Recommendation in Location-based Social Networks

by

Bo Hu

MSc., Renmin University of China, 2009

BSc., Beijing Language and Culture University, 2006

Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy

in the

School of Computing Science

Faculty of Applied Sciences

SIMON FRASER UNIVERSITY

Fall 2014

APPROVAL

Name: Bo Hu

Degree: Doctor of Philosophy

Title: Recommendation in Location-based Social Networks

Examining Committee: **Chair**: Dr. Ke Wang, Professor, Computing Science
Simon Fraser University

Dr. Martin Ester, Professor, Computing Science
Simon Fraser University
Senior Supervisor

Dr. Jian Pei, Professor, Computing Science
Simon Fraser University
Supervisor

Dr. Jiangchuan Liu, Professor, Computing Science
Simon Fraser University
Internal Examiner

Dr. Huan Liu, Professor, Computer Science and Engineering
Arizona State University
External Examiner

Date Approved: November 28, 2014

Abstract

Recommender systems have become popular tools to select relevant personalized information for users. With the rapid growth of mobile network users, the way users consume Web 2.0 is changing substantially. Mobile networks enable users to post personal status on online social media services from anywhere and at anytime. However, as the volume of user activities is growing rapidly, it is getting impossible that for users to read all posts or blogs to catch up with the trends. Similarly, it is hard for producers and manufactures to monitor consumers and figure out their tastes. These needs inspired the emergence of a new line of research, *recommendation in location-based social networks*, i.e., building recommender systems to discover and predict the behavior of users and their engagement with location-based social networks. Extracted users' interests and their spatio-temporal patterns clearly provide more detailed information for producers to make decisions to supply their consumers.

In this thesis, we address the problem of recommendation in location-based social networks and seek novel methods to improve limitations of existing techniques. We first propose a spatial topic model for top-k POI recommendation problem, and the proposed model discovers users' topic and geographical distributions from user check-ins with posts and location coordinates. Then we focus on mining spatio-temporal patterns of user check-ins and propose a spatio-temporal topic model to identify temporal activity patterns of different topics and POIs. In our next work, we argue that all existing social network-based POI recommendation models cannot capture the nature of location-based social network. Hence, we propose a social topic model to effectively exploit a location-based social network. Finally, we address the problem of determining the optimal location for a new store by considering it as a recommendation problem, i.e., recommending locations to a new store. Latent factor models are proposed and proved to perform better than existing state-of-the-art methods.

To my mother Qi Huang and my father Hua Hu,
for giving me a great starting point in life!
To my wife Jingbo Yu,
for adding positive to keep the training data balanced!

“I can’t change the direction of the wind,

but I can adjust my sails to always reach my destination!”

— Jimmy Dean

Acknowledgments

To my advisor Dr. Martin Ester, and to my committee as a whole, Dr. Jian Pei, Dr. Jiangchuan Liu, Dr. Ke Wang, and Dr. Huan Liu: thank you for your guidance.

To all my friends in the Department of Computing Science, Mohsen Jamali, Samaneh Moghaddam, Phuong Dao, Zhihui Guo, Chao Han, Peng Wang, Judy Yeh, Zhensong Qian, Yao Wu, Hao Wang, Xin Wang, Beidou Wang etc.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Millions of people now use social networking websites to enjoy online interaction with friends and meeting new people. Social networking sites, such as Twitter[1], Facebook[2], and LinkedIn[3] etc, are attracting an increasing number of users, many of whom have integrated these sites into their daily practices.

Thanks to the widespread adoption of various smart mobile devices, people can easily post their routine status from anywhere at anytime. Consequently, we can see the unprecedented access to the news, events, and activities, with tons of user-generated data in highly dynamics. Since users tend to have personalized results but are not willing to spend a lot of time to specify their personal information needs, it becomes necessary to have tools to select relevant part of information automatically. Recommender systems have emerged to bridge the gaps between users and social media providing recommendations on all kinds of products, e.g., movies, books, and music etc.

Since the early papers [55, 23] published in mid-1990s on collaborative filtering, recommender systems is increasingly attracting the attention of academic and industry researchers developing new approaches over the last decades [1].

On the one hand, [12, 59, 50, 43, 22] are the research representatives published in early 2000s at different research areas, e.g., information retrieval, web mining, and human factor analysis. Recommender systems grows and becomes an independent research area since the ACM Recommender System conference (RecSys) is founded in 2007. More and more

[1]http://www.twitter.com

[2]http://www.facebook.com

[3]http://www.linkedin.com

researchers from areas of data mining and machine learning join the field inventing new systems and techniques, and publishing experimental results in the broad area of recommender systems.

On the other hand, some real life recommender systems are really successful. For example, one of the most important recommender systems is product recommendation on Amazon [43] or similar online retailers, where recommendation engines return users with some suggestions of products that they might like to purchase. Moreover, Netflix[4] offered a prize of one million dollars for the algorithm that could beat its own recommender system, [39] won the prize by improving 10% recommendation accuracy (measured in terms of the root mean square error) in 2009 after over three years of competition. Following this track, many works [9, 58, 38, 39, 40] have been widely investigated in recent years.

In general, there are two main types of entities in the recommender system: *users* and *items*, where items could be products, movies, or news articles etc. Each user performs actions on a set of items, and an action indicates that the user purchases a product, rates a movie, or reads a news article. Given a user, a recommender system learns the *user preference* from the action history on items of the given user, and identifies and recommends the relevant information (items) for that given user. The formal definition of recommender system on Wikipedia[5] is as follows:

Definition 1 ***Recommender System***. *A recommender system is a particular form of information filtering, that exploits past behaviors and user similarities to generate a list of information items that is personally tailored to a user's preferences.*

1.1 Demands for Recommendation

The Web has become the primary source of information, and search engines are the primary tools that people use to find information. Different from the keyword search on the Web, recommender systems applies personalization techniques for users in finding and selecting products, services, or information. Although the efficiency of recommendation can be worse than keyword search, it enables online web services to provide personalized and accurate recommendations for users. As a result, the recommendation enhances the

[4] http://www.netflix.com

[5] http://en.wikipedia.org/wiki/Recommender_system

satisfactory of user experiences and increases the cohesiveness between users and web services. We note that there are non-personalized recommendations that are useful in certain applications, but they are not normally addressed in the area of recommender systems.

Ricci et al. [56] present some facts that prove the interest in recommender systems has grown in recent years which is shown as follows:

- Recommender systems play an important role in the following highly rated websites as Amazon, YouTube, Netflix, Yahoo, Tripadvisor, Last.fm, and IMDb. Many social media companies are developing and deploying recommender systems as part of their services.

- Related conferences and workshops are booming in recent years, such as, ACM Recommender System conference (RecSys), ACM SIGIR Special Interest Group on Information Retrieval (SIGIR), User Modeling, Adaptation and Personalization (UMAP), and ACM's Special Interest Group on Management Of Data (SIGMOD).

In particular, Posse et al. [54] also report some interesting results of a large-scale recommender system on LinkedIn as follows:

- 80% of a LinkedIn homepage are powered by recommendations.

- 50% of total job applications and job views by LinkedIn users are a direct result of recommendations.

- More than 50% of social connections are from recommendations.

The era of "big data" is coming, and 3Vs (Volume, Variety and Velocity) are three promising properties or dimensions of big data. Volume refers to the amount of data, variety refers to the number of types of data, and velocity refers to the speed of data processing. 3Vs of the big data demand powerful recommender systems provide personalized or customed services for users overcoming the information overload.

In this thesis, we address the problem of recommendation in location-based social networks and seek novel methods to improve limitations of existing techniques. The rest of this thesis is organized as follows. Firstly, we survey the related work of recommendation in location-based social networks in Chapter 2, which are categorized into context-free and context-aware recommendations, and we discuss some existing state-of-the-art works in location-based recommendation.

In Chapter 3, we introduced a spatial topic method for the POI recommendation problem. The proposed model extracts users' topic distributions by mining a set of topics from user check-ins with posts, and models coordinates of checked in POIs using a two dimensional Gaussian distribution with a set of regions. In addition, each user has a probability distribution over all the regions. Experiment results on two real life data sets from Twitter and Yelp[6] demonstrate the accuracy of the proposed model.

In Chapter 4, to exploit time information associated with user check-ins, we introduce a spatio-temporal topic model to learn a set of spatio-temporal topics from the user check-in data. The proposed model jointly identifies user and temporal topics. Again, the experimental evaluation on three real life data sets from Twitter, Gowalla, and Brightkite demonstrate the substantial improvement of recommendation quality in the proposed model.

Social influence and selection together lead to similar behavior among friends. In Chapter 5, we introduce a social topic model which takes advantage of a social network for POI recommendation, but it captures the nature of location-based social network. On two real life data sets from Foursquare[7] and Yelp, experiments demonstrate that the proposed model consistently improves the performance significantly for POI recommendation compared to existing state-of-the-art social network-based recommendation algorithms for all users, all POIs, cold start users, and cold start POIs.

In Chapter 6, another interesting recommendation problem in location-based social networks, i.e., determining the optimal location for a new store, has been studied. To the best of our knowledge, we are the first to formulate this problem as a recommendation problem, i.e., recommending locations to a new store of the given store chain.

Finally, we conclude this thesis and present some directions for future work in Chapter 7.

[6] http://www.yelp.com

[7] http://www.foursquare.com

Chapter 2

Related Work

In this chapter we review the existing related works on recommender systems or recommendation, and recommendation algorithms in the location-based social networks in the literature. Before discussing our categorization of recommendation, we present current formulation schemas proposed in the literature.

The early works [55, 23, 7, 1] survey recommender systems, and divide this area into three categories according to recommendation methodologies: content-based, i.e., systems that recommend an item to a user based upon a description of the item and a profile of the user, collaborative filtering, i.e., systems that recommend an item for a particular user based on the items previously rated by other "similar" users, and hybrid systems, i.e., systems that combine the content-based and collaborative filtering techniques.

Two recent books [56, 34] also define the same classes of recommendations, and they further describe a new direction of recommender systems: context-aware recommendation, i.e., the recommender system provides recommendations considering contextual information such as time and location etc. For example, using the temporal context, a web content recommender system normally recommends the news over the world for readers on weekdays, and recommends movie posters and shopping discount news on weekends. Moreover, some music recommender systems can even detect the listeners' mood when providing music recommendations that songs from different genres apply for different moods.

Here, we borrowed the definition of "context" from Oxford dictionary[1] as follows:

[1] http://www.oxforddictionaries.com

Definition 2 (Context) *The circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood.*

Inspired by the above literature summary and the definition of "context", we group major recommendation approaches according to the data resources by the following two categories:

- Context-free recommendation
- Context-aware recommendation

The main difference between the above two categories is the input data whether contains contextual information. It should be noted that early and traditional recommendation problems such as rating prediction are usually context-free, while recent recommendation problems are often context-aware because of the booming social networking websites, which provide us a huge volume of useful contextual information for improving the recommendation accuracy. For example, given the target user's list of followers on Twitter, a content recommender system can capture the user's interests or preferences without mining the user's action history.

In the following sections, for both categories we will define the research problem and review the existing state-of-the-art works.

2.1 Context-free Recommendation

As we mentioned, many traditional recommendation approaches are designed for context-free recommendations, in which there are only two entities: *users* and *items*. Formally, we assume that there are a set of users $U = \{u_1, u_2, ..., u_{|U|}\}$ and a set of items $I = \{i_1, i_2, ..., i_{|I|}\}$, where $|U|$ and $|I|$ represent the number of users and items, respectively. Each user performs actions on a set of items. The actions performed by users on items are given in a matrix $A = [a_{u,i}]_{|U| \times |I|}$, where $a_{u,i}$ denotes the action of user u on item i. We should note that normally more than 90% entries in the matrix A are missing or unknown. For the remaining known entries, $a_{u,i}$ can be any real number, and convey various meanings according to varying types of recommendations. For example, in movie recommender systems, $a_{u,i}$ can be integers normally in the range of $[1, 5]$ to represent user ratings on items.

Furthermore, $a_{u,i}$ can be binary numbers to represent whether users browse or purchase the item in product recommendations.

We define the problem used in the area of context-free recommendation. Rating prediction is by far the most common recommendation task:

Problem 1 (Rating prediction) *Given an incomplete action matrix $A = [a_{u,i}]_{|U|\times|I|}$, for a user $u \in U$, the recommender system learns his/her preferences, and predicts the unknown rating $a_{u,i}$ of item i.*

Recommendations are evaluated by the quality of their predictions with respect to the predictions of ground truth. Typically, we randomly select a percentage (e.g., 70%, 80%, and 90%) of observed data for each user as the training data, and the remaining (e.g., 30%, 20%, and 10%) as the testing data. We train the model using the training data, and predict the actions in the testing data.

As we discussed, rating prediction is a traditional recommendation task studied in many works [9, 58, 38, 39, 40]. Two common metrics for computing the error of predictions are Root Mean Squared Error (RMSE) [9, 58] in Equation 2.1 and Mean Absolute Error (MAE) [59, 22] in Equation 2.2:

$$RMSE(A_{test}) = \sqrt{\frac{\sum_{u,i|A_{test}}(a_{u,i} - \hat{a_{u,i}})^2}{|A_{test}|}} \tag{2.1}$$

$$MAE(A_{test}) = \frac{\sum_{u,i|A_{test}}|a_{u,i} - \hat{a_{u,i}}|}{|A_{test}|} \tag{2.2}$$

where A_{test} is the test data containing a set of user-item pairs, $|A_{test}|$ represents the number of user-item pairs, and $a_{u,i}$ and $\hat{a_{u,i}}$ represent the actual rating and predicted rating of item i by user u receptively. Note that the smaller value of RMSE or MAE indicates a better recommendation.

Some recent works [67, 18, 41] address the problem of top-k item recommendation, which is more natural than rating prediction in real recommendation applications because users really need recommended items instead of predicted ratings of items. We formally define top-k item recommendation as follows:

Problem 2 (Top-k Item Recommendation) *Given a user $u \in U$ and a set of items I, the recommender system returns a top-k ranked list of unseen items in which the given user u will be interested by using the incomplete action matrix $A = [a_{u,i}]_{|U|\times|I|}$.*

[67, 18, 41] have proposed models implementing learning to rank for top-k item recommendation. We should note that the results of rating prediction of a user can be served for top-k item recommendation, that is predicted top-k rated items are returned for the given user.

For top-k item recommendation, the measurement of the RMSE or MAE is replaced by a ranking function as the metric function. NDCG@k is by far the most commonly used metrics to measure the performance of recommender systems [22, 67, 18] for top-k item recommendation.

Normalized Discounted Cumulative Gain@k (NDCG@k): The NDCG@k for a user is computed by comparing the predicted ranked list of items and the ground truth of ranked list of items.

To compute DCG@k of the recommended list of items for a user u, we use the following equation:

$$DCG@k(u) = \sum_{n=1}^{k} \frac{2^{rel(l_n)} - 1}{\log_2(n+1)} \tag{2.3}$$

where l_n represents the recommended item index that is ranked at n^{th} position, and $rel(l_n)$ represents the relevance of the item. We use its relative position in the ground truth ranked list as used in [22]:

$$rel(l_n) = \frac{|L| - rank(l_n) + 1}{|L|} \tag{2.4}$$

Note that the relevance value is 1 when the item is ranked first and decreases to 0 when the ranking goes down. The DCG@k is normalized by the iDCG@k (ideal DCG@k) as follows:

$$NDCG@k(u) = \frac{DCG@k(u)}{iDCG@k(u)} \tag{2.5}$$

where the iDCG@k is the DCG@k score of the ranked list of ground truth.

