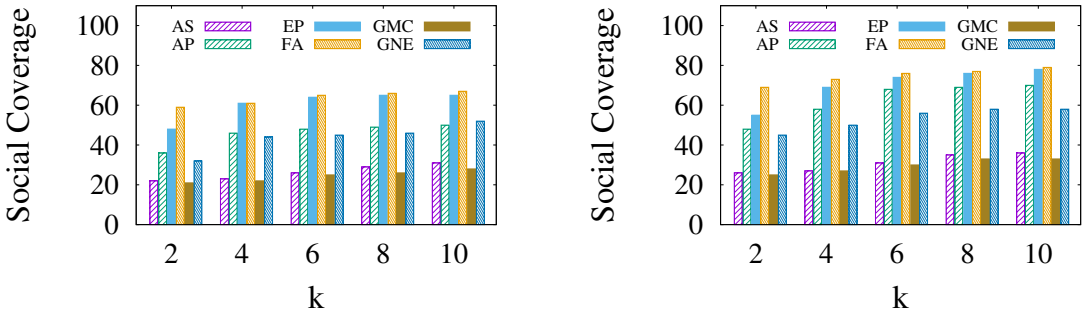


(a) Gowalla

(b) Yelp

Figure 5.19: MMD Comparison

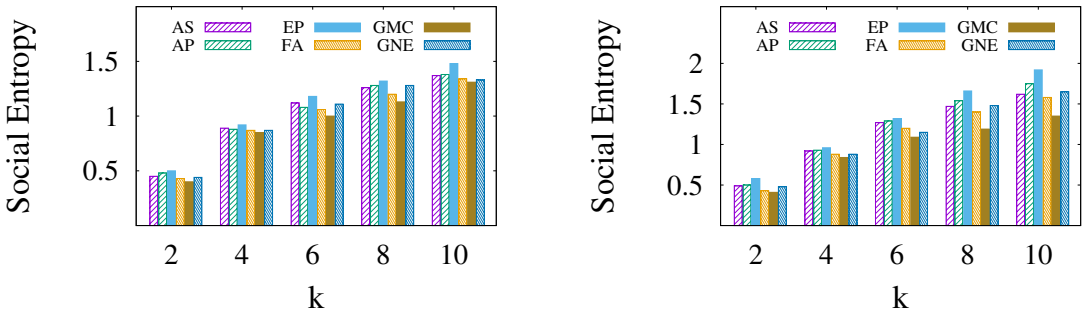
higher than *EP*. This is because, *FA* includes the top socio-spatial relevant locations in the result set. Therefore, the selected set has exact check-ins by a large number of friends.



(a) Gowalla (b) Yelp

Figure 5.20: Social Coverage

Figure 5.21 shows the average social entropy (*SE*) of the approaches when answer set size k varies. Similar trends are followed in both the datasets. The *EP* approach has the highest average *SE*, which means the selected locations by *EP* have diverse participation of friends. Meanwhile, *EP* also has higher social coverage (*SC*) (Figure 5.20). These two metrics *SE* and *SC* together establish that the selected locations in *EP* not only cover a large number of friends, but represent diverse groups. Compared with *GMC* and *GNE*, *AS* has higher social entropy.



(a) Gowalla (b) Yelp

Figure 5.21: Social Entropy

Memory Consumption

We observe that *EP, FA, GMC, GNE, AS* has similar memory consumption, where the average memory usages are reported as 1195MB, 845MB, 2940MB, 1410MB on *Gowalla*, *Brightkite*, *Flickr*, and *Yelp*, respectively. The `Exact` and *AP* methods need to store the intermediate set information in a priority queue, which leads to higher memory cost. For example, in Brightkite, *E* and *AP* consume average 1150MB for the users in check-in group 100.

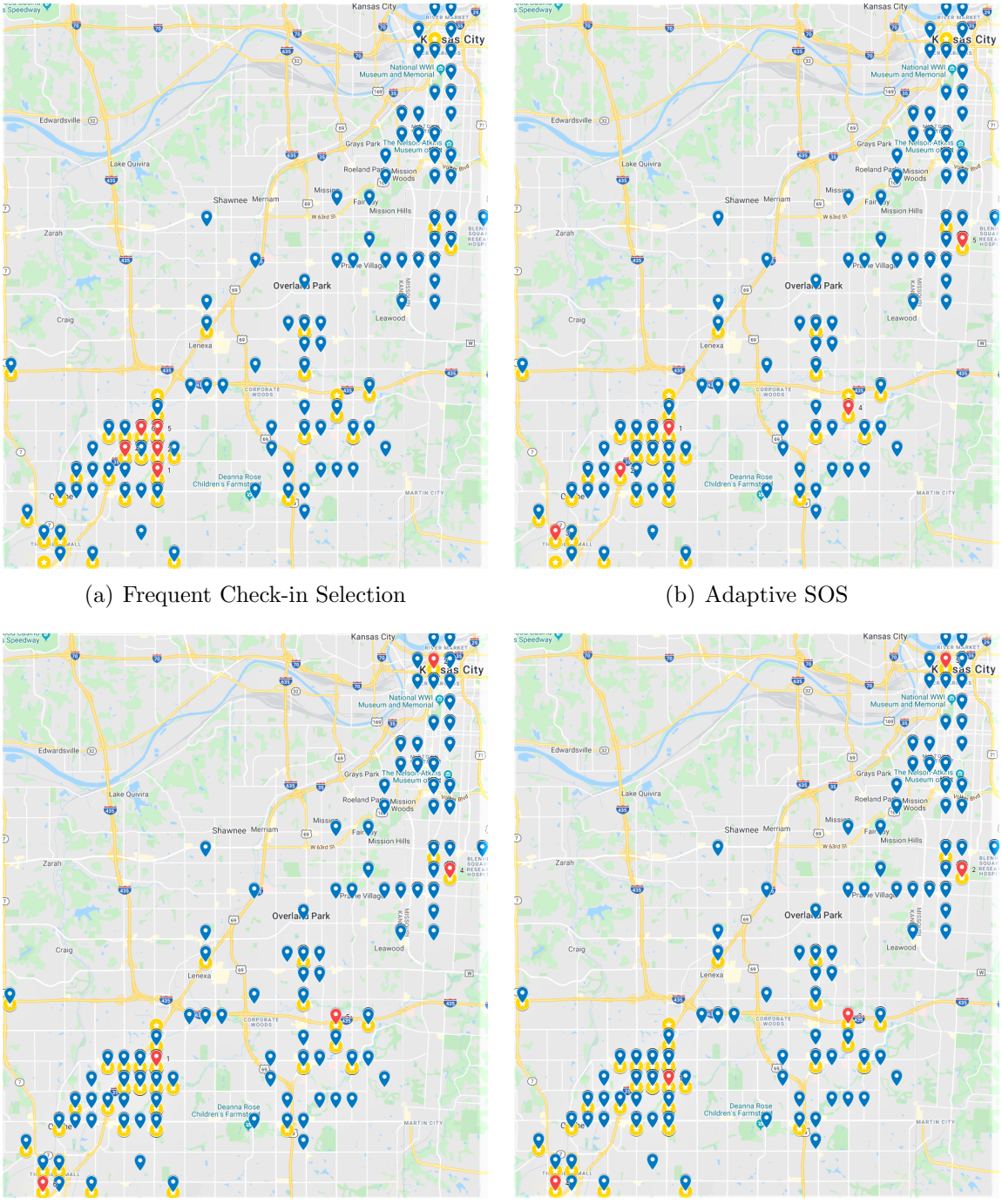
5.6.3 A Case Study

In Figure 5.22, we visualize the selection result of top-5 *SSLS* using Adaptive SOS, `Exact`, and *Approximate* methods considering $\alpha = 0.5$, $\omega = 0.5$. First, we choose a query user (userid '10') from Gowalla [83] dataset, and select the region (38.85, -94.85) to (39.11, -94.58) on map where the user has majority of its check-ins. Further, we obtain the check-in information of the neighbors of the user '10' having at least ten check-in in the mentioned area. There are nine such neighbors available in the selected region. Locations of the user '10' and its neighbors are marked in yellow and blue, respectively (best visible in color with zooming). The user '10' has frequent check-ins concentrated at the red bordered region shown in Figure 5.22(b). The five locations selected by the Adaptive SOS (AS) model are quite distant (shown in red icons in Figure 5.22(b)). However, AS has ignored one important location (39.10, -94.59) (marked as red at NE corner in Figure 5.22(c)) which is included in top-5 SSLS result by our proposed `Exact` and *Approximate* approaches. This location (39.10, -94.59) is spatially relevant to the user '10', as six neighbors (out of nine) have multiple check-ins (total 62) in 7 nearby places within $1.5KM$. In such a configuration, our *Approximate* approach has four common selection as `Exact`. Meanwhile, we provide the snippet of the socio-spatial information of the user '10' (of Gowalla dataset) and its neighbors (who had at least check-ins at the region (38.85, -94.85) to (39.11, -94.58)) at https://github.com/nurjamia/SSLS/blob/master/CaseStydyUserid10_GW.txt.

5.7 Summary

In this chapter, we have proposed a novel problem of *identifying top-k Socio-Spatial co-engaged Location Selection*. It selects k locations for a user from a large number of candidate location set based on the dominance of the combined socio-spatial diversity and relevance score of the selected set over other location combination. The social users may have a large number of locations associated with them, not all are equally important to an application. Therefore, selecting the favorably higher socially and spatially relevant locations are essential.

To do this, we have developed several solutions to solve this NP-hard problem. More specifically, we first propose an `Exact` approach based on the derived lower bounds on the diversity of the already retrieved location set. Then, we have devised an approximate solution based on relaxed bound, which outperforms the `Exact` approach significantly by sacrificing the quality of the answer set slightly. Furthermore, we have developed a more efficient exact method, namely `Exact`$^+$, that effectively prunes the search space by constructing lower bounds based on the relevance and diversity w.r.t. a reference location. `Exact`$^+$ performs 2 to 3 times faster than the approximate solution in default data settings. On the other hand, the Fast Approximate approach is the most efficient one. It executes 9 to 15 times faster than `Exact`$^+$. Finally, the quality of our proposed approaches have been validated by comparing with the state-of-the-art



(a) Frequent Check-in Selection

(b) Adaptive SOS

(c) Exact

(d) Approximate

Figure 5.22: A case study

diversified object selection models. The extensive experimental studies on four real datasets with various socio-spatial characteristics have verified the performance of our proposed approaches.

Chapter 6

Co-engaged Location Group Search in Location-based Social Network

Searching for a user group in social network has been extensively investigated, where the majority of the existing approaches emphasize on finding a cohesive user community in social network. However, a very few studies focus on finding a group of locations in Location-based Social Networks (LBSNs) which are highly co-engaged to socially cohesive user communities. In this chapter, we investigate the problem of identifying Co-engaged Location group Search (*CLS*) from LBSNs where the selected locations should be visited by socially cohesive user groups. Given a group of locations and their corresponding check-in users, we devise a score function to measure the co-engagement score of the location group by combining the social connectivity scores of the checked-in users to the locations and the check-in density to the location group. To solve this problem, we first propose *Filter-and-Verify* algorithm that can effectively filter out the ineligible locations and their check-in users using bound on the check-in counts of each user to a location. We also propose a heuristic based expansion algorithm that can effectively select the intermediate locations and their participating users based on the strict social connectivity. The expansion algorithm appends location points greedily to the solution and prunes some locations based on the aggregated check-in counts of users associated with the locations. Further, we design a greedy algorithm that incrementally adds locations to the solution set using certain greedy criteria. Finally, we measure the effectiveness and efficiency of our proposed solutions by conducting extensive experiments on three real-world datasets.

Chapter map. We give an overall introduction of the co-engaged location group search query in Section 6.1. Section 6.2 formally defines the problem of selecting a top co-engaged location group. The query processing algorithm using Filter-and-verify approach (*FVA*) is discussed in Section 6.3. Two greedy based approaches to the *CLS* solutions, e.g., Greedy Forward Expansion Algorithm (*GFA*) and Greedy Incremental Algorithms (*GIA*) are provided in Section 6.4 and Section 6.5, respectively. The experimental results are presented in Section 6.6, and finally we summarize this chapter in Section 6.7

6.1 Introduction

Location-based services (LBS) have developed a significant importance in recent years. Almost every modern social networking site, commonly referred to as Location-based Social Networks (LBSNs), have enabled their location features. The huge popularity of such LBSNs, i.e., Foursquare, Yelp, Flickr, etc., is playing a vital role in the explosive growth of the location-based services (LBS) market. With the location-enabled features, users can tag their daily activities in different locations through check-ins. These check-ins capture socio-spatial preferences of users and social groups that are being heavily used in many applications such as recommendation systems, location-based advertising, and so on.

However, not all the check-in locations are equally important to all the members of socially cohesive user groups as the locations may reside far from their activity areas or may not have considerable interests. Hence, it is essential to identify a selective set of highly co-engaged locations from the social network which are important to socially cohesive user groups. Identifying such co-engaged locations may provide more sophisticated information to the location-based services which can greatly enhance user experience and togetherness in a spatial region.

There exist several works on socio-spatial user group queries in the LBSN network. Given a set of query nodes and other constraints, the socio-spatial group queries [54, 138, 161, 174] aim to find the best user group near to a single or multiple query location where the users have strong social connections with the group members and have spatial closeness to the query locations. For example, Ghosh et al. [54] propose a framework to find top-k user groups w.r.t. multiple query locations where each group follows the minimum social connectivity constraints. The other works on finding cohesive subgraphs in social networks discover user groups by considering user engagement. For example, Zhang et al. [168] return the user groups as output where the authors adopt k-core to guarantee the engagement of the users in the social network. Such user engagement is based on *structural constraint* that can ensure there exists a considerable number of friends for each individual user (vertex) in each social group. Meanwhile, the work [168] also considers an attribute-based similarity constraint where the similarity between a user pair in the selected group should exceed a given threshold r. The engagement and similarity criteria may often be used together to measure the sustainability of social groups. For instance, Facebook shows that both the engagement (among the friends in a group) and similarity (e.g., similar pages liked and distance closeness) are two important criteria to recommend an existing Facebook group to a user.

Different from the majority of the existing works that find cohesive user groups, in this paper, we demonstrate the spatial level co-engagement of the social users to search for the best location set that can maximize the overall involvement of social users in the selected locations. The participating user groups to the selected locations should satisfy structural constraints, and the check-in density of the participating user groups to the selected location set should be

higher. Meanwhile, the locations in the result set should be distance reachable (close spatially and connected together) containing the query location, which confirms the spatial connectivity between the selected location set. Therefore, identifying such locations with high quality in terms of (i) social connectivity of the participating user groups, (ii) engagements of the participating user group members to the selected locations, (iii) spatially connected within a distance, have good potential to become more useful in the location-based real-life applications.

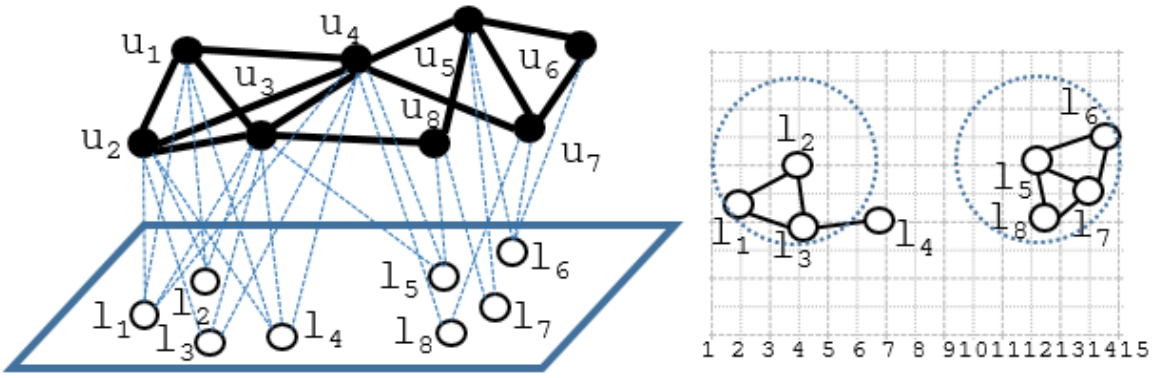


Figure 6.1: Location-based Social Network and Location Graph

Locations visited by the members of a cohesive user group can carry a strong intention of common interests for the specific group of people. Meanwhile, the co-engaged locations in the result groups should be highly checked-in by socially cohesive users. To this end, we model the co-engagement of the locations with a function of check-ins by the socially connected users. Specifically, we propose the top group of Co-engaged Location Search (*CLS*) problem, which given a location-based social network and structural constraints, aims to find the top group of locations, s.t., the group should contain a certain number of locations (say, k) where the users participating at the locations are socially close. Also, a large number of users should have checked-in the locations. In a location-based social network, the groups of co-engaged locations can help to discover interesting social user groups which are promising to become stable and active. The co-engaged locations can greatly enhance users' togetherness and their experiences in a spatial region.

Applications. The *CLS* has a wide range of applications. Among the various usability of the co-engaged groups of locations, we present two relevant applications below.

Tour Planning. Assume a tour planner wants to plan a city tour for a group of visitors. The likely locations for the tour should be selected in a way that are previously visited by the members from other socially cohesive user groups, and should contain a particular place of tourist attraction suggested by the tour planner. Identifying the co-engaged groups of locations with high quality in terms of user engagement at the selected locations will help to emerge a successful tour to the visitors. The selected group of locations should have the potentiality to become active and attractive to new user groups. Therefore, visiting a set of locations that is already known among the other socially connected user groups are more preferable to participate

in various activities together. The Co-engaged Location Search (*CLS*) can be applied to identify the best groups of co-engaged locations to suggest the tour planner to arrange the tour for the visitors.

Event Organization. The event organization applications such as Eventbrite, Meetup support to organize events in various physical locations. Let's assume, an organization wants to organize various events in a group of locations such that, the selected locations should be already known (e.g., visited previously) to the future participating users of the events. Also, the users should be socially connected satisfying some social cohesiveness constraints, such that each member should already know at least a certain number of friends from the corresponding participating user group. The *CLS* can be used to search for the best co-engaged group of locations to organize the events such that the targeted users can easily participate in the events together. The participating users to the location groups will determine the scores to identify the best location set for organizing the events. Meanwhile, the social cohesiveness and size of the participating users can be used as an estimator to the number of participator for the events.

6.2 Problem Formulation

A location based social graph is defined as $G = (V, E, L, E')$, where V is the set of users, E is the set $\{(u, v)|u, v \in V\}$ of edges representing social connections among users, L is the group of locations, and E' is the set of $\{(u, l)|u \in V, l \in L\}$ edges representing the spatial connections between users and locations. We use $e(u, v)$ to represent the social edge between user u and v. Similarly, $e(u, l)$ is used to represent check-in edge between user u and a location l.

Definition 15 (User group)**.** *A user group is a connected subgraph $H = (V_H, E_H)$ where each user node $v \in V_H$ should have minimum m number of social friends.*

Definition 16 (k-Core)**.** *Given an integer k s.t., $k \geq 0$, the k-core of graph G, denoted as $H_k(G)$, is the largest subgraph of G, where $\forall v \in H_k(G)$, $deg(v, H_k) \geq k$.*

However, a k-core H_k may not be a connected graph [44], and the connected k-$\widehat{core}$ components, denoted as $C_k(G)$, are the components returned by the k-core search algorithm. The k-cores of a graph are nested [168]. For two positive integers i and j, if $i < j$, then $H_j \subseteq H_i$. Similarly, the k-$\widehat{core}$ components are also nested, e.g., $C_j \subseteq C_i$ when $i < j$. Similarly, a user v may be a member of more than one k-$\widehat{core}$.

Definition 17 (Location network)**.** *Given a set of location points L, a distance threshold θ, the location network $G_L = (L, E_L)$ is constructed by creating an edge $e(l_i, l_j) \in E_L$ between two locations $l_i, l_j \in L$, s.t., $l_i \neq l_j$ and the distance between l_i, l_j is at most θ, e.g., $dist(l_i, l_j) \leq \theta$.*

In other word, we connect any location points l_i, l_j by an edge if the distance $dist(l_i, l_j) \leq \theta$. In Figure 6.1, we show how to generate edges between a location pair based on a given distance

threshold. We select each location and construct a circle of radius θ centered at the location. Further, we connect edges between the location and the points which fall inside the circle. In Figure 6.1, for a given distance threshold $\theta = 3$, we plot the locations in two-dimensional spatial space. We select location l_2 and connect it with l_1 and l_3 as l_1, l_3 falls inside the circle centered at l_2 with radius 3. Similarly, the locations $\{l_6, l_7, l_8\}$ are connected with edges to l_5. If two locations have a common edge, the location pair is considered as spatially closer w.r.t. the given distance threshold. The concept of the location network is provided here to understand the spatial closeness among the locations.

Definition 18 (θ-distance reachable [77]). *Two locations l_1 and l_i are θ distance reachable if there exists a sequence of locations $\langle l_1, l_2, ..., l_i \rangle$ such that $dist(l_j, l_{j+1}) \leq \theta$ for any $j \in [1, i-1]$.*

In a location network, the θ-distance reachable location set must be connected where two locations are linked by an edge if they are at most θ distance apart. When the concept is clear, we will use 'θ-reachable' in short instead of 'θ-distance reachable'. In Figure 6.1, the location groups $\{l_1, l_2, l_3, l_4\}$ and $\{l_5, l_6, l_7, l_8\}$ are θ-reachable when distance threshold is assigned as $\theta = 3$.

Definition 19 (θ-neighbor). *Given a location l_i, the θ-neighbor of l_i are the set of location points which are within θ distance from l_i. We denote the set of neighbor locations of l_i as $nbr_\theta(l_i)$.*

In a location network (see Definition 17), for a given distance threshold θ, the set of locations having an edge with l_i are considered as the θ-neighbor of l_i. Below, we define the participating user set to a group of locations. Such user set will help us to discover the social and spatial engagement of the locations in a subgraph.

Definition 20 (Participating User Set). *Let L_i be a group of locations and $G_i = (V_i, E_i, L_i, E'_i)$ be an LBSN sub-graph. The set of users V_i is called participating user set to L_i, if each user $u_i \in V_i$ has at least one check-in edge to the location set L_i. E.g., $\forall l_i \in L_i$, $\exists \{u\} \in V_i$ s.t. $e(\{u\}, l_i) \in E'_i$.*

Therefore, each participating user w.r.t. a location group L_i should have at least one check-in to the location set L_i. In Figure 6.1, the user set $\{u_1, u_2, u_3, u_4\}$ are the participating user set to the locations $\{l_1, l_2, l_3, l_4\}$, and each individual user in from the user set is called a participating user to the location set.

On the other hand, for a given social constraint m, the m-$\widehat{core}$ components of the social graph induced by the participating user set are called *participating user group* to the location set L_i. Therefore, a participating user group to the location set should strictly satisfy the social constraint parameter. Nevertheless, a participating user group may be associated with more than one location groups in the network. Besides, multiple socially cohesive participating user

groups (for a given social constraint m) can be identified w.r.t. a particular location group, and a user may be part of multiple participating user group w.r.t. a fixed location group.

For a given LBSN sub-graph $G_i = (V_i, E_i, L_i, E'_i)$, we formally define the set of m-$\widehat{core}$ participating user groups, $\widehat{P}(m, G_i, L_i)$, to a location group L_i below. Each user in a participating user group $\widehat{P}_i \in \widehat{P}$ must have at least m number of social edges and should checks-in at least one location among L_i.

Definition 21 (m-$\widehat{core}$ participating user groups)**.** *Given an LBSN sub-graph $G_i = (V_i, E_i, L_i, E'_i)$, the set of user groups $\widehat{P}(m, G_i, L_i)$ of subgraph G_i are called m-$\widehat{core}$ participating user groups to location set L_i, if $\forall \widehat{P}_i \in \widehat{P}(m, G_i, L_i)$, $e'(u, L_i) \neq \emptyset$, s.t., $u \in V(\widehat{P}_i)$.*

Now, we measure the social connectivity score [7] of a social subgraph as follow.

Definition 22 (Social Connectivity Score)**.** *Given an LBSN subgraph $G_C = (V, E, L, E')$, the social connectivity score of the users in G_C is calculated as, $S_{sc}(G_C) = \frac{2|E|}{|V|(|V|-1)}$, where V is the set of users and E is the set of social edges between the users in V.*

A higher connectivity score of a user group confirms that each user has a large number of neighbors. In the above definition, we define a generalized score function for any sub-graph. From Definition 22, we can calculate the connectivity score of an m-$\widehat{core}$ participating user group $\widehat{P}_i$ using the number of social edges and the users of $\widehat{P}_i$. Similarly, to measure the engagement of the users in a particular set of locations, we need to calculate the check-in density of the locations to the user group.

Definition 23 (Check-in Density)**.** *Given an LBSN subgraph $G_C = (V, E, L, E')$, the check-in density between the users V and the location set L is defined as, $S_{ch}(L, G_C) = \frac{|E'|}{|V|*|L|}$, where E' is the check-in edges between the participating user set V to location set L.*

We follow the existing works [7, 139] to derive a ranking function as the weighted linear combination of social connectivity score and check-in density w.r.t. to a given location group and corresponding participating users. Below, we define the co-engagement score of a location group.

Definition 24 (Co-engagement Score of a Location Group w.r.t. a Participating User group)**.** *Given an LBSN graph $G = (V, E, L, E')$, we measure the co-engagement score (S_{ce}) of a location group $L' \subseteq L$ w.r.t. an m-$\widehat{core}$ participating user group $\widehat{P}_i \subseteq G$ to L' as the weighted linear combination of the social connectivity score and the check-in density,*

$$S_{ce}(L', \widehat{P}_i) = \alpha * S_{sc}(\widehat{P}_i) + (1 - \alpha) * S_{ch}(L', \widehat{P}_i) \tag{6.1}$$

Here, the parameter $\alpha \in (0, 1)$ specifies the relative importance between social score and check-in density. A higher value in α will consider the location group having higher social

connectivity score in the participating user groups. On the other hand, a lower α will emphasize to result the location group having higher check-in density to the location set by participating user groups. In default setting, we set $\alpha = 0.5$. When the context is clear, we simply denote the co-engagement score of a location w.r.t. a participating user group as, $S_{ce} = \alpha * S_{sc} + (1-\alpha) * S_{ch}$.

If a location group has multiple participating user groups satisfying a certain cohesiveness constraint, we aggregate the co-engagement scores of the location set w.r.t. the individual participating user group. Let us assume, a location group $L' \subseteq L$ has p number of m-$\widehat{core}$ participating user groups $\widehat{P}_i$, s.t. $i = 1, ..., p$, satisfying minimum social constraint m. In this case, the co-engagement score of the location group L' is calculated as,

$$S_{ce}(L') = \sum_{i=1}^{p} \big(\alpha * S_{sc}(\widehat{P}_i) + (1-\alpha) * S_{ch}(L', \widehat{P}_i)\big), \tag{6.2}$$

Here, $S_{sc}(\widehat{P}_i)$ and $S_{ch}(L', \widehat{P}_i)$ are the social connectivity score and check-in density of participating user group $\widehat{P}_i$ w.r.t. the location set L', respectively.

In Figure 6.1, let us consider the set of locations $\{l_1, l_2, l_3, l_4\}$ has the participating users $\{u_1, u_2, u_3, u_4\}$. Therefore, the social connectivity score of the users is: $(2*6)/(4*3) = 1$. Similarly, we can calculate the check-in density of the location set $\{l_1, l_2, l_3, l_4\}$ w.r.t. the user set $\{u_1, u_2, u_3, u_4\}$ as: $14/(4*4) = 0.88$. Hence, considering α as 0.5, the co-engagement score of the location set is $\{l_1, l_2, l_3, l_4\}$ calculated as: $0.5*1 + 0.5*0.88 = 0.94$. Meanwhile, the participating users $\{u_1, u_2, u_3, u_4\}$ form 3-core. Therefore, given the check-in graph in Figure 6.1, if the minimum social constraint is set as 3, the co-engaged location group will return $\{l_1, l_2, l_3, l_4\}$ as the answer.

We define the problem as top co-engaged group of location selection (*CLS*) from LBSN. Here, for a given LBSN graph, the *CLS* query will return top co-engaged location group of size k. We formally define the problem as below:

Problem Definition 1 (Co-engaged Location Group Selection Query)**.** *Given a location-based social network $G = (V, E, L, E')$, a query location $l \in L$, a degree threshold m, a distance threshold θ, the CLS problem aims at finding the top group of co-engaged locations $L_i \subseteq L$ of size k containing l, s.t.,*
(1) There exist at least one m-$\widehat{core}$ participating user group $\widehat{P}_i \subseteq G$ to the location group L_i, where $\forall v \in V(\widehat{P}_i)$, $deg(v, \widehat{P}_i) \geq m$.
(2) The locations in L_i should be θ-distance reachable.
(3) $S_{ce}(L_i) \geq S_{ce}(L_j)$ for any $L_j \subseteq L$ s.t., $|L_j| = k$ and $l \in L_j$.

The query parameters m, and θ are provided by the application. The intuition of the *CLS* problem is as follows. Given an LBSN network and a query location l, the *CLS* query will search for the best location group of size k that has maximum co-engagement score among other k-sized location group containing l. Meanwhile, the co-engagement score of a location

Table 6.1: Basic Notations

Symbols	Descriptions
S_{sc}, S_{ch}	Social Score, Check-in Density w.r.t. location set
S_{ce}	Co-engagement Score of location group
m	Social constraint
θ	Distance threshold
V_G	Subgraph induced by nodes V
$V(G)$	Set of users nodes of graph G
$H_m(G)\{C_m(G)\}$	m-core {components} of graph G
$nbr(v, G)$	Set of adjacent vertices of user u in G
$nbr_\theta(l)$	Set of locations within θ distance from l
$deg(v, G)$	Degree of vertex v in G
$U[l]$	Set of users who checked-in location l
$L[u]$	Set of locations checked-in by user u

group is calculated using the social and spatial properties of participating user groups to the locations. The minimum degree of each user of the participating user groups should be m. The basic notations used in this paper are given in Table 6.1.

For ease of presentation, we say that a user v has check-ins at a subset of locations L, s.t. at least one edge $e'(v, L) \subseteq E'$ exist. We use V_G to denote subgraph induced by user set V. On the other hand, $V(G)$ denotes the set of user nodes in grapg G. For a query vertex $l \in L$, we call the set of users V as participating group members to location group L, if $\forall v \in V, deg(v, V_G) \geq m$ and $e'(v, L) \neq \emptyset$. Here, m is the minimum social constraint of the subgraph V_G induced by the user set V.

6.3 Filter-and-Verify Algorithm (FVA)

For a given query location l, a straightforward method to answer a *CLS* query is to enumerate all the possible combinations of k locations containing the query location l. Further, we need to check whether the locations of each set are connected and θ-reachable. Next, we need to validate the minimum social constraint of the participating user groups who have checked-in the selected location group. Finally, the set of locations that maximizes the total co-engagement score in Equation 6.2, is returned as the answer to *CLS* query.

However, the above mentioned straightforward solution may have major drawbacks. For example, enumerating the location groups of size k is time consuming for a large location set L. Also, a large number of potential users are needed to be probed as participating users to the location groups. Hence, the computation overhead renders the straightforward method impractical when the locations need to be selected from a dense spatial region having a large number of users check-in the locations. Therefore, we do not further consider such basic solution in this paper.

To alleviate the above mentioned issues in the straightforward solution of a *CLS* query, we develop *Filter-And-Verify* Algorithm (FVA), that first filters the unnecessary locations, and then validates the remaining location groups using their eligible participating users who can form at least one cohesive group satisfying the minimum social constraint m.