According to the recommendation methodologies, we group context-free recommendation approaches into content-based and collaborative filtering-based categories. We discuss them in the following subsections.

2.1.1 Content-based Recommendation

Content-based recommender systems try to recommend items similar to those items a given user has liked in the past. In particular, a content-based approach analyzes a set of description of items rated by users in the past, summarizes the commonalities among these items, and builds a model on profiling interests of the given user based on features [56]. For example, in movie recommendations, the predicted rating $\hat{a_{u,i}}$ of movie i for user u is estimated by the ratings $a_{u,j}$ for user u of items $j \in I$ that are similar to item i. The commonalities can be actors, directors, or genres etc [1].

A book chapter [52] describes an algorithm usually implemented in content-based recommender systems, which contains two steps: item representation and user profiling. In item representation, each item is described by a set of features from either structured databases or unstructured data. An extreme case of unstructured data is format-free texts in the news recommendation. After removing stop words and stemming from raw texts, each word is represented by a feature, and each news article (item) is represented by a vector of words (features). The Term Frequency-Inverse Document Frequency (TF-IDF) weight is used to represent the feature value. The next step of content-based recommendation is to learn a given user's preferences from the user's action history. The user action history data is served as training data for different machine learning approaches, such as, decision tree, nearest neighor, and naive bayes algorithms etc, that create user profiling models presented in [1, 52]. Creating a model of the user's preference from the user history can be simplified as a classification problem. The training data divided into binary categories: "items that the user likes" and "items that the user dislikes".

Although different models can be built for content-based recommendations by learning users' interests, no content-based recommender systems can make accurate recommendations without sufficient item descriptions for distinguishing items that the user likes from items that the user dislikes [52]. In such cases, collaborative filtering-based recommendation approaches are proposed for exploiting only the user-item matrix. Besides, content-based recommendation methods can be used as a pre-processing tool to filter some items, which speeds up the followed up recommendation process.

2.1.2 Collaborative Filtering-based Recommendation

Collaborative filtering-based recommendation approaches try to predict the rating of items for a user based on the known ratings of the given user and other users. Unlike content-based recommendation methods, item descriptions are not needed in collaborative filtering-based recommenders. A large range of works [12, 59, 9, 58, 38, 39, 40] has proposed Collaborative Filtering (CF) methods in recommender systems for the rating prediction problem. According to [12, 1], there are memory-based, e.g., K Nearest Neighbor (KNN) [59] and model-based, e.g., Matrix Factorization (MF) [58] methods. A CF method [39] won the Netflix Prize competition, considering the temporal dynamics in a matrix factorization model.

Memory-based approaches [12, 59, 9] make rating predictions based on previously rated items by the entire users. The predicted rating $\hat{a_{u,i}}$ of item i for user u is usually computed as a weighted average of ratings of a small number of similar users for the same item i in Equation 2.6.

$$\hat{a_{u,i}} = \frac{\sum_v sim_{u,v} a_{v,i}}{\sum_v sim_{u,v}} \tag{2.6}$$

where user v are normally from top-k users that are most similar to the given user u.

Now the key is that how to compute the similarity $sim_{u,v}$ between user u and user v. Cosine similarity and correlation similarity are two representative approaches used in the literature [12].

Two users u and v are represented by two vectors $\overrightarrow{u}$ and $\overrightarrow{v}$ in $|I|$ dimensional item space. The similarity between them is measured by computing the cosine of the angle between these two vectors $\overrightarrow{u}$ and $\overrightarrow{v}$ as shown in Equation 2.8.

$$sim_{u,v} = cos(\overrightarrow{u}, \overrightarrow{v}) = \frac{\overrightarrow{u} \cdot \overrightarrow{v}}{||\overrightarrow{u}||_2 ||\overrightarrow{v}||_2} \tag{2.7}$$

where "$\cdot$" denotes dot product of two vectors and "$|| \, ||_2$" denotes the L_2 norm of a vector.

Correlation similarity between two users are computed by the Pearson correlation.

$$sim_{u,v} = corr(u, v) = \frac{\sum_{i \in I_{u,v}} (a_{u,i} - \overline{a}_u)(a_{v,i} - \overline{a}_v)}{\sqrt{\sum_{i \in I_{u,v}} (a_{u,i} - \overline{a}_u)^2} \sqrt{\sum_{i \in I_{u,v}} (a_{v,i} - \overline{a}_v)^2}} \tag{2.8}$$

where $\overline{a}_u$ represents the average rating of user u and $I_{u,v}$ represents the set of items rated by both user u and v.

The memory-based approaches are easy to implement but slow to make the recommendation since they need to explore the whole ratings.

Model-based recommendation approaches [58, 38, 39, 40] learn a model with parameters from the user-item matrix A, and make the rating prediction using learned parameters. In contrast with memory-based approaches, model-based approaches have a time consuming learning phase. But the advantage of model-based approaches is that the entire action history data can be discarded after learning the model, and ratings can be predicted using only the parameters of the model.

Based on the assumption that ratings in the user-item matrix can be inferred from a model with a smaller number of parameters, Latent Factor (LF) models [40] have been proposed to find user and item preferences describing the latent factors. A LF model can model the interactions between different types of entities, such as "user-item" in recommendation problems, to discover their latent factors and relationships. LF assumes that each user and each item share a $|K|$ dimensional latent factor vector in [38] as follows:

- Each user $u \in U$ is mapped into a latent factor vector $\boldsymbol{\phi}_u$ from a $|K|$-D real number space $\mathbb{R}^{|K|}$.
- Each item $i \in I$ is mapped into a latent factor vector $\boldsymbol{\phi}_i$ from a $|K|$-D real number space $\mathbb{R}^{|K|}$.
- Each user $u \in U$ is mapped into a latent bias factor b_u from a 1-D real number space $\mathbb{R}^1$.
- Each item $i \in I$ is mapped into a latent bias factor b_i from a 1-D real number space $\mathbb{R}^1$.

In order to learn the latent factors of users and items, [38] assumes that the prediction is done by taking an inner product of the user and item latent factors taking bias factors of users and items into consideration:

$$\hat{a}_{u,i} = b_u + b_i + \boldsymbol{\phi}_u^T \cdot \boldsymbol{\phi}_i \tag{2.9}$$

Conventional Singular Value Decomposition (SVD) methods can solve the factorization problem for complete matrices, but it is undefined for incomplete matrices of rating prediction. Hence Matrix Factorization (MF) is proposed as one of the most common techniques for recommender systems to simply ignore the missing ratings.

The parameter learning process is to minimize the objective function of known ratings defined as follows:

$$\mathcal{O} = \sum_{u,i|A_{train}} \mathcal{L}(a_{u,i} - \hat{a}_{u,i}) + \sum_{u,i|A_{train}} \mathcal{R}(b_u, b_i, \boldsymbol{\phi}_u, \boldsymbol{\phi}_i) \tag{2.10}$$

where $u, i|A_{train}$ represents the known user-item pairs in the training data set, and the $\mathcal{L}(.)$ represents the loss function and normally the least square loss function is used. The $\mathcal{R}(.)$ function represents the regularization function of the bias and latent factors, and normally the L_2 norm is used for each single parameter. Stochastic gradient descent and alternating least squares are two common algorithms for learning the parameters in MF models.

Another line of model-based approaches are originated from topic models for the top-k item recommendation problem, such as Probabilistic Latent Semantic Analysis (PLSA) [24] and Latent Dirichlet Allocation (LDA) [11] etc. Topic modeling is a classic problem and has a long history in text mining. Topic models tackle on modeling and extracting topics from documents. The assumption is that there are a set of topics in the documents, and each document has a probability distribution over all the topics and each topic has a probability distribution over all the words from a fixed set of vocabulary.

Topic models such as PLSA and LDA can be applied for the rating prediction problem if "documents" are replaced by "users" and "words" are replaced by "items". Latent factors in the Matrix Factorization (MF) approaches are equivalent to "topics" in topic models. The user-item matrix can be computed by the user and item latent factors through MF techniques, and these user and item factors are equivalent to users' topic distributions and topics' item distributions, respectively.

Let us take LDA as an example. Figure 2.1 shows the graphical model of LDA, where θ_u denotes the topic distribution of user u and ϕ_z denotes the word distribution of topic z. Items $\mathbf{i}$ are modeled as observed random variables, shown as shaded circles, while the latent random variables of topics $\mathbf{z}$ are shown as unshaded circles. LDA is a generative model. To generate an item i by a user u, LDA samples a topic z from a multinomial distribution θ_u. Given the sampled topic z, LDA samples an item i from a multinomial distribution ϕ_z. Both θs and ϕs have Dirichlet priors, which are known as hyperparameters and omitted in the following figure.

To learn the parameters of the LDA model, the marginal log-likelihood $p(\mathbf{i}|\mathbf{u}, \Theta, \Phi)$ of the observed random variables $\mathbf{i}$ needs to be maximized. The marginalization is performed with respect to the latent random variables $\mathbf{z}$, and it is hard to be maximized directly. Therefore, we apply the MCEM (Monte Carlo Expectation Maximization) algorithm to maximize the complete data likelihood. $p(\mathbf{z}, \mathbf{i}|\mathbf{u}, \Theta, \Phi)$ in Equation 2.11.

$$p(\mathbf{z}, \mathbf{i}|\mathbf{u}, \Theta, \Phi) = p(\mathbf{z}|\mathbf{u}, \Theta) \times p(\mathbf{i}|\mathbf{z}, \Phi) = \prod_{u=1}^{|U|} \prod_{d=1}^{|D_u|} \theta_{u,z_{u,d}} \times \prod_{u=1}^{|U|} \prod_{d=1}^{|D_u|} \phi_{u,z_{u,d}} \qquad (2.11)$$

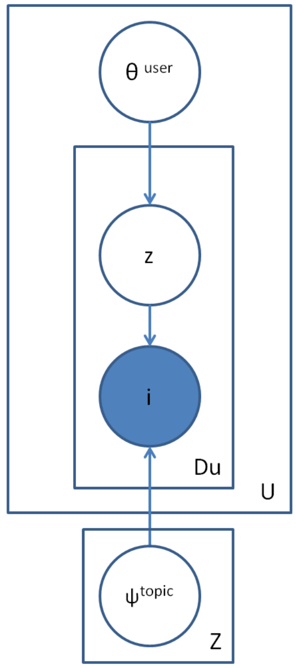


Figure 2.1: The graphical model of LDA

Given the learned parameters Θ and Φ, we use Equation 2.12 to predict a rating $\hat{a}_{u,i}$ of item i for user u.

$$\hat{a}_{u,i} = p(i|u, \Theta, \Phi) \propto \sum_{z}^{Z} p(i, z|u, \Theta, \Phi) = \sum_{z}^{Z} p(z|u, \theta) \times p(i|z, \phi) \tag{2.12}$$

The predicted top-k rated items are recommended for users.

2.2 Context-aware Recommendation

In this section, we summarize the existing works in the area of context-aware recommendations. The majority of existing recommendation approaches focus on recommending most relevant items to users without considering any contextual information. However, in many applications, like recommending a restaurant, it may not be sufficient to consider only users and items. It is important to include contextual information, such as location, time, and social connections, into the recommendation process. Many companies started incorporating some contextual information into their recommendation engines. For example, when recommending a news article for the user, the LinkedIn's news recommender system takes into the consideration the colors of photos or pictures (the context) that the given user liked.

In this section, we discuss two areas of context-aware recommendation: social-based recommendation, i.e., recommending items using social networks, and location-based recommendation, i.e., recommending Point-of-Interest (POI) using spatio-temporal (coordinates of POIs and time) contexts. We would like to demonstrate that depending on the application and the availability of the data, certain contextual information can be helpful for better recommendations. We also believe that there are many other context-aware recommendation problems.

2.2.1 Social-based Recommendation

With the rapidly growing of social networking websites such as Facebook and LinkedIn on the WWW, it is hard to ignore the power of social networks to help existing recommender systems for better recommendations. The most common problem in social-based recommendation is rating prediction. Formally, along with the user-item action matrix, there are social networks available. The problem is to predict unknown ratings of users for items. A social network is usually represented as a directed/undirected graph, where nodes denote users and edges denote social relationships. For example, in Epinions users provide ratings on products, and users establish a trust network, and Flixster serves as a platform for users to rating movies and has a social network from Facebook.

Some representative works [49, 47, 32, 33, 48, 68, 62] in social-based recommendation utilize the social network to help the recommender system make the decision. Intuitively, friends in the social network tend to have similar rating patterns than non-friends. Social recommendation methods have been proved successfully on rating prediction, especially on "cold start" users.

In memory-based social recommendation approaches [49, 32], for a user they propose to find top-k similar users as the same in the memory-based recommendation approaches of context-free recommendation introduced in the previous sections. They also propose to find top-k similar friends of the given user. Additionally, [32] proposes to explore not only the 1st degree friends but also the friends of nth degree, and the importance of each friend in the social network is weighted by the random walk probability starting from the given user, i.e., the importance of each friend is penalized by the length between the friend and the given user. The predicted unknown rating score of a user on an item is a weighted average of the ratings of top-k similar users and top-k similar friends.

In model-based social recommendation approaches [47, 33, 48, 68, 62], they usually propose to model user-item action matrix and user-user relationships.

Ma et al. [47, 48] introduce the notion of *social recommendation*, and assume that *friends in the social network have similar interests.* To model the social relationships into recommendations, they proposed a social regularization terms to constrain the matrix factorization framework. Specifically, they extend the objective function in Equation 2.10 by adding several regularization terms as follows:

$$\mathcal{O} = \sum_{u,i|A_{train}} \mathcal{L}(a_{u,i} - \hat{a}_{u,i}) + \sum_{u,i|A_{train}} \mathcal{R}(b_u, b_i, \boldsymbol{\phi}_u, \boldsymbol{\phi}_i) + \sum_{u,v|E} ||\boldsymbol{\phi}_u - \boldsymbol{\phi}_v||_2 \tag{2.13}$$

where the last term is the individual-based social term that is designed to minimize the difference between the latent factor for a user and his/her friends.

Based on [47], Jamali et al. [33] further consider the trust propagation in social networks. The latent factor of a user is regularized by their direct friends at each iteration in the learning process.

Yang et al. [68] present a friendship-interest propagation (FIP) model that integrates the learning for interest targeting and friendship prediction into one process by defining a coupled model to encode both interest and friendship information. Specifically, in the training process, FIP models the interaction between the friendship network and the interest network by introducing a shared latent factor.

Furthermore, Shen et al. [62] argue that the social network have its unique properties, which tells that the social friendship has heterogeneity and diversity properties. Based on this further understanding, they applied the stochastic blockmodel to handle/express those two properties. This work provides a deeper understanding of the social network influences.

Experimental results show that all the above social recommendation approaches [49, 47, 32, 33, 48, 68, 62] consistently outperform the existing approaches of context-free recommendation, and especially the improvement is noticeably substantial for “cold start” users.

2.2.2 Location-based Recommendation

Recently, Location-based recommendation has attracted a lot of attention. Some methods [71, 76, 75, 72, 70, 8, 20, 13, 42, 45, 44, 73] have been proposed to consider the contexts such as coordinates of POI, time, and texts in recommendation approaches. Since location-based recommendation is the main research problem addressed in this thesis, we first define the problem of location-based recommendation in detail, and then extensively review current methods and approaches.