Before, providing the detailed description of *Filter-And-Verify* algorithm, we will identify the candidate locations w.r.t. a query l. We adopt BallTree [114] to index the check-in locations. Based on the query location co-ordinate, we extract a small portion of the spatial region within a certain radius. The set of selected candidate locations will reduce the search space. Therefore, to search for a best co-engaged location set of size k containing the query location l, we will identify the candidate location set using Property 5.

Property 5 (Candidate Location)**.** *Given a socio-spatial graph $G = (V, E, L, E')$, a query location $l \in L$, a distance threshold θ, and size k of a location group containing l, the location points within the circle of radius $(k-1) \times \theta$ centered at l will be considered as the candidate location (CL).*

Proof: According to Definition 18, any two locations $l_1, l_j \in L'$ of a θ-reachable location group L' should satisfy $dist(l_j, l_{j+1}) \leq \theta$, for any $j \in [l_1, l_{i-1}]$. Hence, for a θ-reachable location group of size k, the farthermost location from a fixed query point l will be at $(k-1) \times \theta$ distance. Therefore, the location points within $(k-1) \times \theta$ distance from the query location l are eligible to probe while searching for a k-sized θ-reachable location groups containing the query location l.

Definition 25 (Candidate Users)**.** *For a given set CL of candidate locations, the candidate users $CU = U[CL]$ are the set of users who have at least one check-in to the candidate location set CL.*

In an LBSN network, the participating users to a location group containing query location l should be a subset of candidate users CU. Therefore, identifying the set CU is useful to reduce the search space to construct participating user groups.

For a given query location, we propose the Filter-and-Verify Algorithm (*FVA*) for answering the top co-engaged location groups of size k. In the process of making a location set of size k, the key idea of *FVA* is to filter out some locations and their corresponding check-in users using derived lower bound on the number of check-ins of each user to an updated location set. This approach will discard some locations that can not improve the co-engagement score of the current location set. Specifically, we exploit the check-in users of a location and check whether at least one check-in user to the location can increase the co-engagement score of the existing location set.

6.3.1 m-$\widehat{core}$ components of candidate users, $\widehat{C}(m, CU)$

In this section, we will pre-compute the m-$\widehat{core}$ components from the graph induced by the candidate users CU w.r.t. query location l. To do so, for a given distance threshold θ, answer set size k, social constraint m, we first retrieve the candidate location set CL w.r.t. the query location l. Next, we identify the candidate users $CU = U[CL]$ who has at least one check-in to the candidate location set CL. Further, we construct the subgraph CU_G induced by the candidate users CU. We identify the m-$\widehat{core}$ components $\widehat{C}(m, CU)$ of the subgraph CU_G. Therefore, users in each m-$\widehat{core}$ component $\widehat{C}_i \in \widehat{C}(m, CU)$ derived from the candidate users should also satisfy the minimum social connectivity constraint m.

The intuition behind pre-computing of m-$\widehat{core}$ components $\widehat{C}(m, CU)$ is to get an idea about the membership of an arbitrary user $u \in UL$ w.r.t. a potential participating user group. The below Property states that, a potential m-$\widehat{core}$ participating user group $\widehat{P}_i$ should be a subset of an m-$\widehat{core}$ candidate user component $\widehat{C}_i \in \widehat{C}(m, CU)$, e.g., $\widehat{P}_i \subseteq \widehat{C}_i$.

Property 6. *For a given query location l, social constraint m, candidate location set CL, and candidate user set CU, a participating user group $\widehat{P}_i$ w.r.t. the query location l is a subset of one of the m-$\widehat{core}$ candidate user components $\widehat{C}_i \in \widehat{C}(m, CU)$.*

Proof: We already assume that, for a given query location l, $\widehat{C}(m, CU)$ is the m-$\widehat{core}$ components of the subgraph induced by the candidate users. Therefore, each member of an m-$\widehat{core}$ component has at least m social edges. On the other hand, a participating user group $\widehat{P}_i$ to a location group containing the query location l should be an m-$\widehat{core}$ that maximizes the co-engagement score of a location group. Hence, $\widehat{P}_i$ will be a subset of an m-$\widehat{core}$ candidate user components $\widehat{C}_i \in \widehat{C}(m, CU)$, e.g., $\widehat{P}_i \subseteq \widehat{C}_i$.

6.3.2 Computing score gains

In this section, we will compute the score gains in social connectivity score, check-in density score, and co-engagement score while a new location and its check-in users are probed to include in the corresponding set. In Section 6.3.1, we already assumed that, a potential participating user group must be a subset of one of the m-$\widehat{core}$ components derived from the subgraph induced by candidate users. Hence, we can identify the membership of a potential participating user w.r.t. the set of m-$\widehat{core}$ components $\widehat{C}(m, CU)$. Nevertheless, a user may be contained in more than one m-$\widehat{core}$ component $\widehat{C}_i$. The users from CU which are not a member of any of the m-$\widehat{core}$ component $\widehat{C}_i$, are discarded to further process w.r.t. query location l and only the remaining users will be used to process for constructing a participating user group to a location set. Below, we calculate the score gains of the intermediate solution set.

Let us assume, V_I be an intermediate set of users that may form one potential participating user group if we add more users to the set. We also assume that L_I is the current intermediate

location set. In *FVA* algorithm, we will add locations to the intermediate location set progressively to make it of size k, and at the same time, we will verify whether the users who have check-ins to the newly added locations can increase the total co-engagement score. Meanwhile, we denote $L_R = nbr_\theta(L_I)$ as the set of location points containing θ-neighbor locations w.r.t. L_I. Nevertheless, as discussed in Section 6.3.1, the users in V_I must be member of at least one m-$\widehat{core}$ candidate user components $\widehat{C}(m, CU)$. Hence, we consider that V_I as a subset w.r.t. the candidate user component $\widehat{C}_i \in \widehat{C}(m, CU)$, e.g., $V_I \subseteq \widehat{C}_i$. As we progressively add locations from current L_R to the intermediate set L_I, we need to identify which users of the newly added location l' can help to construct at least one cohesive participating user group. We assume, V_R be the set of users who have check-ins at l' and also a member of $\widehat{C}_i$, e.g., $V_R = U[l'] \cap \widehat{C}_i$. A user $v \in V_R$ will be added to the intermediate user set V_I if $V_I \cup \{v\}$ generates a higher co-engagement score than the current score. Therefore, to probe a user to be a potential member of a participating user group, we need to first develop a lower bound on the number of additional check-ins to the updated location group $L_I \cup l'$. The total co-engagement score of an intermediate location group is computed based on the weighted combination of the social score and check-in density. Therefore, we first calculate the score gain of an updated intermediate location group w.r.t. the previous intermediate location group when a new location l' is added to L_I and an arbitrary user $v \in U[l']$ is added to V_I.

Social score gain: Let g_c be the total social connectivity of the members of the intermediate user set $V_I \subseteq \widehat{C}_i$ and δ_g be the additional social connectivity when a new user $v \in \widehat{C}_i$ is added to V_I. Therefore, the social score gain of the current user group $V_I \cup \{v\}$ is,

$$\begin{aligned}
\delta S_{sc} &= \frac{g_c + \delta_g}{(|V_I|+1) * |V_I|} - \frac{g_c}{|V_I| * (|V_I|-1)} \\
&= \frac{g_c}{|V_I|} * \Big(\frac{|V_I| - 1 - |V_I| - 1}{(|V_I|+1) * (|V_I|-1)}\Big) + \frac{\delta_g}{(|V_I|+1) * |V_I|} \\
&= \frac{1}{(|V_I|+1) * |V_I|}\Big(\delta_g - \frac{2g_c}{(|V_I|-1)}\Big)
\end{aligned}$$

For social score gain calculation, we assume that the newly added user $v \in U[l']$ is a member of the candidate user component $\widehat{C}_i$ as such the intermediate user set $V_I \subset \widehat{C}_i$ is a subset of the same candidate user component $\widehat{C}_i$.

Now, to calculate the gain in check-in density due to adding a new user (among the check-in users of an existing location set), we need to count the number of additional check-in edges available from the user node to the existing location set. Below, we are calculating the check-in density gain of an existing location set due to adding a user to the participating user set.

Check-in density gain: Let h_c be the total check-in edges from the members of $V_I \subseteq \widehat{C}_i$ to the intermediate location set L_I, and δ_h be the additional check-ins due to adding a new user

$v \in \widehat{C}_i$ to V_I and a location l' to L_I (s.t., $v \in U[l']$). The check-in score gain is:

$$\begin{aligned}\delta S_{ch} &= \frac{h_c + \delta_h}{(|V_I|+1) * (|L_I|+1)} - \frac{h_c}{|V_I| * |L_I|} \\ &= \frac{1}{(|V_I|+1) * (|L_I|+1)} \Big(\delta_h - \frac{h_c * (|V_I| + |L_I| + 1)}{|V_I| * |L_I|} \Big)\end{aligned}$$

Here, we have considered that a new location l' is going to be added to the existing intermediate location set L_I and one user $v \in U[l']$ is also going to be added to V_I. Therefore, the total number of locations will be $(|L_I|+1)$ and the number of users will be (V_I+1).

Total score gain: The gain in total score of the updated location group w.r.t. the previous group can be obtained as,

$$\begin{aligned}\delta =& \alpha * \delta S_{sc} + (1-\alpha) * \delta S_{ch} \\ =& \frac{\alpha}{(|V_I|+1) * |V_I|} \Big(\delta_g - \frac{2g_c}{(|V_I|-1)} \Big) \\ &+ \frac{1-\alpha}{(|V_I|+1) * (|L_I|+1)} \Big(\delta_h - \frac{h_c * (|V_I| + |L_I| + 1)}{|V_I| * |L_I|} \Big)\end{aligned} \tag{6.3}$$

Here, we calculate the score gain of an updated intermediate location group when a check-in user $v \in U[l']$ of the new location l' is added. Hence, the participating user set will also be updated with the new user v among the checked-in users to the newly added location. Our target is to remove such users from the potential user set that can not contribute higher co-engagement score. If all the associated users (e.g., checked-in users) of a location l' are not eligible to be added in V_I, we will ignore the location l' for further processing w.r.t. the current intermediate set.

Therefore, a positive gain in total score implies that a user $v \in U[l']$ of a new location l' will produce a higher score than the current intermediate location set L_I. In this way, if at least one check-in user of l' generates $\delta > 0$, the location l' is eligible to be added to the current intermediate set L_I. Hence, using Equation 6.3, we derive the below expression that will help us to determine whether a user checked-in a new location can contribute a higher co-engagement score.

$$
\begin{aligned}
&\delta = \alpha * \delta S_{sc} + (1-\alpha) * \delta S_{ch} > 0 \\
\Rightarrow &\frac{\alpha}{(|V_I|+1) * |V_I|}\Big(\delta_g - \frac{2g_c}{(|V_I|-1)}\Big) \\
&+ \frac{1-\alpha}{(|V_I|+1) * (|L_I|+1)}\Big(\delta_h - \frac{h_c * (|V_I|+|L_I|+1)}{|V_I| * |L_I|}\Big) > 0 \\
\Rightarrow &\frac{(1-\alpha) * \delta_h}{|L_I|+1} > \frac{1-\alpha}{|L_I|+1}\Big(\frac{h_c * (|V_I|+|L_I|+1)}{|V_I| * |L_I|}\Big) \\
&\qquad - \frac{\alpha}{|V_I|}\Big(\delta_g - \frac{2g_c}{|V_I|-1}\Big) \\
\Rightarrow &\frac{h_c(|V_I|+|L_I|+1)}{|V_I| * |L_I|} - \frac{\alpha(|L_I|+1)}{(1-\alpha)|V_I|}\Big(\delta_g - \frac{2g_c}{|V_I|-1}\Big) < \delta_h
\end{aligned}
\tag{6.4}
$$

6.3.3 Maximum additional social connectivity δ_g for a new user

The participating user group is constructed by adding an eligible user to the intermediate user set that can increase the social score of the user group by satisfying the minimum social constraint m. The Property 6, states that a participating user group must be subset of one of the m-$\widehat{core}$ candidate user components $\widehat{C}_i \in \widehat{C}(m, CU)$. Therefore, the intermediate user set V_I should be subset of one m-$\widehat{core}$ candidate user component, e.g., $\widehat{C}_i$. However, initially the intermediate user set $V_I \subseteq \widehat{C}_i$ may contain less than m users. Therefore, if the number of users $|V_I|$ in the intermediate user set V_I is less than m, the maximum social connectivity of the user group can be increased by $2 * |V_I|$ by the newly added member v. However, among the users in $V_R \in nbr(V_I) \cap \widehat{C}_i$, if the maximum degree is $degMax$ w.r.t. the subgraph induced by $V_I \cup V_R$, any user $v \in V_R$ can not be connected to more than $degMax$ members of V_I. Thus we get, the maximum possible edges of an arbitrary user $v \in V_R$ to the intermediate user set V_I as $g_{max} = min(degMax, |V_I|)$. Therefore, in this case, the maximum total connectivity of V_I can be increased by at most $2 * g_{max}$ when a checked-in user of the newly added location is considered in the existing intermediate user group V_I.

6.3.4 Lower Bound on Check-in

Using Equation 6.4, we derive a lower bound on number of check-ins a user v should have when a new location is added to the existing intermediate location group L_I. The LHS of Equation 6.4 has one unknown variable δ_g. In Section 6.3.3, we have derived the maximum additional social connectivity δ_g as $2 * g_{max}$. Thus, we replace δ_g with $2 * g_{max}$ in Equation 6.3 to get the check-in lower bound as below:

$$
\delta_h^{\downarrow} = \frac{h_c(|V_I|+|L_I|+1)}{|V_I| * |L_I|} - \frac{\alpha(|L_I|+1)}{(1-\alpha)|V_I|}\Big(2 * g_{max} - \frac{2g_c}{|V_I|-1}\Big) \tag{6.5}
$$

Therefore, if the number of check-ins of a new location $v \in V_R$ to the location group L_I (e.g., $|h(v, L_I)|$) is less than $\delta_h^{\downarrow}$, the user v will not be added to the intermediate user set V_I.

6.3.5 User pruning using check-in bound

Using the lower bound $\delta_h^{\downarrow}$ on check-in, we can ignore a large number of users to the potential participating user group. Any user $v \in V_R$ having less than $\delta_h^{\downarrow}$ check-ins to the intermediate location group L_I will be ignored to consider them in the intermediate user set V_I. Using Lemma 8, we can decide whether we should add a user $v \in V_R$ to the intermediate user set V_I.

Lemma 8 (User pruning)**.** *Let V_I be an intermediate set of participating users w.r.t. an intermediate location group L_I. A user $v \in V_R$ can not improve the score of the current intermediate location set L_I w.r.t. the participating user group if $|h(v, L_I)| < \delta_h^{\downarrow}$, and thus the user v will not be added to the intermediate participating user group V_I.*

Proof: Let g_{max} be the maximum number of social connections between an arbitrary user of V_R and the current intermediate user set V_I. Using the value g_{max}, we compute the lower bound on check-in of a user in Equation 6.5. Since v has the maximum social connection g_{max} to V_I, therefore, the number of check-ins by v to L_I should not be less than the lower bound on check-ins $\delta_h^{\downarrow}$ to guarantee that the co-engagement score of $L_I \cup \{l'\}$ is greater than the current set L_I.

In *FVA*, to probe a location to be added into an intermediate set, we need to check the contribution of each check-in user (of the location) to the intermediate location group. We check each user one by one to validate their eligibility to be included in the current intermediate user set V_I. In the next section, we discuss the location pruning process w.r.t. the current intermediate set L_I.

6.3.6 Location pruning using check-in bound

To prune some locations while searching for a potential location group, we need to ensure that no check-in members to those locations can be part of the participating user group. Given a new location l', if all the check-in users $U[l']$ of l' satisfy the below lemma, we will ignore the location l' to the intermediate location set L_I.