As we discussed, in recent years recommender systems have gained in popularity and have become hot topics attracting the attention of academic and industrial researchers. Meanwhile, with the rapid growth of mobile network users, the way users consume Web 2.0 is changing substantially. Mobile networks enable users to post on social media services (e.g., Twitter) from anywhere and at anytime. This new phenomenon led to the emergence of a new line of research, *recommendation in Location-based Social Networks (LBSNs)*, i.e., building recommender systems to not only mine and discover the behavior of users but also to take into account the rich social media aspects, such as, textual, social, and spatio-temporal, of their engagement with location-based social networks.

In the classic recommendation framework there is a user-item matrix and each element in the matrix represents the user's rating of that item. To put it in the context of location-based recommendation, Point-of-Interest (POI) can be viewed as "items", and "user-item ratings" can represent the frequency with which a user has visited the corresponding POIs.

Besides, location-based social networks provide rich social media containing contextual information with user actions, such as reviews. A large number of users generate all kinds of content in location-based social networks. These online activities (also known as "check-ins") of users can be typically represented as follows: a user appears at a certain POI (with a pair of latitude and longitude coordinates) and leaves a post at a certain time. More precisely, a check-in has the following attributes: text, author, POI, coordinates, and time. An example of publicly available check-ins from Twitter is as follows:

- Close to the equator, perfect soil and high elevation make @CafeDAltamira produce the perfect cup of Coffee!! # Honduras # Coffee — @XXX — 37.38 -121.90 — "2012-04-23 12:04:09" — San Jose — CA — United States — 0befbacea94beb06.

This tweet states that a Twitter user from San Jose, California compliments the coffee at a coffee shop, where "37.38, -121.90" are the latitude and longitude coordinates, and "0befbacea94beb06" is the unique label of the coffee shop.

In particular, POI recommender systems can recommend a set of POIs that users may be interested in, based on the history of user check-ins. Similar to the problem of top-k item recommendation in Problem 2, we formally define top-k POI recommendation as follows:

Problem 3 (Top-k POI Recommendation) *Given a user $u \in U$ and a set of POIs I, the POI recommender system returns a top-k ranked list of unvisited POIs where the given user u will be interested checking in by using the user-POI matrix and one or multiple contexts: 1) coordinates (a pair of latitude and longitude) of POIs, 2) a social network among users, 3) user-generated texts.*

As we mentioned, user-generated texts can be tweets posted at the POIs on Twitter, tags of venues on Foursquare, or reviews of businesses on Yelp. Here, "POIs", "venues", and "businesses" are synonyms, and "tweets", "tags", and "reviews" are equivalent to "user-generated texts".

Recommendations are evaluated by the accuracy of their predictions compared to the ground truth. NDCG@k, Precision@k, Recall@k, and perplexity are common metrics to measure the performance of models in top-k POI recommendation. NDCG@k is defined in the previous chapter.

Precision@k: Precision@k is another metric to measure the performance of recommender systems [22]. The top-k precision for a test item is one when the ground truth item in the top-k recommendations, and zero otherwise. The precision@k is the average top-k precision over all test items, i.e.,

$$Precision@k(D_{test}) = \frac{\sum_{d=1}^{|D_{test}|} \mathcal{I}(i_d, \hat{i_{d1}}, \hat{i_{d2}}, ..., \hat{i_{dk}})}{|D_{test}|} \tag{2.14}$$

where $\mathcal{I}(i_d, \hat{i_{d1}}, \hat{i_{d2}}, ..., \hat{i_{dk}})$ is an identity function.

Recall@k is another commonly used metric to measure the performance of top-k recommendations [38, 18].

In order to compute the recall@k, we first compute hit@k. For each check-in of user u on POI i in the testing data set, hit@k is computed as follows:

1. The model randomly selects 1000 additional POIs that user u has not visited. Note that the hypothesis is that all the 1000 randomly selected POIs that user u is not interested in.

2. The model predicts the probability of user u checking in at actual POI i and at the additional 1000 POIs. The estimated probability $p(i_d, w_d|u, \Theta)$ of observed POIs given user u and learned parameters is computed by Equation 5.13.

3. The model ranks the 1001 POIs by their predicted probabilities.

4. The model returns top-k POIs by picking the k top ranked POIs. Let p denote the rank of the actual test POI i. We have a hit and $hit@k = 1$ if $p <= k$, otherwise we have a miss and $hit@k = 0$. Note that the best result is that the actual test POI i is ranked higher than all other additional 1000 POIs (i.e., p = 1).

For each user u, the recall@k is computed by averaging over all user u's test check-ins:

$$Recall@k(u) = \frac{\#hits}{|D_{test}|} \tag{2.15}$$

where D_{test} represents the test data set of user u, and $|D_{test}|$ is the number of test check-ins.

The recall@k of the whole test data set is computed by averaging over all users:

$$Recall@k(U) = \frac{\sum_{u \in U} Recall@k(u)}{|U|} \tag{2.16}$$

Finally, perplexity is introduced to evaluate the methods. We estimate the likelihood of the test data set given the trained models. Perplexity is the standard for measuring how well a probabilistic model fits the data [10], and is monotonically decreasing in the likelihood of the test data set, so that a lower perplexity indicates better performance of the model. We compute the perplexity as follows:

$$Perplexity(D_{test}) = \exp\Big\{ - \frac{\sum_{d=1}^{|D_{test}|} \log p(i_d|u, \Theta)}{|D_{test}|} \Big\} \tag{2.17}$$

where D_{test} represents the test data set, and $|D_{test}|$ is the number of documents in the test data set. We compute the estimated likelihood $p(i_d|u, \Theta)$ of observed POIs given learned parameters of the model.

The problem of POI recommendation can be simplified as a context-free item recommendation problem, so that traditional recommendation approaches discussed in previous sections, such as, collaborative filtering [58, 38, 40], can be applied, as well as the topic models [24, 11] if "words" are replaced by "POIs". For example, the user-POI matrix can be computed by the user and POI latent factors through Matrix Factorization (MF) techniques, and these user and POI factors are equivalent to users' topic distributions and topics' POI distributions, respectively.

Although all of the traditional recommendation methods can be applied for POI recommendation, none of them considers the contextual information in location-based social networks. Recent works [71, 76, 75, 72, 70, 8, 20, 13, 42, 45, 44, 73] propose models for POI recommendation considering one or multiple contexts. According to different input contexts used in their methods, we group them into the following three categories and summarize them in Table 2.1 in a chronological order of their publication dates.

- Social network-based approaches
- Coordinates-based approaches
- Text-based approaches

Table 2.1: State-of-the-art Works for Top-k POI Recommendation. √ represents that the work of the corresponding row belongs to the category of the corresponding column.

Literature	Social Network-based	Coordinates-based	Text-based
Ye et al. 2010 [71]	√		
Zheng et al. 2010 [76]			√
Zheng et al. 2010 [75]			√
Ye et al. 2011 [72]		√	
Bao et al. 2012 [8]			√
Gao et al. 2012 [20]	√		
Cheng et al. 2012 [13]		√	
Kurashima et al. 2013 [42]		√	
Liu et al. 2013 [45]			√
Liu et al. 2013 [44]		√	
Yin et al. 2013 [73]			√

In the following sections, we will discuss each of these categories, their strengths and weaknesses, and the state-of-the-art methods.

2.3 Social Network-based Approaches

Most of the early works in POI recommendation are social network-based approaches [71, 20]. In the problem 3 of POI recommendation, given a user-POI matrix, coordinates of POIs, and a social network among users, they build a POI recommender system that recommends POIs for users. A social network is represented as an undirected graph $G =$

(U, E), where U denotes the set of users, and an undirected edge $(u, v) \in E, u \in U, v \in U$ from user u to user v represents the fact that u and v are friends.

In general, given a target user, these approaches [71, 20] search his/her friends in the social network, and recommend POIs visited by his/her friends for the given user. Their theoretical foundation is based on the prominent phenomenons of homophily and social influence in social networks, i.e., the homophily phenomenon suggests that similar users are more likely to connect to each other, and the social influence phenomenon indicates that friends tend to influence each other's preferences and actions. Basically, the latent factors of users and POIs are learned from the user-POI matrix using collaborative filtering approaches, and the users' latent factors are influenced by their friends. Moreover, based on coordinates of POIs, they compute the social influence weight between two friends by the distance of their historical POIs, and short distances indicate large social network weights.

Experimental results in [71, 20] show that social network-based approaches outperform those approaches without social networks, but the top-k POI recommendation accuracy improvement is not as substantial ($\sim$5%) as in the top-k movie recommendation ($\sim$30%) in [69]. We argue the reason is that check-in actions require users' physical commitment to POIs, which are more serious than actions of rating a movie online. In this case, the co-occurrences of friends at POIs are less than the movies that friends co-like. How to accurately profile users for POI recommendation becomes the major challenge for social network-based approaches.

2.3.1 Ye's Model (2010)

Ye et al. [71] propose a Friend-based Collaborative Filtering (FCF) method in 2010, which is the first work to tackle the problem of POI recommendation. This method is originated from a memory-based collaborative filtering approach. For a specific user, the traditional collaborative filtering approach computes the similarity between the given user and all other users using the cosine similarity in Equation 2.8, and chooses top-k most similar users as k nearest neighbors. The score of a POI for the given user is calculated as the weighted average score of the chosen top-k similar users for that POI (Equation 2.6).

Instead of computing similarities for every pair of users, FCF computes similarities only for pairs of friends, and the predicted POI score is computed as follows:

$$\hat{a_{u,i}} = \frac{\sum_v sim_{u,v|(u,v)\in E} a_{u,i}}{\sum_v sim_{u,v|(u,v)\in E}} \tag{2.18}$$

where $sim_{u,v|(u,v)\in E}$ represents the similarity of friends u and v. Moreover, they extend the FCF model by modifying the similarity computation $sim_{u,v|(u,v)\in E}$. The intuition is that geographically close friends are more important than the ones far away. They propose to model the similarity $sim_{u,v|(u,v)\in E}$ between friends by their distance, and accordingly propose Geo-Measured Friend-based Collaborative Filtering (GM-FCF) which uses linear regression method on power-law distribution of distances between friends to learn a friend similarity model. The similarity is computed as follows:

$$sim_{u,v|(u,v)\in E} = \alpha d(u,v)^{\beta} \tag{2.19}$$

where $d(u,v)$ denotes the average distance of every pair of POIs of friends u and v, and α and β are parameters learned from the training data.

Since the number of pairs of friends is far less than the number of pairs of users, the FCF and GM-FCF model is more efficient than the traditional collaborative filtering method. However, the effectiveness of the FCF and GM-FCF models will be largely reduced because many non-friend similar users are ignored during the training process.

Gao's Model (2012)

Gao et al. analyze the geo-social correlation in location-based social networks in [20]. There are two types of check-ins: "existing check-ins", i.e., users check in at a previous visited POI, and "new check-ins", i.e., users check in at a new POI that the user has never checked in before.

For a given user, they argue that the effect of "new check-ins" largely depends on their social connections other than users with similar historical POIs. Particularly, for a specific user all other users can be divided into four groups: local friends (friends in short geographical distance) and distant friends (friends with long geographical distance), and local non-friends and distant non-friends. In order to model the geo-social correlations of "new check-in", the probability $P_u(i)^t$ of a users u checking in at a new POI i at time t is computed as follows:

$$P_u^t(i) = w_1 P1_u^t(i) + w_2 P2_u^t(i) + w_3 P3_u^t(i) + w_4 P4_u^t(i) \tag{2.20}$$

where with respect to the user u's four groups $P1, P2, P3, P4$ represent the conditional probability and w_1, w_2, w_3, w_4 represent the correlation strengths. Regression methods are proposed to learn Ps and ws accordingly. Experiments are conducted on a Foursquare data set, and a little improvement of POI recommendation accuracy is achieved. This approach explores the social network for POI recommendation to users, but their ad-hoc proposed model cannot improve the performance of standard collaborative filtering methods by much.

We argue that location-based social networks are different from social networks for movie recommendation, e.g., Flixster, and how to interpret and model the location-based social network remains a challenge.

2.4 Coordinates-based Approaches

Given a user-POI matrix and coordinates of POIs, [72, 13, 42, 44] build a POI recommender system that recommends POIs for users. Note that social networks are also given as input data in [72, 13], and we put them in this category because the modeling of social networks is exactly the same in social network-based approaches [71, 20].

[72, 13, 42, 44] focus on a prominent phenomenon of geographical clustering in location-based social networks. This geographical clustering phenomenon shows that a significant percentage of check-ins by the same user are within short distance, and indicates that there is *geographical influence* between users and POIs, i.e., users tend to visit nearby POIs.

In this line of work, the major contribution is to model the distance between users and POIs. In particular, each user is represented by a set of coordinates associated with POIs that she/he visited before. A probability $P_u(i)$ of a user u checking in a POI i is modeled as a power-law distribution as follows:

$$P_u(i) = \alpha d(u, i)^{\beta} \tag{2.21}$$

where α and β are parameters, and $d(u, i)$ represents the average distance between user u's historical POIs and POI i and is usually computed by the L_2 norm.

Furthermore, [72, 13, 42, 44] propose Collaborative Filtering (CF) approaches incorporating the modeling of geographical influence. Interestingly, experimental results show that CF approaches considering geographical influence significantly improve the recommendation accuracy over basic CF methods. The geographical influence factor is very important in location-based social networks.

2.4.1 Ye's Model (2011)

In [72] Ye et al. continue their previous work [71], and fuse the factors of user preferences, social influence, and geographical influence for POI recommendation. The factor of user preference is learned by a traditional memory-based collaborative filtering method introduced in Chapter 2 on the user-POI matrix. Besides, the factor of social influence indicates that the predicted score of an unvisited POI by a given user is computed by the weighted average score of the given user's friends as exactly the same in [71].

The major contribution of this paper is that they model the geographical influence between users and POIs suggesting POIs closer to the user's whereabout are more likely recommended to the user than distant POIs. Moreover, they propose an unified model linearly fusing the factors of user preferences, and social and geographical influences.

The same authors propose an extension model in [70] for POI recommendation to a group of users.

2.4.2 Cheng's Model (2012)

The authors of [13] have proposed a matrix factorization model considering geographical influence for POI recommendation in location-based social networks. Their model detects multiple centers for each user based on their history of POIs, and each center has 2-dimensional coordinates. In other words, the probability of a user's check-in at a POI is modeled as a Multi-center Gaussian distribution.

Motivated by the effect of geographical influence mentioned in [72], the probability of recommending a POI is inversely proportional to the distance between the POI and the user's center. Besides, the proposed model considers social influence, i.e., users tend to check in the POIs visited by their friends, and regularizes the users' latent factor by their friends.

2.4.3 Kurashima's Model (2013)

A recent work [42] extends Latent Dirichlet Allocation (LDA) for POI recommendation and addresses the spatial aspects of user check-ins by capturing the phenomenon of *geographical influence* in [72, 13]. Geographical influence suggests that POIs that are closer to the user's visited POIs are recommended with higher probabilities. The proposed Geo LDA

(GLDA) model assumes that the POI recommendations to a user should be geographically regularized by the set of all of the user's check-ins.

Compared to the model framework of [72, 13], the major contribution of this paper is to incorporate the geographical influence seamlessly into recommendation in a probabilistic way. Similar to LDA [11] as shown in Figure 2.1, the topic of check-ins is considered as a latent random variable. Topic distributions of users θ^{user} model the latent user interests, from which the topics of check-ins are sampled. Topics are associated with POI distributions ϕ^{topic}, which model the latent POI factors. Given the sampled topic, POIs are drawn from the POI distribution of that topic.

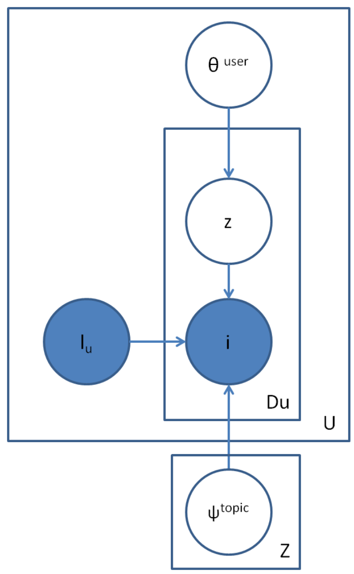


Figure 2.2: The graphical model of GLDA

GLDA extends LDA as shown in Figure 2.2, where l_u represents the set of all check-ins by user u, and the generated POI index is regularized by l_u. The regularization function is in Equation 2.22.

$$P(i|l_u) = \sum_{j \in l_u} \exp\big(-\frac{\beta}{2} d(l_i, l_j)\big) \tag{2.22}$$

where $d(l_i, l_j)$ represents the distance between POI i and j, and β is a parameter controlling the importance of the factor of geographical influence.

2.4.4 Liu's Model (2013)

Liu et al. propose a matrix factorization model in [44] for POI recommendation. Similar to [13], the proposed model considers multiple centers for each user based on their POI history, and POI recommendations to a user should be geographically regularized by the their geographical centers. The major difference between this work and [13] is that the geographical modeling of users is a part of the recommendation model, and the users' geographical modeling and POI recommendation are mutually reinforced by each other.

2.5 Text-based Approaches

Given a user-POI matrix and user-generated texts, [76, 75, 45, 73] build a POI recommender system that recommends POIs for users.

Early work [76, 75] is the first to address the POI recommendation problem using collaborative filtering methods. They propose recommender systems with data collection, extraction, offline modeling, and online recommendation. Textual labels such as "food and drinks" and "shopping" of POIs are used to group POIs into different types, and to study the recommendation results. For example, in [76], the recommendation results on POIs of "food and drinks" are more satisfying than the one on POIs of "shopping".

Recent work [45, 73] takes the advantage of online social media (e.g., Foursquare and Twitter), and they incorporate fruitful textual information such as tweets into collaborative filtering methods. In these works, texts are used to tackle the sparsity issue in location-based social networks, where many "cold start" users have few check-ins. These cold start users' preferences are not reliable due to insufficient training data. For example, a user checks in at a cinema. A basic collaborative filtering method ignores the texts like "movie, pop corn, and coke" associated with the cinema, and can barely learn a reliable user preference based on only one check-in. A text-based recommendation approach can detect the user interests of "movie", "pop corn", and "coke" from the text, and recommend cinemas and recreation centers for the given user with higher probabilities. Therefore, texts are considered as additional information for profiling user preferences.

2.5.1 Zheng's Model (2010)

Zheng et al. [76] propose a system in the architecture level for POI recommendation including data collection, extraction, and modeling. Their work focuses on GPS trajectory data, and they define a "stay point" as a region where a user stayed over a time threshold and within a distance range, and a "POI" is equivalent to a region that contains at least one stay points. They also extract activities from user posts or comments. For example, "delicious food" indicates an activity in the restaurant category. Consequently, a POI-activity matrix can be obtained through data extraction as well as a POI-feature matrix and an activity-activity matrix.

Activity recommendation for POIs and POI recommendation for activities are studied in their experiments. Based on the location-activity matrix, POI specific and activity specific latent factors are factorized by a collective matrix factorization model, and further regularized by the corresponding latent factors learned from the additional POI-feature matrix and activity-activity matrix.

The same authors propose a similar tensor factorization model for POI recommendation to users in [75].

2.5.2 Bao's Model (2012)

Bao et al. propose a recommendation model detecting user preferences and social opinions about POIs in [8]. In this paper, a user-POI matrix is given and POIs are associated with a category hierarchy, e.g., "Food", "Chinese Food", and "Sichuan Food" are the categories from the higher level to lower one. Offline modeling and online recommendation are two parts of the proposed model.

The goal of offline modeling is to learn the social knowledge and personal preference. Firstly, the model is to learn the social knowledge, i.e., the local experts per POI per category. For each category, there is a user-POI matrix. A Hypertext Induced Topic Search (HITS)-based inference model is used on the user-POI matrix to infer hub scores of users and authority scores of POIs. Hence the expertise of users is obtained by summing up all authority scores of the user visited POIs. Secondly, the model is to learn the personal preference. Since each POI is associated with a category, each user is associated with a set of category labels, e.g., "Food" and "Chinese Food", and "Arts" and "Metro Museum". Users are equivalent to documents. The normalized TF-IDF (Term Frequency-Inverse Document Frequency) of a category label is computed denoting the importance of the given category to the user.

In the online recommendation, the model selects POIs based on users' preferences, and uses a collaborative filtering method using the similarity comparison between the user and selected local experts.

2.5.3 Liu's Model (2013)

In [45] Liu et al. propose a two-stage model for topic modeling and POI recommendation. In the first stage, a topic model (LDA) is applied to detect topic distributions of users θ_u and POIs θ_i. In the second stage, they propose a probabilistic matrix factorization model incorporating the topic distribution of users and POIs. Specifically, the probability of a user u checking in a POI i is computed as follows:

$$P_u(i) = \boldsymbol{\phi}_u^T \cdot \boldsymbol{\phi}_i + \theta_u \cdot \theta_i \tag{2.23}$$

where $\boldsymbol{\phi}_u$ and $\boldsymbol{\phi}_i$ are the latent factors of user u and POI i introduced in Equation 6.2, and θ_u and θ_i are topic distributions of user u and POI i.

2.5.4 Yin's Model (2013)

Yin et al. propose a generative graphical model extending LDA in [73]. The generating probability of a POI by a user depends on the user preference and the POI's popularity. The sparsity issue mentioned in this paper is that few check-ins exist when users visit a new city other than their hometown. In this case, users' preferences are not reliable so that the model resorts POIs' popularity. Moreover, textual information of POIs are included in the model similar to [45]. Consequently, the user preferences are enhanced even the user only contains a few check-ins. Different from separate models in [45], this paper merges topic modeling of users and POIs and recommendation modeling into a single model.

2.6 Other Models

There are other models [6, 14, 15, 21, 16, 60, 51, 57] related to user movement analysis in location-based social networks. In this line of work, location prediction is the major task, i.e., given a set of geo-tagged check-ins from many users, the proposed model predicts the locations that users check in in the future. A difference from POI recommendation in Problem 3 is that the predicted locations can be either visited or unvisited by the users. Another difference is that these works focus on mining the user movement pattern using machine learning algorithms, such as Support Vector Machine or Markov Random Field, while the POI recommendation approaches profile user preferences using collaborative filtering methods. We should note that their study covers some interesting phenomenons in location-based social networks, such as the analysis of social and geographical influence in [6, 16], which are valuable for the study of POI recommendation.

The early work [6] analyzes the movement of Facebook users. After examining the relationship between proximity and friendship they find out that the likelihood of friendship drops monotonically as a function of distance, which indicates that friends tend to be geographically close to each other. Therefore, they propose a model to predict the user location by using their friends' locations, and the performance of the proposed model is better than the existing IP-based geolocation method.

Since most tweets are not associated with coordinates on Twitter, these works [14, 15, 21] address the following problem: given a set of geo-tagged posts from many users, learn a model of region specific words, and apply this model to predict the user location of un-tagged posts based on their content. Cheng et al. [14, 15] develop probabilistic methods to identify local words in tweets, and they predict user locations based on the local words in their tweets. Similarly, [21] proposes a Multinomial Naive Bayes model to predict the Twitter user profile's location at the granularity of the city level.

[16] studies the problem of modeling human mobility and location prediction in social networks. One of their interesting findings is that short-ranged travel is periodic and not affected by the social network, while long-distance travel is more influenced by social ties. Another interesting finding is that users tend to move within a small number of regions, e.g., around their home and office. Furthermore, [57] presents a probabilistic model incorporating social networks and achieves better performance for tweet location prediction. A recent work [51] proposes machine learning algorithms, such as, SVM and Decision Tree, tackling the location prediction problem.

Chapter 3

Spatial Topic Model

3.1 Introduction

Most of the location-based social networks such as Twitter and Yelp provide some additional information on top of the check-ins, including a set of user-generated texts related to the users and POIs. However, the existing collaborative-based recommendation methods [13, 42] ignore this additional information. We believe that using these user-generated texts (tweets or reviews) can improve the accuracy of POI recommendation.

The activities of users involve three major entities: user, post, and POI as described in Chapter 2. The interaction of these entities is the key to answer questions such as **w**ho will post a message **w**here and on **w**hat topic? In this chapter's work, we address the problem of profiling mobile users by modeling their activities, i.e., we explore topic modeling considering the spatial and textual aspects of user posts, and predict future user POIs.

Several works in the literature have addressed some of the above aspects. In recommender systems, [71, 13, 42] have proposed probabilistic matrix factorization models mining latent user and POI preferences to predict user POIs, but they totally ignore one of the key components: user posts. Another line of works [63, 74] has focused on user posts and proposed topic models to analyze geographical topics. Most recently, Hong et. al. [25] proposed a geographical topic model to capture language patterns of different regions and different users. Note that the users' distributions over regions are assumed to be independent from each other.

We observe that user movements sometimes correlate if two users have similar lifestyle or living routine. For example, many students from New York University live in the same neighborhood near the campus, and their movement trajectories correlate to each other. They may go to the same restaurants, coffee shops and grocery stores. Therefore, we argue that considering the movements of different users independently as in [25] is not the best way, and that we can predict a user's movement more accurately taking into account the movements of similar users. This idea underlies the paradigm of collaborative filtering.

A second observation is that user interest affects user movement not at the "syntactic" level of 2-dimensional coordinates but at the "semantic" level of places with a certain function. Existing spatial topic models with 2-dimensional coordinates do not distinguish the following two scenarios: 1) two users appear in the same POI, like a hockey themed bar, and 2) two users appear in two different POIs that are adjacent to each other, where one is a hockey themed bar and the other one is a facial salon. Intuitively, male users who are interested in sports often go to sport bars and watch games, while female users often go to facial salons. Two users in the first scenario share the same interest, while two users have totally different interests in the second scenario. As a result, without considering the fact that user movements are influenced not only by the coordinates of a POI but also by its function, the predictive ability of the model will be greatly reduced.

Motivated by the above observations, this chapter's work explores the following two questions:

1. How are user movements correlated to each other?

2. How does user interest affect user movement at the "semantic" level of POIs?

We propose a spatial topic model, called **ST** (**S**patial **T**opic), that takes the correlation of users' movements, and the correlation of user movement and user interest into account. As in existing models, a post is represented as an unordered collection of words (a bag-of-words assumption) associated with user and POI, which are all considered as observed random variables. Different from existing works [63, 74, 25], a POI in this chapter's work is defined as a place with a semantic functionality and with its 2D coordinates. A set of latent random variables is also defined, i.e., regions and topics are latent, and each post is assigned to a region and a topic. We assume that each POI is assigned to one and only one region, and its coordinates are generated by a 2-dimensional Gaussian distribution. For example, in New York City, regions could be areas that corresponded to community districts, such as Manhattan, Brooklyn, and Queens etc. Different from existing models, in order to generate a POI of a post by a particular user, the model considers the user's interest and the POIs of "similar" users. We develop a MCEM (Monte Carlo Expectation Maximization) method to learn the latent random variables and parameters that maximize the likelihood of the observed random variables, and the sparse coding technique is used to improve the efficiency of the learning method.

We perform experiments on two real life data sets from Twitter and Yelp. All posts (tweets and reviews) in the data sets are annotated with corresponding users and POIs. We evaluate the effectiveness of our proposed model and of state-of-the-art models in terms of accuracy of POI prediction, i.e., given a post and its author, we recommend top-k POIs to the user.

The major contributions of this chapter's work are as follows:

- We propose the first spatial topic model to capture the correlation between users' movements and between user interests and the function of POIs.
- We employ the sparse coding technique which greatly speeds up the learning process.
- Through comprehensive experiments, we demonstrate that our proposed model consistently improves the average precision@1,5,10,15,20 for POI recommendation by at least 50% (Twitter) and 300% (Yelp) compared to existing state-of-the-art recommendation algorithms and geographical topic models.

3.2 Spatial Topic Model

In this section, we first introduce the problem definition and then present our proposed **ST** (**S**patial **T**opic) model.

We assume that all the check-ins with attached documents are authored by a user from a fixed set of size U and all the words are from a fixed vocabulary V. We associate each user with a set of posts, and the set of posts of user u is denoted as D_u. Each post is represented by a set of words (the number of its words is denoted as $N_{u,d}$), and a pair of latitude and longitude coordinates. For convenience, we consider "tweet", "review", "post" and "document" as synonyms in this chapter's work. Formally, a document d is defined by $d = \{\mathbf{w}, u, i\}$, where w, u, i represents set of (index of) words, the index of user and POI respectively. l_i represents the coordinates of POI i. A document collection D is defined as a set of documents from all users. We assume that there is a set of latent topics and a set of latent regions in the document collection D. Each document d is assigned to one of the topics z_d and regions r_d. We use Z and R to denote the sets of topics and regions, respectively.

A semantically coherent topic in the document collection D is associated with a probability distribution over all words in the vocabulary, and a probability distribution over all POIs. A region has a geographical center, and it is comprised of a set of documents, which are coherent in topics and close to the center geographically. We assume that different users show different distributions over topics and regions. All notations described above are listed in Table 3.1. Note that we use capital letters to represent the sets and the $|.|$ sign to represent the size of the sets.

Table 3.1: Notations of input and output data

Variable	Interpretation
$w_{u,d,n}$	n^{th} word of the d^{th} document posted by the u^{th} user
$i_{u,d}$	POI index of the d^{th} document posted by the u^{th} user
l_i	latitude and longitude coordinates of the i^{th} POI
$z_{u,d}$	topic assignment of the d^{th} document posted by the u^{th} user
$r_{u,d}$	region assignment of the d^{th} document posted by the u^{th} user
Z	set of topics
R	set of regions
U	set of users
I	set of POIs
D_u	set of documents of user u
$N_{u,d}$	set of words in document d of user u
V	set of the vocabulary

Based on the above definitions, we formalize our research problem as follows:

Problem 4 (Spatial topic modeling) *Given a document collection D, and numbers $|Z|$ of topics and $|R|$ of regions, the task is to model and extract a set of topics and a set of regions.*

3.2.1 Model

To address our research problem, we propose a spatial topic model, called **ST** (**S**patial **T**opic), that takes the correlation of users' movements, and the correlation of user movement and user interest into account. Figure 3.1 shows the graphical model of ST.