Lemma 9 (Location pruning)**.** *Let V_I be an intermediate set of participating users w.r.t. an intermediate location set L_I. A new location $l' \in L_R$ can not improve the score of the current intermediate set L_I, if $\forall v \in U[l'] \subseteq V_R$, $|h(v, L_I)| < \delta_h^{\downarrow}$ holds, and hence the location l' will not be added to L_I.*

Proof: From Lemma 8, we can conclude, if all the unvisited check-in users of a location l' do not have check-ins more than the lower bound $\delta_h^{\downarrow}$, the co-engagement score of the current location group can not increase. Therefore, l' will not be added to L_I.

Note, we will validate the pruning condition in Lemma 9 for each set $V_R = nbr(V_I) \cap \widehat{C}_i$, where $\widehat{C}_i \in \widehat{C}(m, CU)$ is an m-$\widehat{core}$ component of the subgraph induced by candidate users CU.

6.3.7 Algorithm

Algorithm 5: Filter-And-Verify:$CLS(G, l, k, m, \theta)$

Input: LBSN $G = (V, E, L, E')$, query location l, group size k, degree m, distance threshold θ
Output: Top co-engaged location group of size k containing l

1 Initialize: Priority Queue $Q \leftarrow \emptyset$, $S \leftarrow \emptyset$, $bestScore \leftarrow 0$
2 $CL \leftarrow candLoc(G, l, k, \theta)$, $\widehat{C} \leftarrow \widehat{core}(m, U[CL], G)$
3 $L_c \leftarrow connectedLocs(CL, l)$
4 $L_I \leftarrow \{l\}$, $L_R \leftarrow nbr_\theta(l)$, $V_I \leftarrow U[l] \cap \widehat{C}$, $V_R \leftarrow \emptyset$
5 $Q.push(L_I, L_R, 0)$
6 **while** Q *is not empty* **do**
7 $L_I, L_R \leftarrow Q.pop()$
8 **while** $|L_I| < k$ *and* $|L_R| \geq 1$ **do**
9 $l' \leftarrow nextLoc(L_R)$, $L_I.add(l')$;
10 $L_R.add(nbr_\theta(l') \cap L_c \setminus L_I)$
11 $V_R \leftarrow U[l'] \setminus V_I$
12 $V_R \leftarrow pruneUser(V_R, V_I, L_I, \widehat{C})$
13 **if** $V_R == \emptyset$ **then**
14 $L_I.remove(l')$; break
15 $V_I.add(V_R)$
16 $Q.push(L_I, L_R - \{l'\}, |L_I|)$
17 $Q.push(L_I - \{l'\}, L_R - \{l'\}, |L_I| - 1)$
18 **if** $|L_I| == k$ **then**
19 $flag, score \leftarrow validateGroup(U[L_I], m, \widehat{C})$
20 **if** $flag = true$ *and* $score > bestScore$ **then**
21 $S \leftarrow L_I$; $bestScore > score$; break;

The pseudo codes of Filter-and-Verify Algorithm (FVA) is provided in Algorithm 5. Given an LBSN graph G, the CLS returns a k-sized θ-distance reachable location group that has maximum co-engagement score. The FVA algorithm considers the following inputs: a query location l, location group size k. Using BallTree [114] spatial index, we extract a small portion of the locations containing the candidate locations (CL) within $(k-1) * \theta$ distance to the query location l. As defined in Definition 17, we further create the location network using the candidate locations (CL). We identify the m-$\widehat{core}$ components from the subgraph constructed using the check-in users at CL (Line 2). Further, in Line 3, we identify the connected location set L_c from the location network constructed by CL containing query location l. For checking the connectivity of L_c, we use Euler tour tree [67]. In this step, we filter out a large number of locations from the candidate location set which are not contained in L_c.

We initialize the intermediate location set L_I with the query location l, and the neighbors of l within θ distance are arranged in L_R in decreasing order of number of the check-in users to the locations. The check-in users at l are added to the intermediate user set V_I. Next, a priority queue Q is maintained where each entity contains a tuple of intermediate locations L_I, neighbor set L_R of L_I. Entities in Q are maintained in descending order of the size of L_I. In each iteration, the top entry from Q is popped (Line 7). Then, an inner loop is executed that fetches the next location l' from L_R and is added to L_I. After that, the intermediate location set L_I is updated with the selected location l'. Pruning of users from V_R is further executed using Lemma 8 (Line 12). If no users are remaining in V_R, we exclude the location l' to further process w.r.t. L_I. Finally, two new entries are pushed into Q, where the first entry contains the updated $(L_I, L_R - \{l'\})$, and the second entry contains the previous state $(L_I - \{l'\}, L_R - \{l'\})$ of the intermediate set. The code block between Line 18 to Line 21 is executed to find the best co-engaged location group S.

Time Complexity. Time complexity of Algorithm 5 is calculated as follows. For a query location, the maximum number of entries are $O(2^k)$. The inner while loop is executed at most $O(k)$ iterations. The *nextLoc*, *pruneUser*, and *pruneLoc* methods incur $O(1)$, $O(|V|)$, $O(C^2)$, respectively, where C is the number of candidate location for a query. Therefore, the total time complexity of *FVA* is $O(2^k(C^2k) + |E|)$.

6.4 Greedy Forward Expansion Algorithm (GFA)

In *FVA* algorithm, to probe a location, we need to validate all the check-in users associated with the location one by one using the lower bound on check-ins $\delta_h^{\downarrow}$. Therefore, the *FVA* approach still needs to explore a large number of users corresponding to each location. Additionally, in *FVA*, we also need to keep the previous states to avoid missing any intermediate processes. To speed up the *CLS* process, here, we propose one greedy based forward expansion algorithm (*GFA*) that avoids the backtracking, and only progressively add locations (from L_R to L_I) if the users satisfy the minimum social constraint to the existing user set. In Section 6.4.1, we discuss the location pruning in detail by deriving the lower bound on additional check-ins by a set of users.

For a given query location l, distance threshold θ, and group size parameter k, we first identify the candidate locations (CL) that are within $(k-1)*\theta$ distance from l (refer Property 5). We further identify the candidate user set CU using Definition 25 and the m-$\widehat{core}$ components $\widehat{C}(m, CU)$ are decomposed using the subgraph CU_G induced by candidate users CU.

6.4.1 Computing Score Gains

Let, L_I be an intermediate location set. As discussed in Section 6.3.1, a participating user group should be a subset of one of the m-$\widehat{core}$ candidate user components $\widehat{C}_i \in \widehat{C}(m, CU)$. Therefore, the users of an intermediate user set should also be the subset of an m-$\widehat{core}$ candidate user component $\widehat{C}_i$. Let us assume, V_I be the intermediate user set s.t., $V_I \subseteq \widehat{C}_i$. We also assume that the total social connectivity of the graph induced by V_I is g_c and the users in V_I have h_c check-in edges to L_I. Now, we will add some new locations to the intermediate location set L_I to make the location set size k. At the same time, some of the users checked-in the new locations will be added to the intermediate user set V_I. Below, we compute the score gains of an intermediate location set when a new location is added.

Social Score Gain. Let us consider a location $l' \in L_R$ is going to be added in the current intermediate set L_I, where $L_R = nbr_\theta(L_I) \setminus L_I$. We also have considered V_I as the set of current participating users to L_I belongs to an m-$\widehat{core}$ candidate user component $\widehat{C}_i \in \widehat{C}(m, CU)$. In the process, let $V' = U[l'] \cap V(\widehat{C}_i) \setminus V_I$ be the set of potential check-in users of l' that will be added to the current user set $V_I \subseteq \widehat{C}_i$. We also assume the size of V' be r, e.g., $|V'| = r$, and Δ_g be the additional increase of the total social connectivity if the V' of l' are added to V_I. Therefore, the social score gain (w.r.t. m-$\widehat{core}$ component $\widehat{C}_i$) by adding a new location l' to the existing intermediate location group is calculated as,

$$
\begin{aligned}
\Delta S_{sc} &= \frac{g_c + \Delta_g}{(|V_I| + r)(|V_I| + r - 1)} - \frac{g_c}{(|V_I|)(|V_I| - 1)} \\
&= -\frac{g_c(|V_I|^2 + 2|V_I|r + r^2 - |V_I| - r - |V_I|^2 + |V_I|)}{(|V_I|)(|V_I| - 1)(|V_I| + r)(|V_I| + r - 1)} \\
&\qquad + \frac{\Delta_g}{(|V_I| + r)(|V_I| + r - 1)} \\
&= \frac{\Delta_g}{(|V_I| + r)(|V_I| + r - 1)} - \frac{g_c(2|V_I|r + r^2 - r)}{(|V_I| + r)(|V_I| + r - 1)(|V_I|)(|V_I| - 1)} \\
&= \frac{1}{(|V_I| + r)(|V_I| + r - 1)} \Big(\Delta_g - \frac{g_c * r * (2|V_I| + r - 1)}{(|V_I|)(|V_I| - 1)} \Big)
\end{aligned}
$$

Check-in Density Gain. Using Definition 23, we calculate the check-in density between a user set V_I and a location set L_I as, $\frac{h_c}{|V_I| * |L_I|}$, where h_c is the number of current check-in edges between V_I and L_I. If we add a new location l' to the intermediate location set L_I, the potential set of check-in users $V' = U[l'] \cap \widehat{C}_i \setminus V_I$ should be added to $V_I \subseteq \widehat{C}_i$. Therefore, if the size of V' be r, the updated check-in density will be $\frac{h_c + \Delta_h}{(|V_I| + r) * (|L_I| + 1)}$, where Δ_h is the additional increase of

the check-in edges. Hence, gain in check-in density (ΔS_{ch}) can be obtained as,

$$\begin{aligned}\Delta S_{ch} &= \frac{h_c + \Delta_h}{(|V_I| + r) * (|L_I| + 1)} - \frac{h_c}{|V_I| * |L_I|} \\ &= \frac{\Delta_h}{(|V_I| + r) * (|L_I| + 1)} - \frac{h_c(|V_I| + r + |L_I|r)}{|V_I| * |L_I| * (|V_I| + r) * (|L_I| + 1)} \\ &= \frac{1}{(|V_I| + r) * (|L_I| + 1)}\Big(\Delta_h - \frac{h_c * (|V_I| + |L_I| + r|L_I|)}{|V_I| * |L_I|}\Big)\end{aligned}$$

Total Score Gain. Based on the above formulations on social score gain and the check-in density gain, we can derive the combined score gain (Δ) using the updated location group information and the potential user set w.r.t. an m-$\widehat{core}$ component $\widehat{C}_i$,

$$\begin{aligned}\Delta =& \alpha * \Delta S_{sc} + (1 - \alpha) * \Delta S_{ch} \\ =& \frac{\alpha}{(|V_I| + r) * (|V_I| + r - 1)}\Big(\Delta_g - \frac{g_c * r * (2|V_I| + r - 1)}{|V_I| * (|V_I| - 1)}\Big) \\ &+ \frac{1 - \alpha}{(|V_I| + r) * (|L_I| + 1)}\Big(\Delta_h - \frac{h_c * (|V_I| + |L_I| + r|L_I|)}{|V_I| * |L_I|}\Big)\end{aligned} \tag{6.6}$$

A positive gain in total score, e.g., $\Delta > 0$, implies that the newly added location l' to the existing location set L_I will produce a higher co-engagement score than the current intermediate solution. In *GFA*, we expand the current solution by greedily selecting the locations from the current L_R that can generate a better score than the previous set. Therefore, from Equation 6.6 we get, $\Delta > 0$. Hence,

$$\begin{aligned}&\frac{\alpha}{(|V_I| + r) * (|V_I| + r - 1)}\Big(\Delta_g - \frac{g_c * r * (2|V_I| + r - 1)}{|V_I| * (|V_I| - 1)}\Big) \\ &+ \frac{1 - \alpha}{(|V_I| + r) * (|L_I| + 1)}\Big(\Delta_h - \frac{h_c * (|V_I| + |L_I| + r|L_I|)}{|V_I| * |L_I|}\Big) > 0 \\ \Rightarrow &\frac{\alpha}{(|V_I| + r - 1)} * \Big(\frac{g_c * r * (2|V_I| + r - 1)}{|V_I| * (|V_I| - 1)} - \Delta_g\Big) < \\ &\qquad \frac{1 - \alpha}{|L_I| + 1} * \Big(\Delta_h - \frac{h_c * (|V_I| + |L_I| + r|L_I|)}{|V_I| * |L_I|}\Big) \\ \Rightarrow &h_c\Big(1 + \frac{r|L_I|}{|V_I| * |L_I|}\Big) - \frac{\alpha}{1 - \alpha} * \frac{|L_I| + 1}{|V_I| + r - 1} * \Big(\Delta_g \\ &\qquad - \frac{g_c * r * (2|V_I| + r - 1)}{|V_I| * (|V_I| - 1)}\Big) < \Delta_h\end{aligned} \tag{6.7}$$

6.4.2 Computing upper bound of Δ_g

Let g^{max} be the maximum additional social connectivity that an intermediate user set $V_I \subseteq \widehat{C}_i$ can achieve if we add the new members $V' \in V_R \subseteq \widehat{C}_i$ of size r from V_R to V_I. If the users in V' have maximum degree $maxDeg$ w.r.t. $\widehat{C}_i$, the maximum additional social edges (e.g., social

connectivity) among V' to $V_I \cup V'$ can be obtained as,

$$g^{max} = \begin{cases} (|V_I| * r) + |E(V'_G)|, & |V_I| + r < maxDeg \\ maxDeg * r, & |V_I| + r \geq maxDeg \end{cases}$$

Here, $|E(V'_G)|$ is the number of edges of the graph induced by V'_G. Nevertheless, a new edge in a social graph increases the total social connectivity by two. Therefore, the maximum additional social connectivity due to adding r users to V_I can be increased by at most $2 * g^{max}$. Hence, we get the upper bound of Δ_g as $\Delta_g^{\uparrow} = 2 * g^{max}$.

6.4.3 Location Pruning

In this section, we will first derive the lower bound on the number of check-ins by a newly added location to the intermediate location set. Further, we will derive a lemma to prune adding some location to the intermediate location group. To deduce the lower bound on check-ins, we replace Δ_g by the upper bound of Δ_g, e.g., $\Delta_g^{\uparrow}$ in Equation 6.7. Hence, we get,

$$\begin{aligned} \Delta_h^{\downarrow} = & h_c\Big(1 + \frac{r|L_I|}{|V_I| * |L_I|}\Big) - \frac{\alpha}{1-\alpha} * \frac{|L_I|+1}{|V_I|+r-1} * \Big(2 * g^{max} \\ & - \frac{g_c * r * (2|V_I| + r - 1)}{|V_I| * (|V_I| - 1)}\Big) \end{aligned} \tag{6.8}$$

Based on the lower bound on additional check-ins $\Delta_h^{\downarrow}$, we formulate the below Lemma to identify those locations which can not increase the total score of the intermediate location set.

Lemma 10 (Location pruning)**.** *Let L_I be an intermediate location set and V_I be the intermediate user set w.r.t. an m-$\widehat{core}$ candidate user component $\widehat{C}_i$. A new location l' will not produce a higher score that the current intermediate location set, if the additional check-ins due to the newly added location l' is less than $\Delta_h^{\downarrow}$.*

Proof: Let us assume, the location l' has additional V' check-in users w.r.t. current intermediate set V_I, e.g., $V' = U[l'] \cap \widehat{C}_i \setminus V_I$. We also assume that the size of V' is r. Therefore, from Equation 6.8, we get the lower bound on additional check-ins for the newly added location l' is $\Delta_h^{\downarrow}$. Hence, if the location l' has additional check-ins less than $\Delta_h^{\downarrow}$, then l' can not produce a higher score than the current intermediate set L_I.

Based on the check-in user information of the associated users of a new location l', the above lemma will help us to prune the location from the candidate location set which can not be part of the current intermediate solution. Meanwhile, a location may be checked-in by multiple users who may belong to different m-$\widehat{core}$ candidate user component $\widehat{C}_i$. Therefore, to probe a location l', we need to consider all the m-$\widehat{core}$ candidate user components $\widehat{C}_i \in \widehat{C}(m, CU)$ where the users $U[l']$ belong to. If the additional check-ins of the users $U[l']$ w.r.t. the representative

m-$\widehat{core}$ candidate user component $\widehat{C}_i$ has less the corresponding lower bound $\Delta_h^{\downarrow}$, we will prune that location.