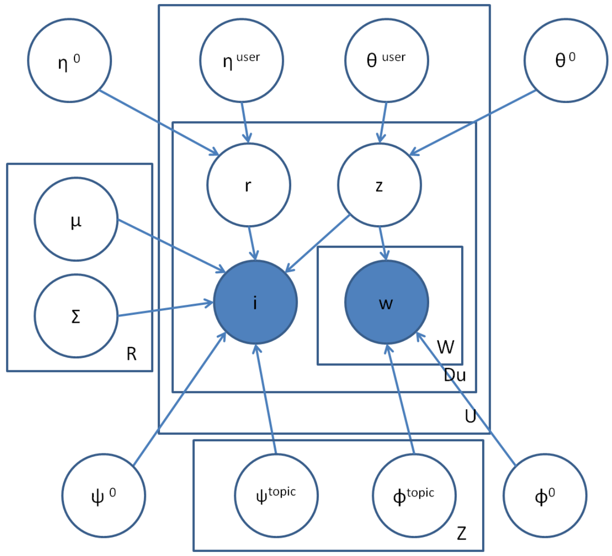


Figure 3.1: The graphical model of ST

We first introduce the notations of our model and listed in Table 3.1. Our input data, i.e., words and POIs, are modeled as observed random variables, shown as shaded circles in Figure 3.1, and we use $w_{u,d,n}$ and $i_{u,d}$ to denote them. l_i is a pair of latitude and longitude real values of i^{th} POI. Similar to existing models as in [74, 25], the topic and region index of documents are considered as latent random variables, which are denoted as $z_{u,d}$ and $r_{u,d}$ respectively. Users are associated with topic and region distributions, i.e., θ^{user} and η^{user}, from which the topics and regions of posts are sampled. Topics are associated with word distributions ϕ^{topic}. Given the sampled topic, words are drawn from the word distribution of that topic. The background distributions of words, topics, and regions are denoted as ϕ^0, θ^0, and η^0. All parameters are listed in Table 6.4.

Table 3.2: Notations of parameters

Variable	Interpretation
θ^0	topic distribution of the background
θ_u^{user}	topic distribution of the u^{th} user
ϕ^0	word distribution of the background
ϕ_z^{topic}	word distribution of the z^{th} topic
η^0	region distribution of the background
η_u^{user}	region distribution of the u^{th} user
ψ^0	POI distribution of the background
ψ_z^{topic}	POI distribution of the z^{th} topic
μ_r	region mean POI of the r^{th} region
Σ_r	region POI covariance of the r^{th} region

An important change of existing models is that in addition to generating the coordinates of posts, ST generates the index of the POI of posts. Another major change is that, to model the impact of user interest on user movement, ST assumes that the POI depends not only on the region but also on the topic. Consequently, it adds a POI distribution ψ^{topic} for each topic. Existing models assume that the 2D Gaussian distribution with center μ and covariance Σ of the sampled region governs the choice of POIs visited, i.e., the closer a POI to the center, the higher the probability of visiting that POI, and ST has the same assumption. Additionally, ST assumes that another important reason why the user visits the POI can be attributed to the user interests. Since POIs with different functions can have very similar coordinates, this assumption is much more meaningful when considering "semantic" POIs.

Particularly, users have different topic distributions, and topics have different POI distributions, so that the dependency between user interests and user POIs is transferred through the topic. ST captures the correlation between movements of different users, such that users who have similar movements share the same topics. Note that the topics serve a similar role as the latent factors in MF (Matrix Factorization). Different from the existing MF methods [13, 42], ST associates a word distribution with a topic so that it can describe the latent user and POI factors. Intuitively, collaborative filtering assumes that POIs A and B should both have high probabilities in POI distributions of some topic(s) in our case if many users frequently co-occur in both A and B. POI A and B do not necessarily have the same functionality. However, ST further assumes that POIs with high probabilities for the same topic should be cohesive in their functions, e.g., a topic with high probabilities for words like "coffee", "Java", and "mocha" should have high probability only for coffee shops. This design enables ST to detect users with similar interests and POIs with similar functions, and enables ST to better deal with "cold start" users, i.e., users who have very few posts, since the words of their few posts are more informative than their POIs.

Next, we describe the generative process of the ST model for a single document d.

- Draw a region index $r_{u,d}$
  - $r_{u,d} \sim p(r_{u,d}|u, \eta^0, \eta^{user})$
- Draw a topic index $z_{u,d}$
  - $z_{u,d} \sim p(z_{u,d}|u, \theta^0, \theta^{user})$

- Draw a POI index $i_{u,d}$, given the region index $r_{u,d}$ and topic index $z_{u,d}$
  - $i_{u,d} \sim p(i_{u,d}|r_{u,d}, z_{u,d}, \psi^0, \psi^{topic}, \mu, \Sigma)$
- Draw each word in d given the topic index $z_{u,d}$
  - $w_{u,d,n} \sim p(w_{u,d,n}|z_{u,d}, \phi^0, \phi^{topic})$

For each document, the ST model generates the POI and words consecutively. To generate a POI, the model first samples a region from the set of regions. To generate a region r, we use a multinomial distribution as follows:

$$p(r_{u,d}|u, \eta^0, \eta^{user}) = p(r_{u,d}|\eta^0 + \eta_u^{user}) \tag{3.1}$$

where η^0 is the global distribution of regions and η_u^{user} is the region distribution of user u. To simplify the notations, we use $p(r|\eta^0 + \eta_u^{user}) = \beta_{u,r}$. This approach employs the sparse coding technique introduced in the SAGE (Sparse Additive Generative) model [19]. The major advantage of SAGE is that it does not require additional latent "switching" variables when the model needs to take multiple factors into account. For example, in order to model topics, based on the background word distribution, for each topic SAGE models the difference in log-frequencies from the background word distribution instead of the log-frequencies themselves.

Each POI i is drawn depending on its corresponding region r and corresponding topic z. Given the sampled topic z and sampled region r, ST draws the POI $i_{u,d}$ as follows:

$$i_{u,d} \sim p(i_{u,d}|r_{u,d}, z_{u,d}, \psi^0, \psi^{topic}, \mu, \Sigma) = p(i_{u,d}|\psi^0 + \psi^{topic}_{z_{u,d}}) \times p(l_{i_{u,d}}|\mu_{r_{u,d}}, \Sigma_{r_{u,d}}) \tag{3.2}$$

where $p(i|\psi^0 + \psi_z^{topic}) = \delta_{z,i}$ and $p(l_i|\mu_r, \Sigma_r) = \mathcal{N}(l_i|r, \mu, \Sigma)$, which is the PDF of the Multivariate Gaussian distribution. This is the product of the probability of drawing the coordinates of the POI from the 2D Gaussian distribution μ_r, Σ_r of that region, and the probability of drawing the index of the POI from the POI distribution ψ_z^{topic} of that topic.

Similarly, for generating the topic and word index, the model uses a multinomial distribution considering the background and user topic distributions together, and the background and topic word distributions together, respectively as follows:

$$p(z_{u,d}|u, \theta^0, \theta_u^{user}) = p(z_{u,d}|\theta^0 + \theta_u^{user}) \tag{3.3}$$

$$p(w_{u,d,n}|z_{u,d}, \phi^0, \phi^{topic}) = p(w_{u,d,n}|\phi^0 + \phi^{topic}_{z_{u,d}}) \tag{3.4}$$

where $p(z|\theta^0 + \theta^{user}_u) = \alpha_{u,z}$ and $p(w_{u,d,n}|\phi^0 + \phi^{topic}_z) = \gamma_{z,w}$.

3.2.2 Parameter Learning

Our goal is to learn parameters that maximize the marginal log-likelihood of the observed random variables $\mathbf{i}, \mathbf{w}$. The marginalization is performed with respect to the latent random variables $\mathbf{z}, \mathbf{r}$, and it is hard to be maximized directly. Therefore, we apply the MCEM (Monte Carlo Expectation Maximization) algorithm to maximize the complete data likelihood $p(\mathbf{z}, \mathbf{r}, \mathbf{w}, \mathbf{i}|\Theta)$ in Equation 5.5 (see Figure 3.2), where $\Theta = \{\mu, \Sigma, \theta^0, \theta^{user}, \phi^0, \phi^{topic}, \eta^0, \eta^{user}, \psi^0, \psi^{topic}\}$.

$$\begin{aligned} p(\mathbf{z}, \mathbf{r}, \mathbf{w}, \mathbf{i}|\Theta) &= p(\mathbf{z}|\theta^0, \theta^{user}) \times p(\mathbf{r}|\eta^0, \eta^{user}) \times p(\mathbf{w}|\mathbf{z}, \phi^0, \phi^{topic}) \times p(\mathbf{i}|\mathbf{r}, \mathbf{z}, \mu, \Sigma, \psi^0, \psi^{topic}) \\ &= \textstyle\prod_{u=1}^{|U|} \prod_{d=1}^{|D_u|} \alpha_{u,z_{u,d}} \times \prod_{u=1}^{|U} \prod_{d=1}^{|D_u|} \beta_{u,r_{u,d}} \times \prod_{u=1}^{|U|} \prod_{d=1}^{|D_u|} \prod_{n=1}^{|N_{u,d}|} \gamma_{z_{u,d},w_{u,d,n}} \\ &\quad \times \textstyle\prod_{u=1}^{|U|} \prod_{d=1}^{|D_u|} \delta_{z_{u,d},i_{u,d}} \times \prod_{u=1}^{|U|} \prod_{d=1}^{|D_u|} \mathcal{N}(l_{i_{u,d}}|r_{u,d}, \mu, \Sigma) \end{aligned} \tag{3.5}$$

$$\alpha_{u,z} = \frac{\exp(\theta^0_z + \theta^{user}_{u,z})}{\sum_{zz=1}^{|Z|} \exp(\theta^0_{zz} + \theta^{user}_{u,zz})}, \beta_{u,r} = \frac{\exp(\eta^0_r + \eta^{user}_{u,r})}{\sum_{rr=1}^{|R|} \exp(\eta^0_{rr} + \eta^{user}_{u,rr})} \tag{3.6}$$

$$\gamma_{z,w} = \frac{\exp(\phi^0_w + \phi^{topic}_{z,w})}{\sum_{ww=1}^{V|} \exp(\phi^0_{ww} + \phi^{topic}_{z,ww})}, \delta_{z,i} = \frac{\exp(\psi^0_i + \psi^{topic}_{z,i})}{\sum_{ii=1}^{|I|} \exp(\psi^0_{ii} + \psi^{topic}_{z,ii})} \tag{3.7}$$

Figure 3.2: The joint probability of random variables given parameters in the ST model

According to the MCEM method, we sample the latent variables $\mathbf{r}, \mathbf{z}$ in the E step and maximize the parameters Θ in the M step. To sample a single variable $r_{u,d}$ given all other variables fixed, we use Equation 4.5. After $\mathbf{r}$ is sampled, we sample $z_{u,d}$ similarly according to Equation 5.6.

$$p(r_{u,d}|\mathbf{z}, \mathbf{r}_{-u,d}, \mathbf{w}, \mathbf{i}, \Theta) \propto \beta_{u,r_{u,d}} \times \delta_{z_{u,d},i_{u,d}} \times \mathcal{N}(l_i|r, \mu, \Sigma) \tag{3.8}$$

$$p(z_{u,d}|\mathbf{z}_{-u,d}, \mathbf{r}, \mathbf{w}, \mathbf{i}, \Theta) \propto \alpha_{u,z_{u,d}} \times \prod_{n=1}^{N_{u,d}} \gamma_{z_{u,d},w_{u,d,n}} \times \delta_{z_{u,d},i_{u,d}} \times \mathcal{N}(l_i|r, \mu, \Sigma) \tag{3.9}$$

Figure 3.3: The sampling formulas for latent variables $\mathbf{r}$,$\mathbf{z}$ in the ST model

In the M step, fixing all the latent variables $\mathbf{r}, \mathbf{z}$ that are sampled in the E step, we maximize the log likelihood of Equation 5.5 with respect to the parameters Θ. For variables μ and Σ, to obtain the maximum likelihood estimate, we take the derivative of its log likelihood with respect to μ_r and Σ_r, and set it to zero. Only one term in Equation 5.5 contains μ_r, so we use Equation 4.7 to update μ_r, where $\mathcal{I}(.)$ is an identity function, i.e. one where $r_{u,d}$ equals to r and zero otherwise, and $d(r)$ represents the number of documents assigned to region r. μ_r denotes the mean coordinates of POIs of the documents assigned to region r in the E step. We use Equation 3.11 to update the parameter Σ_r.

$$\mu_r = \frac{1}{d(r)} \sum_{u=1}^{|U|} \sum_{d=1}^{|D_u|} \mathcal{I}(r_{u,d} == r) l_{i_{u,d}} \tag{3.10}$$

$$\Sigma_r = \frac{1}{d(r) - 1} \sum_{u=1}^{|U|} \sum_{d=1}^{|D_u|} \mathcal{I}(r_{u,d} == r)(l_{i_{u,d}} - \mu_r)^T (l_{i_{u,d}} - \mu_r) \tag{3.11}$$

To update the other parameters, we use the gradient descent learning algorithm PSSG (Projected Scaled Sub-Gradient) [61], which is designed to solve optimization problems with L1 regularization on the parameters. More importantly, PSSG is scalable because it uses the quasi-Newton strategy with line search that is robust to common functions. According to the limited-memory BFGS [46] updates for the quasi-Newton method, the partial derivative functions of the parameters η^0, η^{user} are provided in the following Equations 4.8 and 4.9, where $d(u, r)$ represents the number of documents assigned to region r by user u, and $d(u)$ represents the number of documents by user u.

$$\frac{\partial L}{\partial \eta_r^0} = \sum_{u=1}^{|U|} d(u, r) - \sum_{u=1}^{|U|} \Big(d(u) \times \beta_{u,r} \Big) \tag{3.12}$$

$$\frac{\partial L}{\partial \eta_{u,r}^{user}} = d(u, r) - d(u) \times \beta_{u,r} \tag{3.13}$$

In Figure 5.2, where $d(u, z)$ represents the number of documents assigned to topic z by user u, $n(z, w)$ represents the number of words assigned to topic z, $n(z)$ represents the number of words assigned to topic z, and $d(z, i)$ represents the number of documents assigned to topic z at location i.

3.2.3 POI Recommendation

The ST model can be employed for POI recommendation as follows. Given a document with a user, our task is to recommend top-k "new " POIs, i.e., the POIs that the user has

$$\frac{\partial L}{\partial \theta_z^0} = \sum_{u=1}^{|U|} d(u,z) - \sum_{u=1}^{U} \Big(d(u) \times \alpha_{u,z} \Big) \tag{3.14}$$

$$\frac{\partial L}{\partial \theta_{u,z}^{user}} = d(u,z) - d(u) \times \alpha_{u,z} \tag{3.15}$$

$$\frac{\partial L}{\partial \phi_w^0} = \sum_{z=1}^{|Z|} n(z,w) - \sum_{z=1}^{Z} \Big(n(z) \times \gamma_{z,w} \Big) \tag{3.16}$$

$$\frac{\partial L}{\partial \phi_{z,w}^{topic}} = n(z,w) - n(z) \times \gamma_{z,w} \tag{3.17}$$

$$\frac{\partial L}{\partial \psi_i^0} = \sum_{z=1}^{|Z|} d(z,i) - \sum_{z=1}^{Z} \Big(d(z) \times \delta_{z,i} \Big) \tag{3.18}$$

$$\frac{\partial L}{\partial \psi_{z,i}^{topic}} = d(z,i) - d(z) \times \delta_{z,i} \tag{3.19}$$

Figure 3.4: The derivative equations for parameters θ^0, θ^{user}, ϕ^0, ϕ^{topic}, ψ^0, ψ^{topic} in the ST model

not visited in the training data set, which that user will visit. More precisely, given the words and author of a document d, the probability that author u visits POI i is computed as in Equation 5.13:

$$\begin{aligned} p(i|\mathbf{w},\Theta) \propto \sum_r^R \sum_z^Z p(\mathbf{w},i,z,r|\Theta) & \\ = \quad & \textstyle\sum_r^R \sum_z^Z p(z|\theta^0,\theta^{user}) \times p(r|\eta^0,\eta^{user}) \\ & \times p(\mathbf{w}|z,\phi^0,\phi^{topic}) \times p(i|z,r,\mu,\Sigma,\psi^0,\psi^{topic}) \end{aligned} \tag{3.20}$$

We rank the POIs in descending order of $p(i|\mathbf{w},\Theta)$.

3.3 Experiments

In this section, we experimentally evaluate the effectiveness of the ST (Spatial Topic) model, and we compare it against some baseline methods, one of the state-of-the-art POI recommendation methods [42], and one of the state-of-the-art geographical topic models [25]. We report our experimental results on Twitter and Yelp data sets, using the top-k average precision of POI recommendation for measuring the quality.

3.3.1 Data Sets

We report our experimental results on a Twitter data set downloaded from [14][1]. We extract a data set from a representative city in the US: NYC (New York City), where all tweets contain a POI label and geographical coordinates. To determine the coordinates of the POI, we use the mean of the coordinates of all tweets associated with a POI. Hence each POI corresponds to a unique mean coordinate, and each tweet of that POI has the same coordinates. Another data set is from Yelp, and it is publicly available[2]. It is from a US city – Phoenix. In the Yelp data set, each review has a POI (being reviewed) that is associated with a unique pair of latitude and longitude coordinates. Note that Twitter users often check in at the same POI multiple times, while Yelp users write reviews for a POI only once.

In the pre-processing steps, texts are processed by tokenizing on whitespace and punctuations, while we remove the URLs starting with "http" and user names starting with "@". Then we remove all texts with non-latin characters, followed by removing stop words, and the words with occurrences less than 100. To reduce noise, we remove both users and POIs with less than 10 posts. Some statistics about the data sets are presented in Table 6.1.

Table 3.3: Statistics of data sets from New York City on Twitter and Phoenix on Yelp.

#	Twitter	Yelp
Unique users	9,508	3,963
Posts	607,885	107,981
POIs	3,518	2,951
Avg. posts/user	64.93	27.24
Avg. posts/POI	172.79	36.59

Note that our data sets are much larger than the ones used in [42]. Another related work [25] uses data sets from all over the world, while our data sets are at the city level. From this point of view, the size of these two data sets is comparable or larger than the ones in [25].

[1] http://infolab.tamu.edu/data/

[2] https://www.yelp.com/dataset_challenge/

3.3.2 Experimental Setup

In our data sets, we randomly select 70% of observed data for each user as the training data, and the remaining 30% as the test data. We focus on the task of POI recommendation for users based on each document, which is by far the most commonly used performance measure for spatial topic model in the literature [71, 13, 42]. In particular, we train models in the training data set, and recommend the POIs based on posts by users in the test data set. Precision@k (top-k average precision, As introduced in Chpater 3, is used to evaluate the methods. The top-k precision for a test post is $\frac{1}{k}$ if its POI is among the top-k recommendations, and zero otherwise. The precision@k is the average top-k precision over all test posts.

Comparison Partners. In our experiments, we evaluate the following comparison partners, which all model (can predict) either the coordinates or index of POIs:

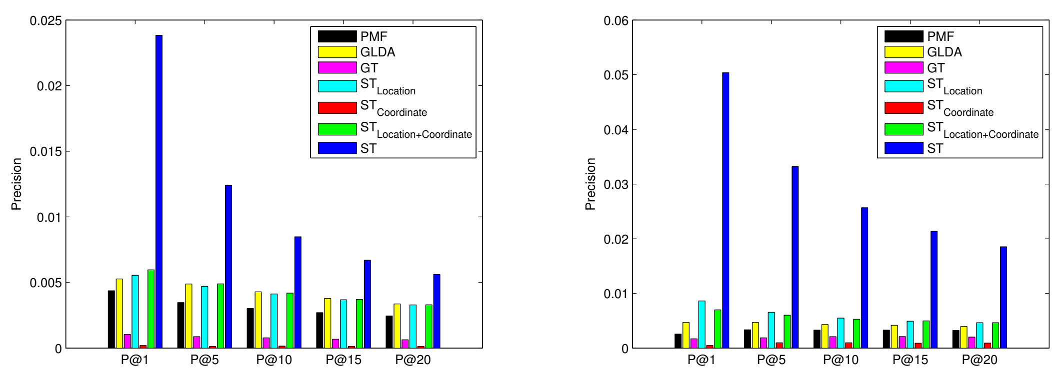
- Probabilistic Matrix Factorization (PMF). This is a well-known model in matrix factorization in [58].
- Geo Latent Dirichlet Allocation ($GLDA$). This is the modified LDA model, which is one of the state-of-the-art methods for POI recommendation proposed in [42].
- Geographical Topic (GT). This is one of the state-of-the-art geographical topic models proposed in [25].
- $ST_{location}$ (ST_{loc} for short). This is a simplified version of the ST model, where we remove the posts, and the only observed variable is the index of POIs $\mathbf{i}$ and the only latent variable is the topic $\mathbf{z}$. Note that this model is equivalent to an LDA model that generates index of POIs instead of words.
- $ST_{coordinate}$ (ST_{coo} for short). This is a simplified version of the ST model. Similar to ST_{loc}, we remove the posts from the data. Instead of generating the index of POIs, ST_{coo} generates the coordinates of POIs $\mathbf{l}$, and the only latent variable is the region $\mathbf{r}$.
- $ST_{coordinate+location}$ ($ST_{loc+coo}$ for short). This is another simplified version of the ST model, that generates both the coordinates and index of POIs, and the latent variables are the topic $\mathbf{z}$ and region $\mathbf{r}$. The only difference between this model and the full ST model is the lack of words.

- *ST*. This is the spatial topic model proposed in this chapter's work.

Note that there are other existing models [64, 63, 74] proposed for geographical topic modeling. We do not compare against them because the GT model proposed in [25] is a generalization of the existing models, and it performs better than the existing models in terms of location prediction in the experiments of [25]. We do not compare against [13], since it is similar to GLDA, which is the most recent work [42] on POI recommendation.

3.3.3 Experimental Results

For POI recommendation, Figure 3.5(a) and 3.5(b) show the precision@1,5,10,15,20 results of the comparison partners in the Twitter and Yelp data sets. Note that the number of topics and regions is set to 30 and 20. We observe that our ST model consistently and



(a) Twitter data set.

(b) Yelp data set.

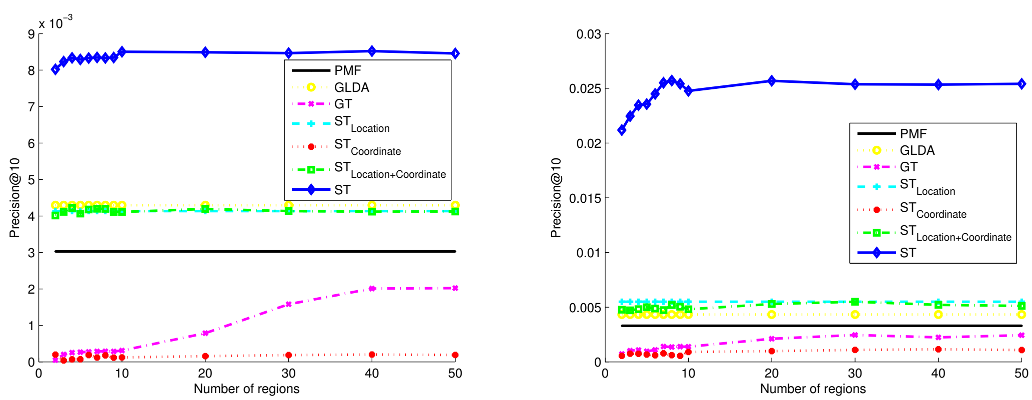
Figure 3.5: Precision@1,5,10,15,20 of comparison partners.

drastically outperforms all other models on both data sets. Compared to the state-of-the-art methods, GLDA and GT, in the areas of recommender systems and geographical topic modeling, ST improves the precision@20 by 50% (Twitter) and 300% (Yelp), and the gain is even higher for smaller values of k. This indicates that modeling the user interests and the correlation of user movements can help improve the accuracy of POI recommendation.

We also observe that the precision difference between ST and other models on Yelp is much larger than on Twitter. We argue that this is because 1) the posts on Yelp are much longer than on Twitter; 2) the words used on Yelp are more formal than on Twitter. As a result, it is easier to capture the user interests on Yelp than on Twitter.

We further analyze the contributions of different components in ST, by comparing the performance of ST and its simplified versions: ST_{loc}, ST_{coo}, and $ST_{loc+coo}$. We observe that modeling the index (semantics) of POIs in ST_{loc} is much more precise than modeling the coordinates of POIs in ST_{coo}. Comparing ST and $ST_{loc+coo}$, we see that the user interests expressed in the posts indeed enable more accurate POI recommendation. Furthermore, we observe that $ST_{loc+coo}$ clearly outperforms ST_{coo}, demonstrating the contribution of exploiting the correlation of user movements.

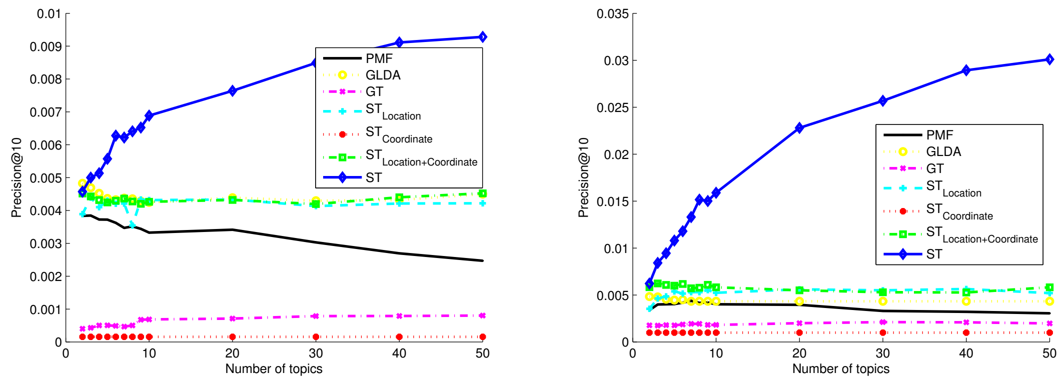
To analyze the impact of the input parameters, we show the precision@10 of the comparison partners for different numbers of regions (see Figure 3.6(a) and 3.6(b)) and topics (see Figure 3.7(a) and 3.7(b)). The results for precision@1,5,15,20 are similar to the results for precision@10. We observe that ST consistently outperforms the other comparison partners for all number of regions and topics. Furthermore, as the number of regions increases, the precision@10 of ST and GT increases and reaches a peak at first, and it plateaus when the number of regions reaches 10 or 20. Similarly, as the number of topics increases, the precision of ST increases. Some models, such as PMF, $GLDA$ and ST_{loc}, do not take the number of regions as their input, so that their precision is constant in Figure 3.6. Overall, the results of ST are relatively robust to the choice of the input parameters.



(a) Twitter data set.

(b) Yelp data set.

Figure 3.6: Precision@10 of the comparison partners for different number of regions. The number of topics is set to 30.



(a) Twitter data set. (b) Yelp data set.

Figure 3.7: Precision@10 of the comparison partners for different number of topics. The number of regions is set to 20.

3.4 Conclusion

In this chapter's work, we address the problem of spatial topic modeling in online social media, such as Twitter and Yelp, for user-generated content with POI. Previous work has explored topic models and recommendation algorithms that model either user and POI, or user and post, but they do not consider all of them together. We propose the first spatial model to capture spatial and textual aspects of posts, as well as user profiles in a single topic model, called Spatial Topic (ST) model. ST exploits the interdependencies between user movements, and between user interests and user movements. More specifically, ST is based on the intuition that 1) users' movements correlate with each other; 2) users' interests affect the movements of users. We argue that taking the correlation of users' movements, and the correlation of user movement and user interest into account enables a more accurate discovery of relevant regions and topics. We present the graphical model of ST and a corresponding method of parameter learning. We perform an experimental evaluation on Twitter and Yelp data sets from New York City and Phoenix. We compare ST against a state-of-the-art geographical topic model and a state-of-the-art recommendation method in terms of POI recommendation. Our experiments demonstrate drastically improved performance in POI recommendation.

Chapter 4

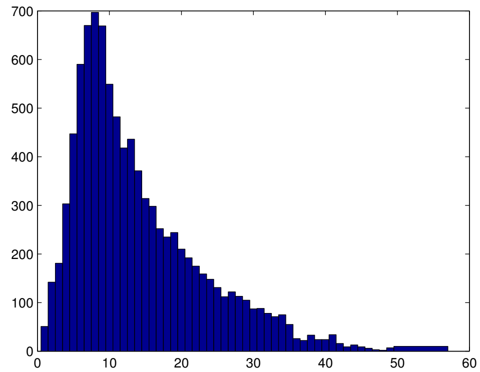
Spatio-Temporal Topic Model

4.1 Introduction

A recent work [42] extends LDA (Latent Dirichlet Allocation) for POI recommendation and addresses the spatial aspects of user check-ins by capturing the phenomenon of *geographical influence* [72]. Geographical influence suggests that POIs that are closer to the user's visited POIs are recommended with higher probabilities, and the existing methods [72, 42] assume that the POI recommendations to a user should be geographically regularized by the set of all of the user's check-ins. However, this assumption does not hold when the check-ins of a user are spread over multiple regions. Let us take an example of a user, who commutes between two cities (regions). A good recommendation should be one of those POIs in either one of the regions, but the existing models will recommend POIs along the commute route, since they are on average closer to the user's check-ins in both regions.

Thus, our first observation is that users may have multiple visited regions. To further illustrate this point, let us take an example of a collection of check-ins from NYC (New York City) on Twitter with 9,508 users, 3,518 POIs and 607,885 check-ins (the details of the data set are presented in Section 5.3.1). Figure 4.1(a) shows an example of a user with all his/her check-ins, which shows that this user has three frequently visited regions (Newark, Jersey City and Lower Manhattan). To further analyze the number of regions that users have visited, we cluster all check-ins into 100 regions by k-means based on their coordinates and plot the histogram in Figure 4.1(b), where the x-asis represents the number of regions and the y-asis represents the number of users. We observe that users have visited 14 regions on average, and most users have visited 10 to 20 regions.

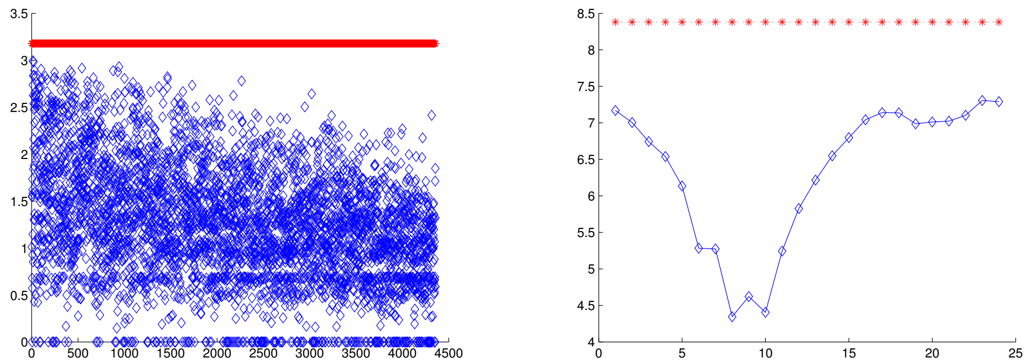
(a) Check-ins of a sample user are from three regions (Newark, Jersey City and Lower Manhattan).



(b) The histogram of the number of users versus the number of regions that users have visited. X-asis represents the number of regions and y-asis represents the number of users.

Figure 4.1: An example of check-ins in New York City on Twitter.