6.4.4 Algorithm

The *GFA* approach expands the location set iteratively by including locations that can maximize the co-engagement score of the current intermediate location set. Algorithm 6 shows the pseudo-codes of the *GFA* algorithm. The intermediate location set L_I is initialized with the query location l and the set of the candidate locations (CL) are identified using *candLoc* method. We further decompose the m-$\widehat{core}$ components using the candidate users $U[CL]$ (refer Line 2).

Algorithm 6: ForwardExpansion:$CLS(G, l, k, m, \theta)$

Input: LBSN $G = (V, E, L, E')$, group size k, degree m, distance threshold θ
Output: Top co-engaged location group of size k containing l

```
1  Initialize: L_I ← {l}, L_R ← nbr_θ(L_I)
2  CL ← candLoc(G, l, k, θ), Ĉ ← core^(m, U[CL], G)
3  while |L_I| < k and |L_R| ≥ 1 do
4      L_R ← pruneLoc(nbr_θ(L_I) ∩ Ĉ \ L_I, L_I)
5      l_top ← maxGain(L_R, L_I)
6      if l_top not Null then
7          L_I.append(l_top)
8      else
9          break;
10 if |L_I| == k then
11     C_m ← m-core^ components of graph U[L_I]_G induced by U[L_I]
12     score ← validateGroup(C_m, m)
13     return L_I
```

The code blocks between Lines 3 to 9 iteratively add new locations to the intermediate set L_I. First, we prune some locations from the θ-distance neighbor set of L_I using Lemma 10 (Line 4). We assign the remaining locations to L_R. After that, the location l_{top} among the updated set L_R having maximum total score gain (see Equation 6.6) is added to L_I (Line 7). Here, *maxGain* is used to identify the location among L_R that can produce the maximum gain in total score. Finally, when the size of the intermediate set L_I becomes k, the m-$\widehat{core}$ components of graph induced by $U[L_I]$ are identified. We further validate the m-$\widehat{core}$ components and aggregate the co-engagement scores of location set L_I w.r.t. the m-$\widehat{core}$ user user C_m (Line 12). Here, the C_m are the participating user groups to the co-engaged location set L_I.

Time Complexity. The time complexity of *GFA* algorithm is $O(C^2k + |E|)$ where C is the number of candidate locations of a query. The while loop is executed at most $O(k)$ iterations. The time complexity of the major functions are: *pruneLoc* takes $O(C^2)$ and *maxGain* takes $O(C)$. The *validateGroup* has time complexity $O(|E|)$. Therefore, the total complexity of *GFA*

is $O(C^2k + |E|)$.

6.5 Greedy Incremental Algorithm (GIA)

The main idea of the Greedy Incremental Algorithm (GIA) is to start from the query location and add the remaining $(k-1)$ location nodes iteratively to the solution based on certain *greedy* criteria. The locations among the candidate location set are added to the solution set by maximizing the chance to have at least one participating user group satisfying the minimum social cohesive constraint m. Finally, k-sized location set and the participating user groups are returned.

The incremental algorithm maintains a priority queue that stores the candidate locations to be inserted one by one in the solution set. The main difference of *GIA* with the *GFA* expansion algorithm is *GIA* selects the locations based on a set of predefined heuristic criteria to maximize the social score and the check-in scores. However, in *GFA* a *strict* social connectivity constraint was used to prune locations . Therefore, in *GFA*, we may miss some eligible users who may be part of the participating user group. Therefore, in *GIA* algorithm, we design some rule sets that provide the flexibility to select the co-engaged locations group greedily.

Rule Sets. The most important task of the Greedy Incremental Algorithm is to define the priority of the locations to be selected incrementally. To set the priority of the locations in the priority queue Q, we design the following criteria to sort the location nodes.

- Rule 1. number of common check-in users to the current solution.
- Rule 2. number of social connections between the check-in users to the participating users $U[S]$ to the current intermediate solution set S.
- Rule 3. number of check-in users.

The above three criteria are used to prioritize the locations that can generate higher check-in scores and can result in a cohesive participating user group to the selected location group. Rule 1 is to give higher priority to those locations that have a larger check-in users in common with the current solution. Adding locations using this criterion will increase the check-in score of the current solution. Further, the newly added users also have higher chance to increase the social connectivity score to the existing users. The remaining rules are used for tie breaks. Rule 2 is to give higher priority to the locations where a large number of checked-in users to the location have social edges to the participating user set w.r.t. the current intermediate solution. This criterion will increase the chance to maximize the social connectivity by adding users in further iterations that may build new social edges. Hence, the social cohesiveness and the social score of the participating user set will be improved. Rule 3 is to give a higher priority to the locations that have larger amount of check-in users.

Algorithm. The pseudo-code of the *GIA* algorithm is shown in Algorithm 7. Initially, the solution set S contains the query location l, and a priority queue Q stores the neighbor locations of l. The locations in Q are sorted using the rule sets mentioned in the previous paragraph (Section 6.5). In each iteration, the location l_{top} is popped that has highest priority among the queue Q (Line 3), and is added into the current solution S. Next, the locations within θ distance from the currently removed location node e.g., l_{top} are inserted in the priority queue Q, and at the same time, we update the priority of the nodes (in Q) w.r.t. the current solution set S (see Line 5). Finally, when the size of S becomes k, we collect all the users who have checked-in the locations in S. Now, we will validate the social constraints to the user graph induced by the checked-in users at S, and finally the maximum co-engagement score is calculated using the participating users to the m-$\widehat{core}$ components.

Algorithm 7: GreedyIncrementalSolution:$CLS(G, l, k, m, \theta)$

Input: LBSN $G = (V, E, L, E')$, group size k, degree m, distance threshold θ, minimum check-in threshold p
Output: Top co-engaged location group of size k containing l

```
1  Initialize: S ← {l}, Priority Queue Q ← nbr_θ(l), List containing participant user group 𝕌 ← ∅
2  while |S| < k do
3      l_top ← Q.pop(); S.append(l_top)
4      if |nbr_θ(l_top)| > 0 then
5          Q ← update(Q ∪ nbr_θ(l_top), S) (using the rule sets provided in Section 6.5)
6      else
7          return null;
8  if |S| == k then
9      C_m ← m-core components graph U[S]_G induced by S
10     for each m-core component C in C_m do
11         flag, score ← validateGroup(C, k)
12         if flag = true and score > bestScore then
13             S ← C.nodes; bestScore > score; break;
```

Time Complexity. The *GIA* algorithm takes $O((|V| + k + |E'|)log(|V| + k))$ time for updating and maintaining the priority queue. The time complexity of *validateGroup* function is $O(|E|)$). Thus, the time complexity of *GIA* algorithm is $O((|V| + k + |E'|)log(|V| + k) + |E|)$.

6.6 Experiments

We evaluate the proposed algorithms using three real-world location-based social networks. All the experiments were carried out using Windows 10 desktop environment with 64GB RAM and 3.40GHz Intel i7 CPU.

Table 6.2: Dataset Statistics

Dataset	#Users	#Edges	#Check-ins	#Places	ACU	ACP
Gowalla	196,591	456,830	6,442,892	1,280,969	32.77	5.03
Brightkite	58,228	197,167	4,747,281	772,783	81.53	6.14
Yelp	270,323	3,827,002	5,425,778	192,609	20.07	28.17

6.6.1 Experimental Setup

Datasets.

Table 6.2 shows the statistics about the datasets used in our experiments. The Gowalla and Brightkite datasets were collected from [83]. Both the datasets contain social connection, and the check-in information are available over the period Feb. 2009 - Oct. 2010 and Apr. 2008 - Oct. 2010 respectively. The third dataset Yelp[1] was collected from the official Yelp Dataset Challenge Round 13, Year 2019. This dataset contains friendship network and location information in the form of location-tags in users' tips and reviews. If a user does not have any check-ins, we discard the user along with her social edges from the network. Among the three datasets, Yelp has the maximum number of social connections, whereas the Brightkite dataset has the maximum average number of check-ins. The columns 'ACU' and 'ACP' in Table 6.2 denote the average check-ins by users and average number of check-ins at each place in the datasets.

Query Location Selection.

To obtain the query locations we first generate the result of k-core decomposition in the social network with minimum degree m. After that, we collect all the check-in locations of the user set available in the k-core subgraph. We further randomly select a set of 10% of the locations having checked-in by at least 50 users among the k-core subgraph. The intuition behind selecting the query locations having a large number of check-ins by the k-core members is, if many people check-in a place from the same community, the other locations near to the highly checked-in place have the higher chance to be visited by a large number of socially connected friends.

Parameter Settings.

The default parameter values and their ranges are given in Table 6.3. If there is no specific declaration about a parameter value, we will select the default values as underlined in their corresponding ranges.

Algorithms.

To the best of our knowledge, the problem of selecting top co-engaged location groups from LBSNs has no direct competitor in the literature. Therefore, we compare our proposed algorithms

[1]https://www.yelp.com/dataset/

Table 6.3: Parameters and their value ranges

Parameter	Values	Description
m	$\underline{5}, 10, 15, 20$	minimum degree threshold
θ	$1, \underline{2}, 3, 4, 5$	maximum distance threshold (KM)
k	5, $\underline{10}$, 15, 20	location group size

using different parameter settings in our experiment.

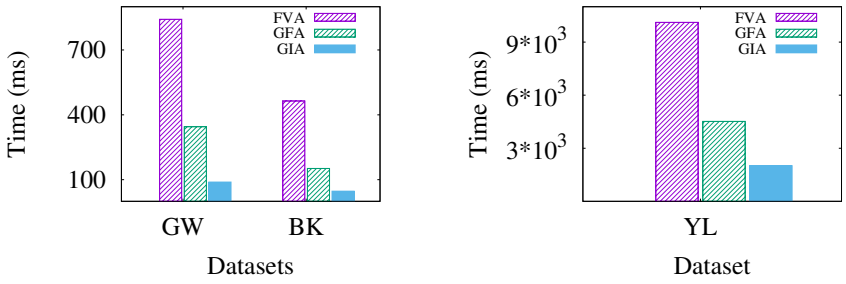
The efficiency of the proposed algorithms is measured using running time. Similarly, we use the co-engagement score metric to evaluate the quality of the location group returned by the algorithms. The co-engagement score of a location group measures the social connectivity of the locations and the check-in density of the user groups w.r.t. the location set returned by the proposed algorithms. We calculate co-engagement score of a location group using Equation 6.2.

6.6.2 Experimental Results and Discussions

Performance on different datasets. We first compare the performance of our proposed algorithms on the three datasets, e.g, Gowalla (GW), Brightkite (BK), and Yelp (YL), mentioned in Table 6.2. Here, we set the default parameters as $m = 5$, $\theta = 2KM$, $k = 10$. For each case, we report the average of 10 different queries.

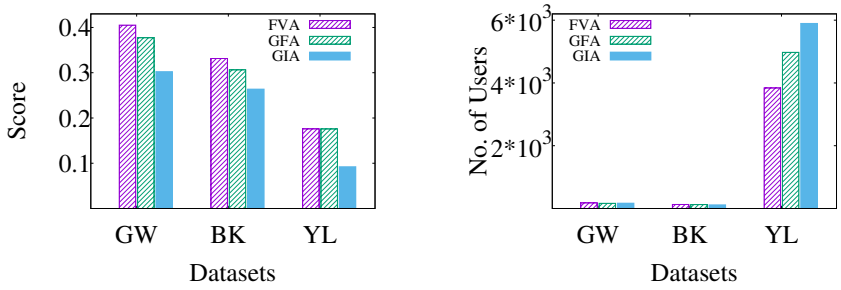
The Figure 6.2 shows the average running time of the algorithms. Clearly, the algorithms consume maximum time to execute in Yelp dataset and faster in Brightkite. For example, in *GIA*, the Brightkite dataset is 42 times faster than Yelp. This is because, in Yelp, the majority of the query nodes are checked-in by a large number of users. We also compare the co-engagement score and the participating user group size in Figure 6.3(a) and Figure 6.3(b), respectively. From the figures, we can see that *FVA* algorithm has achieved the highest co-engagement score in each dataset. Comparing the greedy solutions, e.g., *GFA* and *GIA*, the *GFA* algorithm has higher score than *GIA* approach. However, in terms of running time, *GIA* algorithm is more than five times faster than the *GFA* (refer Figure 6.2). This is because, the *GIA* approach greedily selects the top location based on the certain rule set, and adds them to the solution set without validating. Whereas, the forward expansion algorithm *GFA* adds the location that creates the maximum score to the intermediate set. Therefore, we need to calculate the score for each neighbor's locations and sort them based on the score. This consumes more time than the *GIA* algorithm. In terms of the number of average participating users to the selected location groups, the *FVA* returns a higher value than the other two approaches. This is because, *FVA* searches all the k-sized connected location combinations and returns the location set that produces a higher co-engagement score. We also observe that the average number of participating users in the Yelp is much more than the Gowalla and Brightkite datasets. For example, in Yelp dataset, the *GFA* algorithm returns user group size 21 times more than the Gowalla dataset.

Varying location group size k. This set of experiments is to study the effect of the location



(a) Running time in GW and BK (b) Running time in YL

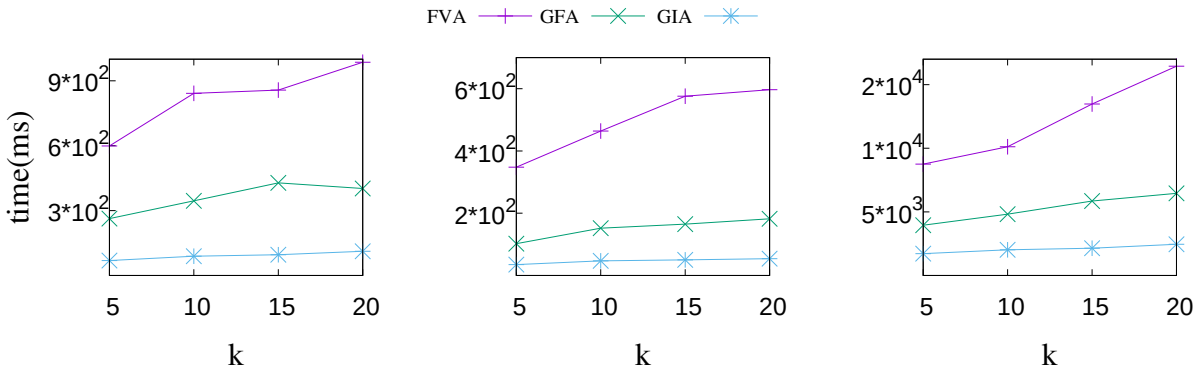
Figure 6.2: Average running time in default configuration



(a) Co-engagement score comparison (b) User size comparison

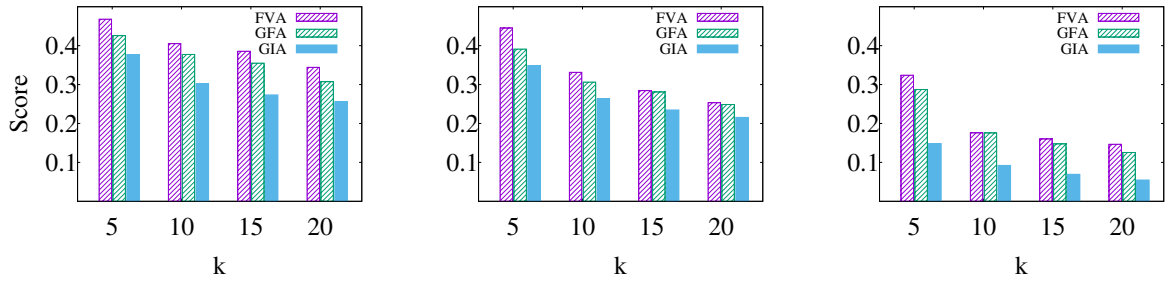
Figure 6.3: Average Co-Engagement score and participating user size in default configuration

group size k on the performance. Figure 6.4, Figure 6.5, and Figure 6.6 show the running time of the algorithms, co-engagement scores of the answer set, and the number of participating users, respectively when the value of k varies from 5 to 20. In each dataset, we observe a similar trend of the proposed approaches. The *FVA* algorithm has a much higher total co-engagement score than the other two approaches. The running time of *FVA* increases very fast with the location group size. This is because, the *FVA* algorithm needs to compare all the possible groups of size k containing the query location, and when k increases, the algorithm requires more time to execute. The running time of *GIA* and *GFA* increase almost linearly with the size of k. We also notice the user group size grows steadily in each dataset when k increases.