Our second observation is that different POIs have different "temporal activity patterns", i.e., probability distributions of check-ins over relative timestamps. For example, we use 1-24 to represent the time of the day in hours. Business, residence, and entertainment POIs have different daily activity patterns, and their daily activity normally peaks in the morning, evening, and late night, respectively. With temporal activity patterns, we can more accurately discover and profile POIs. In Figure 4.2(a), we plot the entropies of the daily activity patterns of all POIs in NYC from the same Twitter data set. Note that higher entropies indicate more uncertainty (more Uniform distribution of check-ins over the day) than lower ones, and the top red curve represents the entropies of Uniform distributions (maximum entropies). The figure shows that a large number of POIs have very small entropies, which implies that daily activity patterns do affect the check-ins at these POIs. Figure 4.2(b) shows the entropies of the distribution of check-ins over POIs for the 24 different hours, indicating that user movements are more predicable in the period from 5AM to 12AM than in the rest of the day. The reason is that users are likely to be at home or at work during that period of time. To sum up, we observe that different temporal activity patterns affect the decision of user check-ins at different POIs.



(a) The entropy of daily activity patterns at 3,518 different POIs. The red curve at the top in both Figure 4.2(a) and 4.2(b) represents the entropies of Uniform distributions (maximum entropies).

(b) The entropy of the distribution of check-ins over POIs for the 24 different hours.

Figure 4.2: Another example of check-ins in New York City on Twitter.

Based on the above observations, we propose the **S**patio-**T**emporal **T**opic (**STT**) model of check-ins that takes the geographical influence and temporal activity patterns into account, defining a probabilistic generative model. Basically, a check-in is represented by a user, a POI with a pair of coordinates, and a relative timestamp, which are all considered as observed random variables. Note that we can use 1-24 to represent relative timestamps in a day, and/or 1-7 to represent relative timestamps in a week. Similar to LDA, a set of latent topics is defined. Each user is associated with a probability distribution over topics, which captures the user interests, and each topic has a probability distribution over POIs, which captures the semantic relationship between POIs. Topics are assumed to represent sets of POIs that have similar functions such as parks, night clubs, or restaurants. Each check-in is assigned to a topic. [42] extends LDA and generates the POI of a check-in based on the POI distribution of the assigned topic and the regularization of the coordinates of all the user's check-ins. To model multiple regions of users, STT assumes that there is a set of latent regions, and each user is associated with a probability distribution over regions. Instead of regularizing POIs to be close to all the user's check-ins, STT regularizes them to the center of the sampled region. Additionally, STT considers temporal activity patterns. It selects a topic of a check-in based on its user's and time's topic distributions, and it generates (recommends) a POI based on the topic and time dependent POI distributions.

We propose an EM (Expectation-Maximization) algorithm to learn the latent random variables and parameters of STT that maximizes the likelihood of observed random variables. We perform experiments on real life data sets from Twitter, Gowalla and Brightkite. We evaluate the effectiveness of STT and of state-of-the-art models in terms of the perplexity of the test data set, and the precision of POI and time recommendation.

The major contributions of this chapter's work are as follows:

- We propose the first spatio-temporal topic model for POI and time recommendation, capturing the geographical influence between user regions and POIs, and temporal activity patterns of different topics and POIs.
- We employ the sparse coding technique which greatly speeds up the learning process.
- Through comprehensive experiments, we demonstrate that the proposed STT model consistently improves the test perplexity, the average precision@1,5,10 for POI recommendation, and the average precision@1,2,3 of time recommendation compared to existing state-of-the-art recommendation algorithms and geographical and temporal topic models.

4.2 Spatio-Temporal Topic Model

In this section, we first introduce the problem definition and then present our proposed **STT** (**S**patio-**T**emporal **T**opic) model.

4.2.1 Problem Definition

We first introduce the notations needed in our problem and listed in Table 6.4. We assume that all the check-ins are authored by a user from a fixed set U with size $|U|$. We associate each user u with a set of check-ins D_u, and each check-in is represented by a user, a POI with a pair of latitude and longitude coordinates, and a timestamp. Formally, a check-in d is defined by $d = \{u, i, l_i, t\}$, where u, i, t represents the (index of) user, POI, and timestamp, respectively. l_i represents the coordinates of POI i, and the values of t are discrete, e.g., 1-7 represent Sunday, Monday, ..., Saturday in a week and 1-24 represent hour 1, 2, ..., 24 in a day. A check-in collection D is defined as a set of check-ins from all users.

We assume that there is a set of latent topics Z and a set of latent regions R in the collection D. Each check-in d is assigned to one of the topics (z_d) and regions (r_d). To model user interests and movements, users are associated with topic distributions θ^{user} and region distributions η^{user}. A "semantically" coherent topic in the collection D is associated with a probability distribution over all POIs ψ^{topic}. Additionally, we use θ^{time} to represent time dependent topic distributions, e.g., check-ins from a night club topic usually happen on weekends and at late hours of the day, and ψ^{time} to represent time dependent POI distributions. A region has a geographical center μ, and it is associated with a set of check-ins, which are coherent in topics and close to the center geographically. Finally, θ^0, η^0, and ψ^0 represent the background distributions for topics, regions, and POIs, respectively.

Table 4.1: Notations of parameters

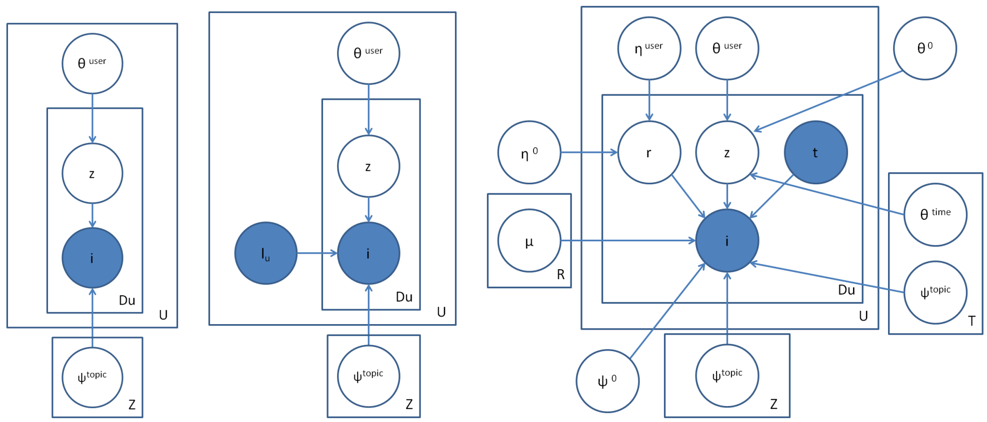
Variable	Interpretation
$i_{u,d}$	POI index of the d^{th} check-in by the u^{th} user
l_i	latitude and longitude coordinates of the i^{th} POI
$t_{u,d}$	relative timestamp in the d^{th} check-in by the u^{th} user
$z_{u,d}$	topic assignment of the d^{th} check-in by the u^{th} user
$r_{u,d}$	region assignment of the d^{th} check-in by the u^{th} user
U	set of users
I	set of POIs
T	set of times
Z	set of topics
R	set of regions
D_u	set of check-ins of the u^{th} user
θ^0	topic distribution of the background
θ_u^{user}	topic distribution of the u^{th} user
θ_t^{time}	topic distribution of the t^{th} timestamp
η^0	region distribution of the background
η_u^{user}	region distribution of the u^{th} user
ψ^0	POI distribution of the background
ψ_z^{topic}	POI distribution of the z^{th} topic
ψ_t^{time}	POI distribution of the t^{th} timestamp
μ_r	region mean of the r^{th} region

Based on the above definitions, we formalize our research problem as follows:

Problem 5 (Spatio-Temporal Topic Modeling for POI and Time Recommendation) *Given a check-in collection D, and numbers $|Z|$ of topics and $|R|$ of regions, the task is to model and learn the spatio-temporal parameters of users, topics, regions and POIs for POI and time recommendation.*

4.2.2 Model

Figure 4.3(c) shows the graphical model of STT. POIs **i** and timestamps **t** are modeled as observed random variables, shown as shaded circles, while the latent random variables of topics **z** and regions **r** and all parameters listed in Table 6.4 are shown as unshaded circles.



(a) LDA. (b) GLDA. (c) STT.

Figure 4.3: The graphical models

Similar to LDA [11] as shown in Figure 4.3(a), the topic of check-ins is considered as a latent random variable. Topic distributions of users θ^{user} model the latent user interests, from which the topics of check-ins are sampled. Topics are associated with POI distributions ϕ^{topic}, which model the latent POI factors. Given the sampled topic, POIs are drawn from the POI distribution of that topic. GLDA [42] extends LDA as shown in Figure 4.3(b), where l_u represents the set of all check-ins by user u, and the generated POI index is regularized by l_u.

We observe that users have visited multiple regions, which has not been considered by existing models. Therefore, as shown in Figure 4.3(c), an important difference between STT and existing models is that users are associated with region distributions η^{user}, and the regions of check-ins are sampled from the mixture of users' region distributions η^{user} and a background region distribution η^0. Instead of regularizing POIs to be close to all the user's check-ins, STT regularizes them to the sampled region center μ_r. This model design is more meaningful when observing that users have visited multiple regions.

Another major difference in STT is that it models the impact of temporal activity patterns on POI recommendations. Given a timestamp t of check-in d, STT assumes that the topic $z_{u,d}$ depends not only on the user u's topic distribution θ_u^{user} but also on the time's topic distribution θ_t^{time}. Moreover, the probability of generating (recommending) the index of POI $i_{u,d}$ depends on the given time's POI distribution ϕ_t^{time}. STT also models the popularity of POIs by using the background distribution of POIs ψ^0.

Next, we describe the generative process of the STT model for a single check-in d of a given user u and time $t_{u,d}$.

- Draw a topic index $z_{u,d}$
  - $z_{u,d} \sim p(z_{u,d}|u, t_{u,d}, \theta^0, \theta^{user}, \theta^{time})$
- Draw a region index $r_{u,d}$
  - $r_{u,d} \sim p(r_{u,d}|u, \eta^0, \eta^{user})$
- Draw a POI index $i_{u,d}$, given the region index $r_{u,d}$ and topic index $z_{u,d}$
  - $i_{u,d} \sim p(i_{u,d}|r_{u,d}, z_{u,d}, t_{u,d}, \psi^0, \psi^{topic}, \psi^{time}, \mu)$

For each check-in, the STT model first samples a topic from the set of topics. To generate a topic z, the model uses multinomial distributions of the background, user, and time's topic together as follows:

$$p(z|u, t, \theta^0, \theta^{user}, \theta^{time}) = \frac{\exp(\theta_z^0 + \theta_{u,z}^{user} + \theta_{t,z}^{time})}{\sum_{zz=1}^{|Z|} \exp(\theta_{zz}^0 + \theta_{u,zz}^{user} + \theta_{t,zz}^{time})} \tag{4.1}$$

where θ^0 is the background topic distribution, and θ^{user} and θ^{time} are the topic distributions of user u and time t, respectively. To simplify the notations, we use $p(z|u, t, \theta^0, \theta^{user}, \theta^{time}) = \alpha_{u,t,z}$. This approach employs the sparse coding technique introduced in the SAGE (Sparse Additive Generative) model [19]. The major advantage of SAGE is that it does not require additional latent "switching" variables when the model needs to take multiple factors into account. For example, in order to model a mixture topic distribution of background, user, and time factors, based on the background topic distribution θ^0, it models the difference in log-frequencies, e.g., θ^{topic} and θ^{time}, from the background topic distribution θ^0, instead of the log-frequencies themselves.

Similarly, for generating the region index, we use a multinomial distribution as follows:

$$p(r|u, \eta^0, \eta^{user}) = \frac{\exp(\eta^0_r + \eta^{user}_{u,r})}{\sum_{rr=1}^{|R|} \exp(\eta^0_{rr} + \eta^{user}_{u,rr})} \tag{4.2}$$

where η^0 is the global distribution of regions and η^{user}_u is the region distribution of user u. To simplify the notations, we use $p(r|u, \eta^0, \eta^{user}) = \beta_{u,r}$.

Each POI index i is drawn depending on the sampled topic z and sampled region r as follows:

$$\begin{aligned} p(i|r, z, t, \psi^0, \psi^{topic}, \psi^{time}, \mu) & \\ &= p(i|\psi^0, \psi^{topic}_z, \psi^{time}_t) \times p(l_i|\mu_r) \\ &= \frac{\exp(\psi^0_i + \psi^{topic}_{z,i} + \psi^{time}_{t,i}) \exp(-\frac{\rho}{2}||\mu_r - l_i||)}{\sum_{ii=1}^{|I|} \exp(\psi^0_{ii} + \psi^{topic}_{z,ii} + \psi^{time}_{t,ii}) \exp(-\frac{\rho}{2}||\mu_r - l_{ii}||)} \end{aligned} \tag{4.3}$$

where $p(i|r, z, t, \psi^0, \psi^{topic}, \psi^{time}, \mu) = \delta_{z,r,t,i}$. The probability of a POI index is the product of the probability of drawing the index of the POI from the mixture of POI distributions $\psi^0 + \psi^{topic}_z + \psi^{time}_t$, and the probability of drawing the coordinates l_i of the POI i, which is inversely proportional to the distance between μ_r and $l_{i_{u,d}}$, i.e., the L^2-norm $||\mu_r - l_i||$. ρ controls the trade-off between the geographical factor and the topic and time factors. By increasing the value of ρ, the model gradually puts more weights on the geographical influence and recommends more POIs nearby. When the value of ρ decreases, the model recommends more POIs based on the topic and time factors.

4.2.3 Parameter Learning

Our goal is to learn parameters that maximize the marginal log-likelihood of the observed random variables $\mathbf{i}, \mathbf{t}$. The marginalization is performed with respect to the latent random variables $\mathbf{z}, \mathbf{r}$, and it is hard to be maximized directly. Therefore, we apply the MCEM (Monte Carlo Expectation Maximization) algorithm to maximize the complete data likelihood $p(\mathbf{z}, \mathbf{r}, \mathbf{i}|\mathbf{t}, \mathbf{u}, \Theta)$ in Equation 5.5, where $\Theta = \{\theta^0, \theta^{user}, \theta^{time}, \eta^0, \eta^{user}, \psi^0, \psi^{topic}, \psi^{time}, \mu\}$.

$$\begin{aligned} p(\mathbf{z}, \mathbf{r}, \mathbf{i}|\mathbf{t}, \mathbf{u}, \Theta) & \\ &= p(\mathbf{z}|\mathbf{u}, \mathbf{t}, \theta^0, \theta^{user}, \theta^{time}) \times p(\mathbf{r}|\mathbf{u}, \eta^0, \eta^{user}) \\ &\times p(\mathbf{i}|\mathbf{r}, \mathbf{z}, \mathbf{t}, \mu, \psi^0, \psi^{topic}, \psi^{time}) \\ &= \prod_{u=1}^{|U|} \prod_{d=1}^{|D_u|} \alpha_{u,t_{u,d},z_{u,d}} \times \prod_{u=1}^{|U|} \prod_{d=1}^{|D_u|} \beta_{u,r_{u,d}} \\ &\times \prod_{u=1}^{|U|} \prod_{d=1}^{|D_u|} \delta_{z_{u,d},r_{u,d},t_{u,d},i_{u,d}}, \end{aligned} \tag{4.4}$$

where $\alpha_{u,t,z}, \beta_{u,r}, \delta_{z,r,t,i}$ are shown in Equation 5.2, 5.3, and 5.4.