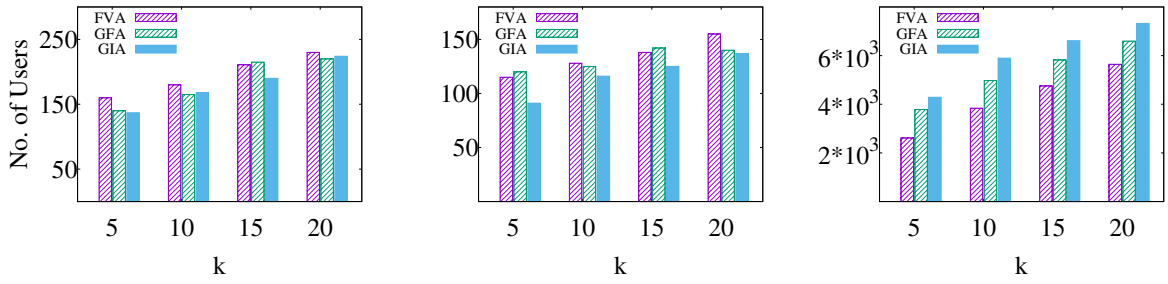
(a) Gowalla (b) Brightkite (c) Yelp

Figure 6.4: Running time when k varies



(a) Gowalla (b) Brightkite (c) Yelp

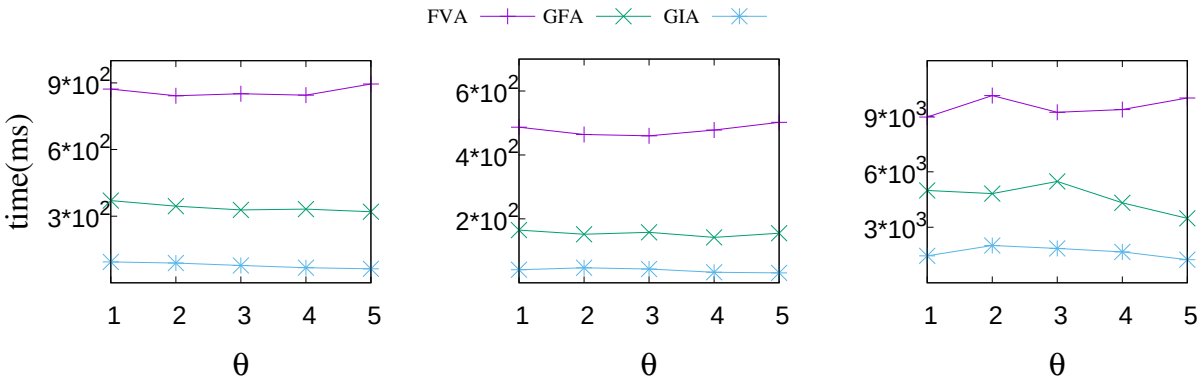
Figure 6.5: Co-Engagement score when k varies



(a) Gowalla (b) Brightkite (c) Yelp

Figure 6.6: Participating user size when k varies

Varying the distance threshold θ. We study the effect of the maximum distance threshold on the performance of the proposed algorithms. In Figure 6.7, we observe that the greedy-based methods have almost constant running time when the distance threshold θ increases from 1 to 5. In each dataset, the three algorithms have similar trends of execution time, where *FVA* is on average two to three times slower than *GFA*.

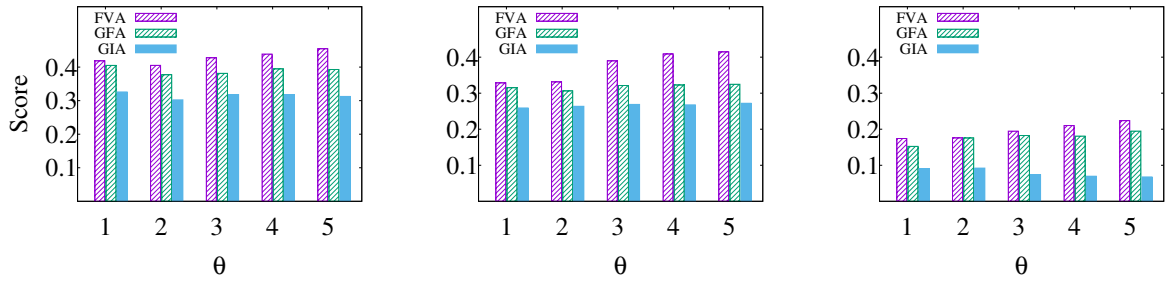


(a) Gowalla (b) Brightkite (c) Yelp

Figure 6.7: Running time when θ varies

Figure 6.8 shows the co-engagement scores of the location groups returned by the algorithms in each dataset. The Yelp dataset has the lowest score than the other two datasets. This is because the participating user group size in Yelp is much higher than Gowalla and Brightkite datasets (refer Figure 6.9), which reduces the social score of the participating user set. In this

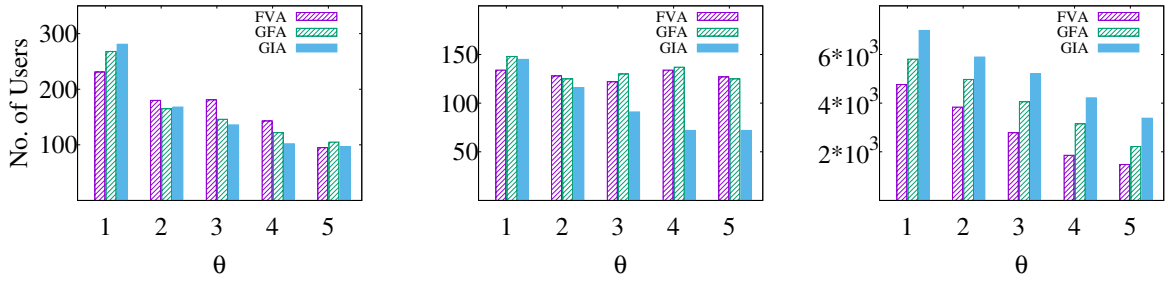
case, the *GIA* algorithm produces the lowest co-engagement score. The co-engagement score of *FVA* and *GFA* increase with the value of θ. This is because, the number of locations in the search space increases automatically with the increase of distance threshold parameter value θ. Therefore, all the algorithms need to explore more location points to form a location group. This also increases the chance to have a better set of participating user groups to the locations.



(a) Gowalla (b) Brightkite (c) Yelp

Figure 6.8: Co-Engagement score when θ varies

The Figure 6.9 shows the participating user size when θ varies. We notice in *GIA* the number of users decreases linearly when θ increases. This is because, *GIA* incrementally adds locations to the group based on the rule set mentioned in Section 6.5. Therefore, when θ increases, the *GIA* algorithm gets more options to select the locations that have common check-ins to the current user set, which reduces the participating user set size.

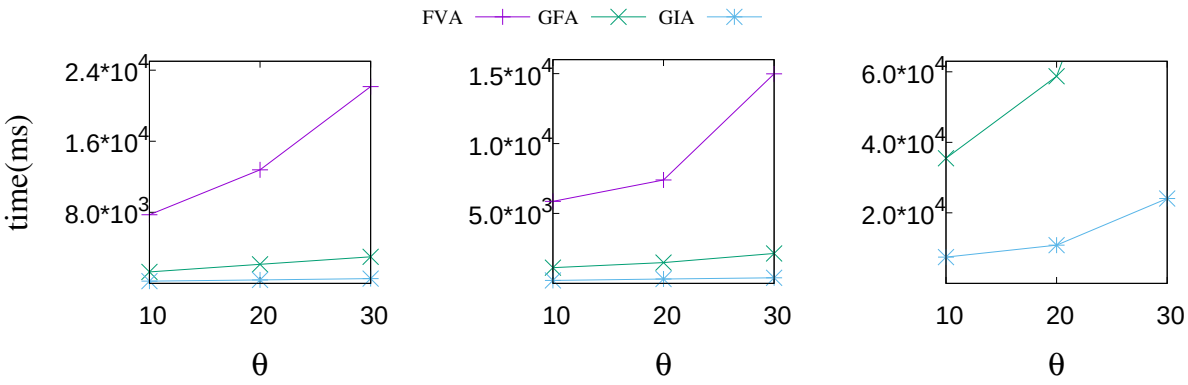


(a) Gowalla (b) Brightkite (c) Yelp

Figure 6.9: Participating user size when θ varies

Scalability. To demonstrate the scalability of our algorithms, we consider the location group size with a higher value $k = 40$ and vary the distance threshold θ with a higher range from 10 to 30 KM. Therefore, for a given location as query, the search space will automatically increase w.r.t. a query location. Nevertheless, we also need to consider a large number of check-in users with more check-in edges when the spatial space increases. Figure 6.10 shows the running time of the three algorithms. We observe that the running time of *FVA* increases very fast in each dataset when the distance threshold increases from 10 to 30 KM.

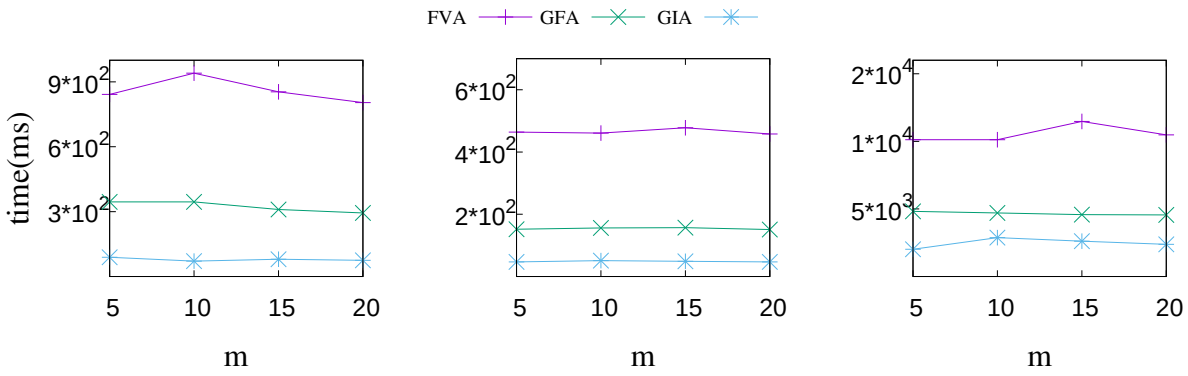
Effect of other parameters. *(1) Varying minimum degree threshold m:* Figure 6.11 shows the running time of the algorithms when minimum degree threshold m varies from 5 to 20. The



(a) Gowalla (b) Brightkite (c) Yelp

Figure 6.10: Scalability analysis by varying θ, $k = 40$

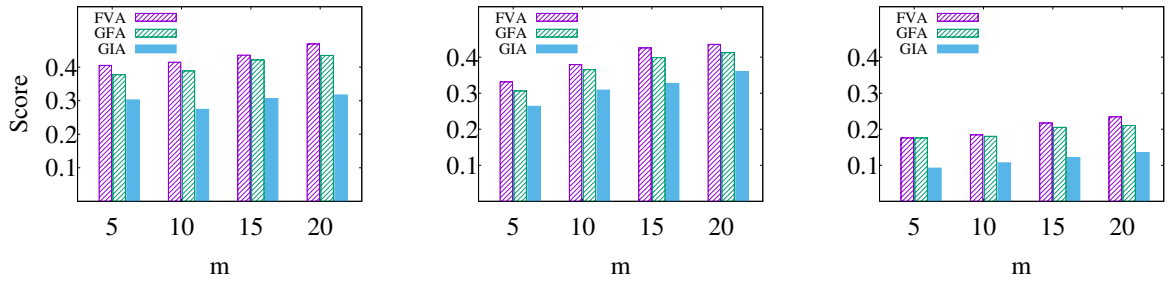
number of friends of each user in a social subgraph increases with m. However, in some cases, we may not find any participating user group satisfying a higher degree threshold m. Therefore, we observe that the average running time decreases a little in *FVA* since many user nodes are pruned in social subgraphs that do not satisfy the degree constraints. In Figure 6.12, we also notice that the co-engagement scores of the algorithms increase linearly with the value of m in each dataset.



(a) Gowalla (b) Brightkite (c) Yelp

Figure 6.11: Running time when m varies

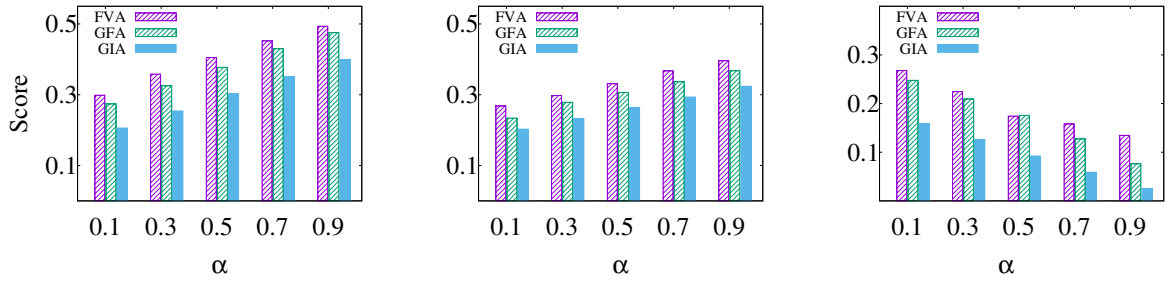
(2) Varying trade-off parameter α: We do not observe noticeable changes in running time when the value of α increases. This is because the parameter α does not change the search space size. The parameter α is only used to calculate the scores while pruning or selecting locations. Therefore, the runtime of the algorithms remains almost constant with the size of the participating user set. In Figure 6.13, we show the co-engagement scores of the algorithms by varying the trade-off parameter α from 0.1 to 0.9. In Gowalla and Brightkite datasets, the co-engagement scores gradually increase with α. For example, the average co-engagement score of the *FVA* algorithm increases by 0.3 when the value of α is raised from 0.1 to 0.9. Interestingly, the scores returned by the algorithms decline in Yelp dataset with the increase of α. This is because, the average check-ins per location in the Yelp dataset is very high than the other



(a) Gowalla (b) Brightkite (c) Yelp

Figure 6.12: Co-Engagement score when m varies

two datasets. Therefore, when α increases, the average check-in density score in Yelp drops significantly.



(a) Gowalla (b) Brightkite (c) Yelp

Figure 6.13: Co-Engagement score when α varies

Discussions. The *FVA* algorithm achieves higher co-engagement score than *GEA* and *GIA*. In majority of the cases, the *GIA* approach generates results with comparable quality with *GEA*, while *GIA* is much faster than the *GEA* approach. However, as the *GIA* algorithm heuristically selects the next location based on certain rule sets, in some of the cases we may not find any participating user group satisfying the social constraint.

6.7 Summary

In the previous chapter, we identified the top locations for each social user in a network. The selected locations of each user were both socially and spatially co-engaged w.r.t. the friends. However, the user level selection of the co-engaged locations in a large social network may not be useful to the enterprise real-time applications where group of co-engaged locations are important for their businesses. Therefore, in this chapter, we have formulated the problem on Co-engaged Location Group Selection (*CLS*) that aims to find a set of locations, such that the selected locations are highly co-engaged to socially cohesive user groups. To incorporate the trade-offs among different socio-spatial factors, we formulate a scoring function to measure the co-engagement of the social user to a location set. The scoring function is devised by combining

the social connectivity and check-in density. To effectively process the *CLS* query, we propose three different solutions. We also have conducted extensive empirical studies on large-scale real-world location-based social network datasets to demonstrate the effectiveness and efficiency of our proposed algorithms.

Chapter 7

Conclusions and Future Works

In this chapter, the primary research contributions of this thesis are summarized. We also discuss some interesting future direction in socio-spatial research domain can be further explored.

7.1 Conclusion

In this thesis, we have investigated the social and spatial properties of locations in large social networks. More specifically, we aim to effectively utilize the socio-spatial information of a network to support real-time applications where the location information of users are crucial. We have discussed various aspects of modeling the relationships between user mobility and social connections to explore the following research objectives in this thesis: (1) Inferring locations of unlabeled social users by exploiting various implicit information available in the network; (2) Selecting socially and spatially relevant but diverse location set for each individual in the network; (3) Search for location groups that are highly co-engaged and personalized to social users as a group.

Towards objective (1), we have studied the existing location prediction models and provided an in-depth empirical comparison of eight representative prediction models using five metrics on four real-world large-scale datasets, namely Twitter, Gowalla, Brightkite, and Foursquare. We formulate a generalized procedure-oriented location prediction framework that allows us to evaluate and compare the prediction models systematically and thoroughly under extensive experimental settings. Based on our results, we perform a detailed analysis of the merits and limitations of the existing models by providing significant insights into the location prediction problem. We also propose a method to infer the activity location of social users using the implicit information of other socially connected users in the network. Our proposed approach can estimate activity location of a user in LBSNs by propagating the spatial information through the friendship edges by maintaining an inference sequence. We find that the proposed method significantly improves the state-of-the-art network-based location inference techniques in terms of accuracy.

Towards objective (2), we have explored a new problem on top-k socio-spatial co-engaged location selection for each social user in a social graph, that selects the best set of k locations from a large number of location candidates relating to the user and her friends. The selected locations should be (i) spatially and socially relevant to the user and her friends, and (ii) diversified in both spatially and socially to maximize the coverage of friends in the spatial space. To address the problem, we first develop an `Exact` solution by designing various pruning strategies based on the derived bounds on diversity. To make the solution scalable for large datasets, we also develop an approximate solution by deriving the relaxed bounds and advanced termination rules to filter out insignificant intermediate results. To further accelerate the efficiency, we present one fast exact approach and a meta-heuristic approximate approach by avoiding the repeated computation of diversity at the running time. Finally, we have conducted extensive experiments to evaluate the performance of our proposed algorithms against the adapted existing methods using real-world LBSN datasets.

Towards objective (3), we have studied the problem of co-engaged location group search in geo-social networks. For a given query location, the co-engaged location group search problem aims to find a k-sized location set (containing the query location) which are highly co-engaged w.r.t. cohesive participant user groups. To solve this problem, we first propose Filter-and-Verify algorithm that can effectively filter out the ineligible locations and their check-in user, and further can terminate the search process using bound on check-in counts. We also propose a heuristic based expansion algorithm that can effectively select the intermediate locations and their participating users based on the strict social connectivity. Further, we design a greedy algorithm that incrementally adds locations to the solution using certain greedy criteria. Finally, we measure the effectiveness and efficiency of our proposed solutions by conducting extensive experiments on real-world datasets.

7.2 Future Works

Advanced location inference model. The traditional network-based location inference models majorly focus on node features and their relationships, and ignore various latent edge features available in the graphs. On the other hand, existing other location inference models designed for graph learning (e.g., GCN) mainly focus on the node properties, and utilizes the edge connections derived from the existing relationships between nodes. Such graph-learning based models ignore the latent edge features available in the graphs which are usually amalgamated with the datasets properties and the socio-spatial behavior of the users. In geo-social network, edges are often in possession of rich and multi-modal information, where the edge relationships should be effectively exploited to model a complex system. Meanwhile, in heterogeneous geo-social network, there may exists different types of relationships between the nodes, e.g., user-user, user-location, location-location. Therefore, due to the complex structure of such

heterogeneous geo-social graph, it is challenging to find a fascinating way to incorporate the different information sources into a model. Hence, a unified graph learning model is essential to aggregate the heterogeneous relationships in the networks. It is also important to exploit the rich source of multidimensional edge features for the user location inference tasks.

Time-variant location set selection. The location check-in patterns of social users can be modeled using the temporal information available in the check-ins. The preference for interesting locations to a user or cohesive social group changes with time. Therefore, it will be promising to develop some system that can continuously update the set of top locations from the network which are active currently to a individual or different social groups. The selected locations will carry more sophisticated information about the participant user groups. As an extension of this future work, we also can provide more meaningful explanations about the types of the selected locations using the social, spatial, and temporal properties of the locations that users visit frequently. Therefore, development of real-time system for analyzing the temporal interactions of users in various Point-of-Interests in Location-based Social Networks is an interesting research direction.

Bibliography

[1] G. Agarwal and D. Kempe. Modularity-maximizing graph communities via mathematical programming. *The European Physical Journal B*, 66(3):409–418, 2008.

[2] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *ACM WSDM*, pages 5–14, 2009.

[3] R. Ahuja, N. Armenatzoglou, D. Papadias, and G. J. Fakas. Geo-social keyword search. In *International Symposium on Spatial and Temporal Databases*, pages 431–450. Springer, 2015.

[4] O. Ajao, J. Hong, and W. Liu. A survey of location inference techniques on twitter. *Journal of Information Science*, 41(6):855–864, 2015.

[5] A. Angel and N. Koudas. Efficient diversity-aware search. In *Proceedings of ACM SIGMOD*, pages 781–792. ACM, 2011.

[6] J. Ao, P. Zhang, and Y. Cao. Estimating the locations of emergency events from twitter streams. *Procedia Computer Science*, 31:731–739, 2014.

[7] N. Armenatzoglou, R. Ahuja, and D. Papadias. Geo-social ranking: functions and query processing. *The VLDB Journal*, 24(6):783–799, 2015.

[8] N. Armenatzoglou, S. Papadopoulos, and D. Papadias. A general framework for geo-social query processing. *Proceedings of the VLDB Endowment*, 6(10):913–924, 2013.

[9] 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*, pages 61–70. ACM, 2010.

[10] J. Bao, Y. Zheng, and M. F. Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *SIGSPATIAL GIS*, pages 199–208, 2012.

[11] J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel. Recommendations in location-based social networks: a survey. *GeoInformatica*, 19(3):525–565, 2015.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[13] H. Bo, P. Cook, and T. Baldwin. Geolocation prediction in social media data by finding location indicative words. In *Proceedings of COLING*, pages 1045–1062, 2012.

[14] A. Borodin, H. C. Lee, and Y. Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *Proceedings of PODS*, pages 155–166, 2012.

[15] T. Cai, J. Li, A. S. Mian, T. Sellis, J. X. Yu, et al. Target-aware holistic influence maximization in spatial social networks. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[16] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st ACM SIGIR*, pages 335–336, 1998.

[17] I. Catallo, E. Ciceri, P. Fraternali, D. Martinenghi, and M. Tagliasacchi. Top-k diversity queries over bounded regions. *ACM TODS*, 38(2):10, 2013.

[18] H.-w. Chang, D. Lee, M. Eltaher, and J. Lee. @ phillies tweeting from philly? predicting twitter user locations with spatial word usage. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pages 111–118. IEEE Computer Society, 2012.

[19] J. Chen, Y. Liu, and M. Zou. From tie strength to function: Home location estimation in social network. In *Computing, Communications and IT Applications Conference (ComComAp), 2014 IEEE*, pages 67–71. IEEE, 2014.

[20] L. Chen, C. Liu, R. Zhou, J. Li, X. Yang, and B. Wang. Maximum co-located community search in large scale social networks. *Proceedings of the VLDB Endowment*, 11(10):1233–1246, 2018.

[21] L. Chen, C. Liu, R. Zhou, J. Xu, J. X. Yu, and J. Li. Finding effective geo-social group for impromptu activities with diverse demands. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 698–708, 2020.

[22] Y. Chen, Y. Fang, R. Cheng, Y. Li, X. Chen, and J. Zhang. Exploring communities in large profiled graphs. *IEEE Transactions on Knowledge and Data Engineering*, 31(8):1624–1629, 2018.

[23] Y. Chen, J. Zhao, X. Hu, X. Zhang, Z. Li, and T.-S. Chua. From interest to function: Location estimation in social media. In *AAAI*, 2013.

[24] 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*, pages 759–768. ACM, 2010.

[25] Z. Cheng, J. Caverlee, and K. Lee. A content-driven framework for geolocating microblog users. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1):2, 2013.

[26] 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*, pages 1082–1090. ACM, 2011.

[27] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Buttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of ACM SIGIR*, pages 659–666. ACM, 2008.

[28] A. Clauset, M. E. Newman, and C. Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[29] R. Compton, D. Jurgens, and D. Allen. Geotagging one hundred million twitter accounts with total variation minimization. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 393–401. IEEE, 2014.

[30] W. Cui, Y. Xiao, H. Wang, Y. Lu, and W. Wang. Online search of overlapping communities. In *Proceedings of the 2013 ACM SIGMOD international conference on Management of data*, pages 277–288, 2013.

[31] W. Cui, Y. Xiao, H. Wang, and W. Wang. Local search of communities in large graphs. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 991–1002, 2014.

[32] A. Das Sarma, H. Lee, H. Gonzalez, J. Madhavan, and A. Halevy. Efficient spatial sampling of large geographical tables. In *ACM SIGMOD*, pages 193–204. ACM, 2012.

[33] 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.

[34] E. M. Delmelle. Spatial sampling. *Handbook of regional science*, pages 1385–1399, 2014.

[35] T. H. Do, D. M. Nguyen, E. Tsiligianni, B. Cornelis, and N. Deligiannis. Multiview deep learning for predicting twitter users' location. *preprint arXiv:1712.08091*, 2017.

[36] M. Drosou and E. Pitoura. Diversity over continuous data. *IEEE Data Eng. Bull.*, 32(4):49–56, 2009.

[37] M. Drosou and E. Pitoura. Disc diversity: result diversification based on dissimilarity and coverage. *Proceedings of the VLDB Endowment*, 6(1):13–24, 2012.

[38] M. Drosou and E. Pitoura. Dynamic diversification of continuous data. In *Proceedings of the 15th International Conference on Extending Database Technology*, pages 216–227, 2012.

[39] M. Drosou and E. Pitoura. Diverse set selection over dynamic data. *IEEE TKDE*, 26(5):1102–1116, 2014.

[40] R. Dunn and A. Harrison. Two-dimensional systematic sampling of land use. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 42(4):585–601, 1993.

[41] M. Ebrahimi, E. ShafieiBavani, R. Wong, and F. Chen. A unified neural network model for geolocating twitter users. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 42–53, 2018.

[42] E. Elhamifar and M. Clara De Paolis Kaluza. Online summarization via submodular and convex optimization. In *Proceedings of IEEE CVPR*, pages 1783–1791, 2017.

[43] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[44] Y. Fang, R. Cheng, X. Li, S. Luo, and J. Hu. Effective community search over large spatial graphs. *Proceedings of the VLDB Endowment*, 10(6):709–720, 2017.

[45] Y. Fang, R. Cheng, S. Luo, and J. Hu. Effective community search for large attributed graphs. *Proceedings of the VLDB Endowment*, 9(12):1233–1244, 2016.

[46] T. Fornaciari and D. Hovy. Dense node representation for geolocation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 224–230, 2019.

[47] T. Fornaciari and D. Hovy. Geolocation with attention-based multitask learning models. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 217–223, 2019.

[48] S. Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

[49] P. Fraternali, D. Martinenghi, and M. Tagliasacchi. Top-k bounded diversification. In *ACM SIGMOD*, pages 421–432. ACM, 2012.

[50] H. Gao, J. Tang, and H. Liu. Exploring social-historical ties on location-based social networks. In *International AAAI Conference on Weblogs and Social Media*, 2012.

[51] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 29. wh freeman New York, 2002.

[52] V. Gaul and S. Baul. Location-based services market statistics and forecast. https://www.alliedmarketresearch.com/location-based-services-market, 2019 (accessed November, 2019).

[53] J. Gelernter and S. Balaji. An algorithm for local geoparsing of microtext. *GeoInformatica*, 17(4):635–667, 2013.

[54] B. Ghosh, M. E. Ali, F. M. Choudhury, S. H. Apon, T. Sellis, and J. Li. The flexible socio spatial group queries. *Proceedings of the VLDB Endowment*, 12(2):99–111, 2018.

[55] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[56] K. Golenberg, B. Kimelfeld, and Y. Sagiv. Keyword proximity search in complex data graphs. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 927–940, 2008.

[57] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *WWW*, pages 381–390, 2009.

[58] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[59] Y. Gu, J. Song, W. Liu, and L. Zou. Hlgps: A home location global positioning system in location-based social networks. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 901–906. IEEE, 2016.

[60] T. Guo, X. Cao, and G. Cong. Efficient algorithms for answering the m-closest keywords query. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 405–418, 2015.

[61] T. Guo, K. Feng, G. Cong, and Z. Bao. Efficient selection of geospatial data on maps for interactive and visualized exploration. In *ACM SIGMOD*, pages 567–582. ACM, 2018.

[62] R. Haining. Estimating spatial means with an application to remotely sensed data. *Communications in Statistics-Theory and Methods*, 17(2):573–597, 1988.

[63] N. A. H. Haldar, J. Li, M. Reynolds, T. Sellis, and J. X. Yu. Location prediction in large-scale social networks: an in-depth benchmarking study. *The VLDB Journal*, pages 1–26, 2019.

[64] B. Han, P. Cook, and T. Baldwin. A stacking-based approach to twitter user geolocation prediction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 7–12, 2013.

[65] F. Hao, G. Min, Z. Pei, D.-S. Park, and L. T. Yang. k-clique community detection in social networks based on formal concept analysis. *IEEE Systems Journal*, 11(1):250–259, 2015.

[66] 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*, pages 237–246. ACM, 2011.

[67] M. R. Henzinger and V. King. Randomized fully dynamic graph algorithms with polylogarithmic time per operation. *Journal of the ACM (JACM)*, 46(4):502–516, 1999.

[68] B. Huang and K. M. Carley. A hierarchical location prediction neural network for twitter user geolocation. *arXiv preprint arXiv:1910.12941*, 2019.

[69] X. Huang and L. V. Lakshmanan. Attribute-driven community search. *Proceedings of the VLDB Endowment*, 10(9):949–960, 2017.

[70] X. Huang, L. V. Lakshmanan, J. X. Yu, and H. Cheng. Approximate closest community search in networks. *arXiv preprint arXiv:1505.05956*, 2015.

[71] M. Hulden, M. Silfverberg, and J. Francom. Kernel density estimation for text-based geolocation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*. Citeseer, 2015.

[72] J. Illenberger, K. Nagel, and G. Flötteröd. The role of spatial interaction in social networks. *Networks and Spatial Economics*, 13(3):255–282, 2013.

[73] M. Islam, M. E. Ali, Y.-B. Kang, T. Sellis, F. M. Choudhury, et al. Keyword aware influential community search in large attributed graphs. *arXiv preprint arXiv:1912.02114*, 2019.

[74] D. Jurgens. That's what friends are for: Inferring location in online social media platforms based on social relationships. *ICWSM*, 13:273–282, 2013.

[75] D. Jurgens, T. Finethy, J. McCorriston, Y. T. Xu, and D. Ruths. Geolocation prediction in twitter using social networks: A critical analysis and review of current practice. *Icwsm*, 15:188–197, 2015.

[76] P. K. Kefaloukos, M. V. Salles, and M. Zachariasen. Declarative cartography: In-database map generalization of geospatial datasets. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 1024–1035. IEEE, 2014.

[77] J. Kim, T. Guo, K. Feng, G. Cong, A. Khan, and F. M. Choudhury. Densely connected user community and location cluster search in location-based social networks. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 2199–2209, 2020.

[78] L. Kong, Z. Liu, and Y. Huang. Spot: Locating social media users based on social network context. *Proceedings of the VLDB Endowment*, 7(13):1681–1684, 2014.

[79] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *Proceedings of ACM SIGIR*, pages 195–202, 2009.

[80] A. Lancichinetti and S. Fortunato. Limits of modularity maximization in community detection. *Physical review E*, 84(6):066122, 2011.

[81] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

[82] K. Lee, R. K. Ganti, M. Srivatsa, and L. Liu. When twitter meets foursquare: tweet location prediction using foursquare. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 198–207. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

[83] J. Leskovec and A. Krevl. Snap datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[84] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel. Lars: A location-aware recommender system. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 450–461. IEEE, 2012.

[85] C. Li and A. Sun. Fine-grained location extraction from tweets with temporal awareness. In *Proceedings of the SIGIR conference on Research & development in information retrieval*, pages 43–52. ACM, 2014.

[86] C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B.-S. Lee. Twiner: named entity recognition in targeted twitter stream. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 721–730. ACM, 2012.

[87] J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, and F. Xia. Community-diversified influence maximization in social networks. *Information Systems*, page 101522, 2020.

[88] J. Li, X. Wang, K. Deng, X. Yang, T. Sellis, and J. X. Yu. Most influential community search over large social networks. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 871–882. IEEE, 2017.

[89] R. Li, S. Wang, and K. C.-C. Chang. Multiple location profiling for users and relationships from social network and content. *Proceedings of the VLDB Endowment*, 5(11):1603–1614, 2012.

[90] R. Li, S. Wang, H. Deng, R. Wang, and K. C.-C. Chang. Towards social user profiling: unified and discriminative influence model for inferring home locations. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1023–1031. ACM, 2012.

[91] R.-H. Li, L. Qin, J. X. Yu, and R. Mao. Influential community search in large networks. *Proceedings of the VLDB Endowment*, 8(5):509–520, 2015.

[92] R.-H. Li, L. Qin, J. X. Yu, and R. Mao. Finding influential communities in massive networks. *The VLDB Journal*, 26(6):751–776, 2017.

[93] W. Li, P. Serdyukov, A. P. de Vries, C. Eickhoff, and M. Larson. The where in the tweet. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2473–2476. ACM, 2011.

[94] J. Lingad, S. Karimi, and J. Yin. Location extraction from disaster-related microblogs. In *Proceedings of the 22nd international conference on world wide web*, pages 1017–1020. ACM, 2013.

[95] W. Liu, W. Sun, C. Chen, Y. Huang, Y. Jing, and K. Chen. Circle of friend query in geo-social networks. In *International Conference on Database Systems for Advanced Applications*, pages 126–137. Springer, 2012.

[96] X. Liu, S. Zhang, F. Wei, and M. Zhou. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 359–367. ACL, 2011.

[97] Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-link lda: joint models of topic and author community. In *proceedings of the 26th annual international conference on machine learning*, pages 665–672, 2009.

[98] I. Lourentzou, A. Morales, and C. Zhai. Text-based geolocation prediction of social media users with neural networks. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 696–705. IEEE, 2017.

[99] M. Mahdian, O. Schrijvers, and S. Vassilvitskii. Algorithmic cartography: Placing points of interest and ads on maps. In *Proceedings of ACM SIGKDD*, pages 755–764. ACM, 2015.

[100] J. Mahmud, J. Nichols, and C. Drews. Where is this tweet from? inferring home locations of twitter users. *ICWSM*, 12:511–514, 2012.

[101] J. McGee, J. Caverlee, and Z. Cheng. Location prediction in social media based on tie strength. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 459–468. ACM, 2013.

[102] E. Minack, W. Siberski, and W. Nejdl. Incremental diversification for very large sets: a streaming-based approach. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 585–594, 2011.

[103] Y. Miura, M. Taniguchi, T. Taniguchi, and T. Ohkuma. Unifying text, metadata, and user network representations with a neural network for geolocation prediction. In *Proceedings of the 55th Annual Meeting of the ACL*, volume 1, pages 1260–1272, 2017.

[104] D. Mok, B. Wellman, and J. Carrasco. Does distance matter in the age of the internet? *Urban Studies*, 47(13):2747–2783, 2010.

[105] F. Morstatter, H. Gao, and H. Liu. Discovering location information in social media. *IEEE Data Eng. Bull.*, 38(2):4–13, 2015.

[106] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen. Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 542–550, 2008.

[107] M. E. Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004.

[108] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[109] J. Neyman. On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society*, 97(4):558–625, 1934.

[110] J. Neyman. Contribution to the theory of sampling human populations. *Journal of the American Statistical Association*, 33(201):101–116, 1938.

[111] R. T. Ng and J. Han. Clarans: A method for clustering objects for spatial data mining. *IEEE transactions on knowledge and data engineering*, 14(5):1003–1016, 2002.

[112] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. Mining user mobility features for next place prediction in location-based services. In *2012 IEEE 12th international conference on data mining*, pages 1038–1043. IEEE, 2012.

[113] S. Nutanong, M. D. Adelfio, and H. Samet. Multiresolution select-distinct queries on large geographic point sets. In *Proceedings ACM SIGSPATIAL GIS*, pages 159–168, 2012.

[114] S. M. Omohundro. *Five balltree construction algorithms*. International Computer Science Institute Berkeley, 1989.

[115] J. Pang and Y. Zhang. Deepcity: A feature learning framework for mining location check-ins. In *Eleventh AAAI Conference on Web and Social Media*, 2017.

[116] M. J. Paul and M. Dredze. You are what you tweet: Analyzing twitter for public health. *Icwsm*, 20:265–272, 2011.

[117] S. Peng, H. Samet, and M. D. Adelfio. Viewing streaming spatially-referenced data at interactive rates. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 409–412. ACM, 2014.

[118] R. Priedhorsky, A. Culotta, and S. Y. Del Valle. Inferring the origin locations of tweets with quantitative confidence. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 1523–1536, 2014.

[119] Y. Qian, J. Tang, Z. Yang, B. Huang, W. Wei, and K. M. Carley. A probabilistic framework for location inference from social media. *arXiv preprint arXiv:1702.07281*, 2017.

[120] L. Qin, J. X. Yu, and L. Chang. Diversifying top-k results. *Proceedings of VLDB Endowment*, 5(11):1124–1135, 2012.

[121] A. Rahimi, T. Cohn, and T. Baldwin. Twitter user geolocation using a unified text and network prediction model. *arXiv preprint arXiv:1506.08259*, 2015.

[122] A. Rahimi, T. Cohn, and T. Baldwin. A neural model for user geolocation and lexical dialectology. In *Proceedings of the 55th Annual Meeting of the ACL, ACL 2017, Volume 2*, pages 209–216, 2017.

[123] A. Rahimi, T. Cohn, and T. Baldwin. Semi-supervised user geolocation via graph convolutional networks. *arXiv preprint arXiv:1804.08049*, 2018.

[124] A. Rahimi, D. Vu, T. Cohn, and T. Baldwin. Exploiting text and network context for geolocation of social media users. *arXiv preprint arXiv:1506.04803*, 2015.

[125] V. Rakesh, C. K. Reddy, and D. Singh. Location-specific tweet detection and topic summarization in twitter. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 1441–1444. ACM, 2013.

[126] K. Ren, S. Zhang, and H. Lin. Where are you settling down: Geo-locating twitter users based on tweets and social networks. In *Asia Information Retrieval Symposium*, pages 150–161. Springer, 2012.

[127] D. Rout, K. Bontcheva, D. Preoiuc-Pietro, and T. Cohn. Where's wally?: a classification approach to geolocating users based on their social ties. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, pages 11–20. ACM, 2013.

[128] K. Ryoo and S. Moon. Inferring twitter user locations with 10 km accuracy. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 643–648. ACM, 2014.

[129] M. Sachan, D. Contractor, T. A. Faruquie, and L. V. Subramaniam. Using content and interactions for discovering communities in social networks. In *Proceedings of the 21st international conference on World Wide Web*, pages 331–340, 2012.

[130] A. Sadilek, H. Kautz, and J. P. Bigham. Finding your friends and following them to where you are. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 723–732. ACM, 2012.

[131] T. Sakaki, M. Okazaki, and Y. Matsuo. Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):919–931, 2013.

[132] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data mining and knowledge discovery*, 2:169–194, 1998.

[133] A. D. Sarma, H. Lee, H. Gonzalez, J. Madhavan, and A. Halevy. Consistent thinning of large geographical data for map visualization. *ACM Transactions on Database Systems (TODS)*, 38(4):22, 2013.

[134] S. Scellato, C. Mascolo, M. Musolesi, and V. Latora. Distance matters: Geo-social metrics for online social networks. In *WOSN*, 2010.

[135] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell. Nextplace: a spatio-temporal prediction framework for pervasive systems. In *International Conference on Pervasive Computing*, pages 152–169. Springer, 2011.

[136] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo. Socio-spatial properties of online location-based social networks. In *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[137] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1046–1054. ACM, 2011.

[138] C.-Y. Shen, D.-N. Yang, L.-H. Huang, W.-C. Lee, and M.-S. Chen. Socio-spatial group queries for impromptu activity planning. *IEEE TKDE*, 28(1):196–210, 2016.

[139] J. Shi, N. Mamoulis, D. Wu, and D. W. Cheung. Density-based place clustering in geo-social networks. In *Proceedings of ACM SIGMOD*, pages 99–110. ACM, 2014.

[140] C. Shim, W. Kim, W. Heo, S. Yi, and Y. D. Chung. Nearest close friend search in geo-social networks. *Information Sciences*, 423:235–256, 2018.

[141] R. W. Sinnott. Virtues of the haversine. *Sky Telesc.*, 68:159, 1984.

[142] A. Sohail, M. A. Cheema, and D. Taniar. Geo-social temporal top-k queries in location-based social networks. In *ADC*, pages 147–160. Springer, 2020.

[143] A. Sohail, G. Murtaza, and D. Taniar. Retrieving top-k famous places in location-based social networks. In *ADC*, pages 17–30. Springer, 2016.

[144] M. Sozio and A. Gionis. The community-search problem and how to plan a successful cocktail party. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 939–948, 2010.

[145] J. Su, K. Kamath, A. Sharma, J. Ugander, and S. Goel. An experimental study of structural diversity in social networks. *arXiv preprint arXiv:1909.03543*, 2019.

[146] Y. Takhteyev, A. Gruzd, and B. Wellman. Geography of twitter networks. *Social networks*, 34(1):73–81, 2012.

[147] H. Tian, M. Zhang, X. Luo, F. Liu, and Y. Qiao. Twitter user location inference based on representation learning and label propagation. In *Proceedings of The Web Conference 2020*, pages 2648–2654, 2020.

[148] A. Tigunova, J. Lee, and S. Nobari. Location prediction via social contents and behaviors: Location-aware behavioral lda. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*, pages 1131–1135. IEEE, 2015.

[149] M. R. Vieira, H. L. Razente, M. C. Barioni, M. Hadjieleftheriou, D. Srivastava, C. Traina, and V. J. Tsotras. On query result diversification. In *ICDE*, pages 1163–1174. IEEE, 2011.

[150] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1079–1088. ACM, 2010.

[151] F. Wang, G. Wang, and S. Y. Philip. Why checkins: Exploring user motivation on location based social networks. In *ICDMW*, pages 27–34. IEEE, 2014.

[152] J.-F. Wang, A. Stein, B.-B. Gao, and Y. Ge. A review of spatial sampling. *Spatial Statistics*, 2:1–14, 2012.

[153] K. Wang, X. Cao, X. Lin, W. Zhang, and L. Qin. Efficient computing of radius-bounded k-cores. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 233–244. IEEE, 2018.

[154] M. Wang, C. Wang, J. X. Yu, and J. Zhang. Community detection in social networks: an in-depth benchmarking study with a procedure-oriented framework. *Proceedings of the VLDB Endowment*, 8(10):998–1009, 2015.

[155] Z. Wang, C. Ye, and H. Zhou. Geolocation using gat with multiview learning. In *2020 IEEE International Conference on Smart Data Services (SMDS)*, pages 81–88. IEEE, 2020.

[156] D. Wu, Y. Li, B. Choi, and J. Xu. Social-aware top-k spatial keyword search. In *IEEE International Conference on Mobile Data Management Proceedings*. IEEE, 2014.

[157] S. Xu, R. Zhang, W. Cheng, and J. Xu. Mtlm: a multi-task learning model for travel time estimation. *GeoInformatica*, pages 1–17, 2020.

[158] W. Xu, C.-Y. Chow, and J.-D. Zhang. Calba: capacity-aware location-based advertising in temporary social networks. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 364–373. ACM, 2013.

[159] Y. Yamaguchi, T. Amagasa, and H. Kitagawa. Landmark-based user location inference in social media. In *Proceedings of the first ACM conference on Online social networks*, pages 223–234. ACM, 2013.

[160] Y. Yamaguchi, T. Amagasa, H. Kitagawa, and Y. Ikawa. Online user location inference exploiting spatiotemporal correlations in social streams. In *Proceedings of the 23rd ACM*

International Conference on Conference on Information and Knowledge Management, pages 1139–1148. ACM, 2014.

[161] D.-N. Yang, C.-Y. Shen, W.-C. Lee, and M.-S. Chen. On socio-spatial group query for location-based social networks. In *Proceedings of ACM SIGKDD*, pages 949–957. ACM, 2012.

[162] S. Yardi and D. Boyd. Tweeting from the town square: Measuring geographic local networks. In *Fourth international AAAI conference on weblogs and social media*, 2010.

[163] M. Ye, P. Yin, and W.-C. Lee. Location recommendation for location-based social networks. In *Proceedings of SIGSPATIAL GIS*, pages 458–461, 2010.

[164] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *ACM SIGIR*, pages 325–334, 2011.

[165] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen. Lcars: a location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 221–229. ACM, 2013.

[166] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Who, where, when and what: discover spatio-temporal topics for twitter users. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 605–613. ACM, 2013.

[167] F. Zhang, L. Yuan, Y. Zhang, L. Qin, X. Lin, and A. Zhou. Discovering strong communities with user engagement and tie strength. In *International Conference on Database Systems for Advanced Applications*, pages 425–441. Springer, 2018.

[168] F. Zhang, Y. Zhang, L. Qin, W. Zhang, and X. Lin. When engagement meets similarity: efficient (k, r)-core computation on social networks. *arXiv preprint arXiv:1611.03254*, 2016.

[169] F. Zhang, Y. Zhang, L. Qin, W. Zhang, and X. Lin. Efficiently reinforcing social networks over user engagement and tie strength. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 557–568. IEEE, 2018.

[170] C. Zheng, J.-Y. Jiang, Y. Zhou, S. D. Young, and W. Wang. Social media user geolocation via hybrid attention. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1641–1644, 2020.

[171] X. Zheng, J. Han, and A. Sun. A survey of location prediction on twitter. *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[172] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma. Recommending friends and locations based on individual location history. *ACM TWEB*, 5(1):1–44, 2011.

[173] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, and T. Zhao. Neural document summarization by jointly learning to score and select sentences. *arXiv preprint arXiv:1807.02305*, 2018.

[174] Q. Zhu, H. Hu, C. Xu, J. Xu, and W.-C. Lee. Geo-social group queries with minimum acquaintance constraints. *The VLDB Journal*, 26(5):709–727, 2017.

[175] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Citeseer, 2002.

[176] X. Zhu, J. Guo, X. Cheng, P. Du, and H.-W. Shen. A unified framework for recommending diverse and relevant queries. In *Proceedings of WWW*, pages 37–46, 2011.

[177] Y. Zhuang, S. Fong, M. Yuan, Y. Sung, K. Cho, and R. K. Wong. Location-based big data analytics for guessing the next foursquare check-ins. *The Journal of Supercomputing*, 73(7):3112–3127, 2017.

[178] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Proceedings of WWW*, pages 22–32. ACM, 2005.