According to the MCEM method, we sample the latent variables $\mathbf{r}, \mathbf{z}$ in the E step and maximize the parameters Θ in the M step. To sample a single variable $r_{u,d}$ given all other variables fixed, we use Equation 4.5. After $\mathbf{r}$ is sampled, we sample $z_{u,d}$ similarly according to Equation 5.6.

$$p(r_{u,d}|\mathbf{z}, \mathbf{r}_{-u,d}, \mathbf{i}, \mathbf{t}, \Theta) \propto \beta_{u,r_{u,d}} \times \delta_{z_{u,d},r_{u,d},t_{u,d},i_{u,d}} \tag{4.5}$$

$$p(z_{u,d}|\mathbf{z}_{-u,d}, \mathbf{r}, \mathbf{i}, \mathbf{t}, \Theta) \propto \alpha_{u,t_{u,d},z_{u,d}} \times \delta_{z_{u,d},r_{u,d},t_{u,d},i_{u,d}} \tag{4.6}$$

In the M step, fixing all the latent variables $\mathbf{r}, \mathbf{z}$ that are sampled in the E step, we maximize the log likelihood of Equation 5.5 with respect to the parameters Θ. To update the parameters, we use the gradient descent learning algorithm PSSG (Projected Scaled Sub-Gradient) [61].

For variable μ, we take the derivative of its log likelihood $L = \log(p(\mathbf{z}, \mathbf{r}, \mathbf{i}|\mathbf{t}, \mathbf{u}, \Theta))$ with respect to μ_r. Only one term $p(\mathbf{i}|\mathbf{r}, \mathbf{z}, \mathbf{t}, \mu, \psi^0, \psi^{topic}, \psi^{time})$ in Equation 5.5 contains μ_r, so we use Equation 4.7 to compute the partial derivative of μ_r, where $d(z, r, t, i)$ represents the number of check-ins assigned to topic z, region r, time t, and POI i, and $\exp(\psi)$ denotes the exponential summation of ψ^0, ψ^{topic} and ψ^{time}, i.e., $\exp(\psi) = \exp(\psi_i^0 + \psi_{z,i}^{topic} + \psi_{t,i}^{time})$.

$$\begin{aligned}\frac{\partial L}{\partial \mu_r} = \sum_{z=1}^{|Z|}\sum_{r=1}^{|R|}\sum_{t=1}^{|T|}\sum_{i=1}^{|I|} d(z,r,t,i) \times \left(\frac{-\frac{\rho}{2}(\mu_r - l_i)}{||\mu_r - l_i||}\right) - \sum_{z=1}^{|Z|}\sum_{r=1}^{|R|}\sum_{t=1}^{|T|} d(z,r,t) \\ \times \frac{\sum_{i=1}^{|I|}\exp(\psi)\exp(-\frac{\rho}{2}||\mu_r - l_i||)\left(\frac{-\frac{\rho}{2}(\mu_r - l_i)}{||\mu_r - l_i||}\right)}{\sum_{ii=1}^{|I|}\exp(\psi)\exp(-\frac{\rho}{2}||\mu_r - l_{ii}||)}\end{aligned} \tag{4.7}$$

Figure 4.4: The derivative equation for parameter μ_r.

According to the limited-memory BFGS [46] updates for the quasi-Newton method, the partial derivative functions of the parameters η^0, η^{user} are provided in the following Equations 4.8 and 4.9, where $d(u, r)$ represents the number of documents assigned to region r by user u, and $d(u)$ represents the number of documents by user u.

$$\frac{\partial L}{\partial \eta_r^0} = \sum_{u=1}^{|U|} d(u,r) - \sum_{u=1}^{|U|}\Big(d(u) \times \beta_{u,r}\Big) \tag{4.8}$$

$$\frac{\partial L}{\partial \eta_{u,r}^{user}} = d(u,r) - d(u) \times \beta_{u,r} \tag{4.9}$$

Similarly, we get derivative functions for the remaining parameters in Figure 5.2, where $d(u,z)$ represents the number of documents assigned to topic z by user u, $n(z,w)$ represents the number of words assigned to topic z, $n(z)$ represents the number of words assigned to topic z, and $d(z,i)$ represents the number of documents assigned to topic z at location i.

$$\frac{\partial L}{\partial \theta_z^0} = \sum_{u=1}^{|U|} d(u,z) - \sum_{u=1}^{|U|} \Big(d(u) \times \alpha_{u,z} \Big) \tag{4.10}$$

$$\frac{\partial L}{\partial \theta_{u,z}^{user}} = d(u,z) - d(u) \times \alpha_{u,z} \tag{4.11}$$

$$\frac{\partial L}{\partial \psi_i^0} = \sum_{z=1}^{|Z|} d(z,i) - \sum_{z=1}^{|Z|} \Big(d(z) \times \delta_{z,i} \Big) \tag{4.12}$$

$$\frac{\partial L}{\partial \psi_{z,i}^{topic}} = d(z,i) - d(z) \times \delta_{z,i} \tag{4.13}$$

Figure 4.5: The derivative equations for parameters θ^0, θ^{user}, ψ^0, ψ^{topic} in the STT model.

Good initializations of parameters can speed up the learning process towards the convergence of the objective function by reducing the number of iterations. We cluster all the check-ins into $|R|$ regions by k-means based on their coordinates, and initialize each region mean μ_r to the center of region r. According to the region assignment of each check-in, we set the background region distribution η_r^0 to the log-frequency of check-ins of the region r. Similarly, the background POI distribution ψ_i^0 is set to the log-frequency of check-ins of each POI i.

4.2.4 POI and Time Recommendation

The STT model can be employed for POI recommendation as follows. Given a check-in with a user, our task is to recommend top-k POIs, that user will visit in the future. More precisely, given the user u and time t of a check-in d, the probability that user u visits POI

i at time t is computed as in Equation 5.13:

$$p(i|t,u,\Theta) \propto \sum_r^R \sum_z^Z p(i,z,r|t,u,\Theta)$$
$$= \sum_r^R \sum_z^Z p(z|\theta^0,\theta^{user},\theta^{time}) \times p(r|\eta^0,\eta^{user}) \times p(i|z,r,t,\psi^0,\psi^{topic},\psi^{time},\mu) \quad (4.14)$$

For time recommendation, given the user u and POI i of a check-in d, the probability that user u visits POI i at time t is in Equation 4.15, where $p(t)$ is a prior and set to the log-frequency of check-ins of time t.

$$p(t|i,u,\Theta) \propto p(i|t,u,\Theta) \times p(t|u,\Theta) = p(i|t,u,\Theta) \times p(t) \quad (4.15)$$

We rank the POIs and times in descending order of $p(i|t,u,\Theta)$ and $p(t|i,u,\Theta)$, respectively.

4.3 Experiments

In this section, we experimentally evaluate the effectiveness of the STT (Spatio-Temporal Topic) model. We also compare it against some baseline methods, e.g., PMF and LDA, one of the state-of-the-art POI recommendation methods [42], and one of the state-of-the-art geographical topic models [25] and temporal topic models [66]. We report our experimental results on Twitter, Gowalla, and Brightkite data sets, and we evaluate them using the perplexity of the test data sets, and the top-k average precision for POI and time recommendation.

4.3.1 Data Sets

We used three publicly available data sets: a Twitter data set from [15][1], and Gowalla and Brightkite data sets from [16][2]. We generate subsets from a representative city NYC (New York City) in the US for Twitter, Gowalla, and Brightkite, where all check-ins contain a POI label and geographical coordinates. On Twitter, the coordinates of check-ins at the same POI may have some variance, so we use the mean of these coordinates to create the coordinates of the POI. Hence each POI corresponds to a unique mean coordinate, and each check-in of that POI has the same coordinates. Since our task is to recommend POIs to

[1]http://infolab.tamu.edu/data/

[2]http://snap.stanford.edu/data/

users, uninteresting POIs, such as home, should be removed. Therefore, in all three data sets we remove POIs checked in by less than 5 different users. Some statistics about the data sets are presented in Table 6.1.

Table 4.2: Statistics of data sets from New York City on Twitter, Gowalla and Brightkite.

#	Twitter	Gowalla	Brightkite
Unique users	9,508	5,588	1,820
POIs	3,518	4,358	348
Check-ins	607,885	89,294	34,710
Avg. check-ins/user	64.93	15.97	19.07
Avg. check-ins/POI	172.79	20.48	99.74

4.3.2 Experimental Setup

In our data sets, we randomly select 70% of observed data for each user as the training data, and the remaining 30% as the test data. We focus on the tasks of POI and time recommendation for users. POI recommendation is by far the most commonly used performance measure for spatial models in the literature [71, 42]. We train models in the training data set. For every check-in in the test set, we recommend a POI given the user and time of the check-in, and compare the recommended POI against the actual POI of the check-in. Additionally, making the recommendation at the right time (time recommendation) is important in recommender systems as mentioned in a pioneer work [65]. For time recommendation, for every check-in in the test set, we recommend a time for the user and POI of the check-in, and compare the recommended time against the actual time of the check-in.

Comparison Partners. In our experiments, we evaluate the following comparison partners, which all model (and can predict) either the coordinates or the index of POIs:

- Multi-Region (MR). This is a simplified version of the STT model. It assumes that users are associated with region distributions, and the coordinates of POIs are drawn from 2D Gaussian distributions. As a result, it generates the coordinates of POIs $\mathbf{l}$, and the only latent variable is the region $\mathbf{r}$. Intuitively, the MR model recommends closest POIs to users based on their regions.

- Probabilistic Matrix Factorization (PMF). This is a well-known matrix factorization model proposed in [58].

- Latent Dirichlet Allocation (*LDA*). This is a modified LDA model, where the only observed variable is the index of POIs $\mathbf{i}$ and the only latent variable is the topic $\mathbf{z}$. Note that this model is equivalent to the original LDA model that generates index of POIs instead of words.

- Geographical Topic (*GT*). This is one of the state-of-the-art geographical topic models proposed in [25].

- Topics Over Time (*TOT*). The TOT model is a temporal topic models proposed in [66], and it assumes that the continuous timestamp of a document is drawn from a topic-specific Beta distribution. Since we consider only relative timestamps in hours or days, the modified model assumes the timestamp is drawn from a Multinomial distribution.

- Geo LDA (*GLDA*). This is a spatial extension of the LDA model, which is one of the state-of-the-art methods for POI recommendation proposed in [42].

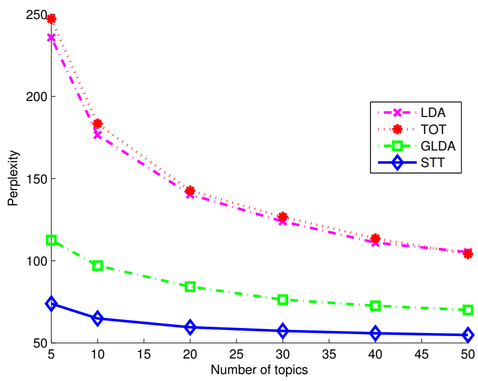
- Spatio-Temporal Topic (*STT*). This is the full spatio-temporal topic model proposed in this chapter's work. The default number of relative timestamps is 24 (hours). Optionally, we use STT_{week} to denote the model considering 7 timestamps (7 days).

Note that we do not compare against [71, 72, 13], since they are similar to GLDA [42], which is the most recent work on POI recommendation. Also, there are other existing models [64, 63, 74] proposed for geographical topic modeling. We do not compare against them because the GT model proposed in [25] is a generalization of the existing models, and it performs better than the existing models in terms of POI prediction in the experiments of [25]. GT cannot be applied to the Gowalla and Brightkite data sets, since they do not have texts associated with check-ins, which are the input of GT.

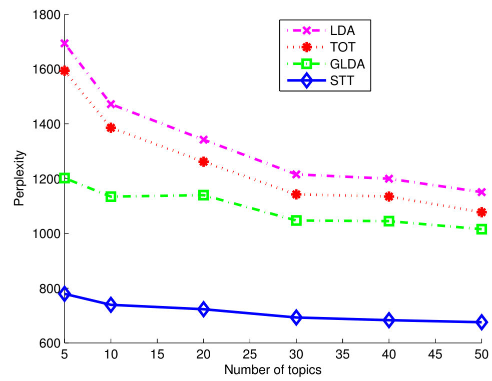
4.3.3 Experimental Results

Perplexity

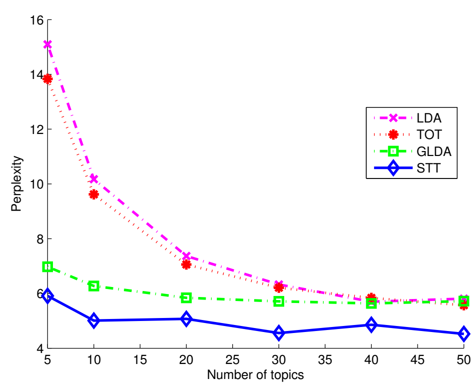
Figure 4.6 shows the perplexity of the comparison partners for different numbers of topics. For all models, the number of regions is set to 50 for the Twitter and Gowalla data sets and to 30 for the Brightkite data set. Figure 4.7 shows the perplexity of the comparison partners for different numbers of regions. The number of topics is set to 30 for all three data sets. To establish a fair comparison, we only compare the LDA, TOT, GLDA, and STT models, as they have the same observed random variables (the index of POIs and timestamps).



(a) Twitter data set.



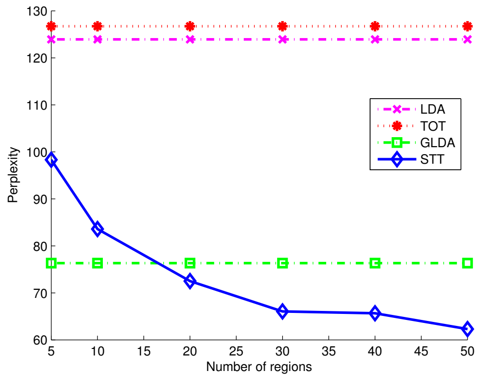
(b) Gowalla data set.



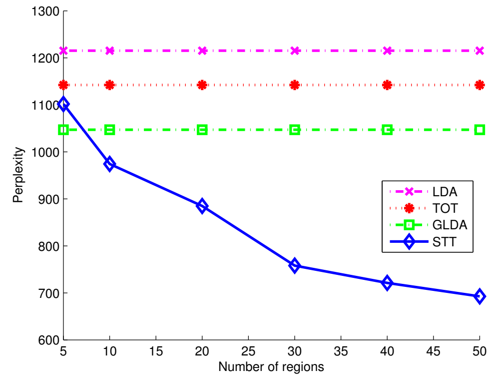
(c) Brightkite data set.

Figure 4.6: Perplexity of the comparison partners for 5,10,20,30,40,50 topics.

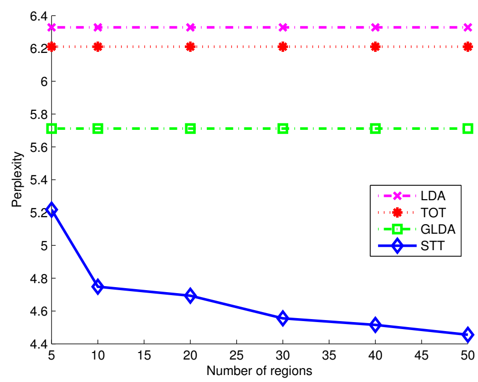
We observe that the STT model consistently achieves the smaller (better) perplexity than the LDA, TOT, and GLDA models in all the three data sets for different numbers of topics and regions, which means that the STT model fits the data better than the other models. As expected, we observe that the perplexity of all models decreases as the number of topics and regions increases.



(a) Twitter data set.



(b) Gowalla data set.



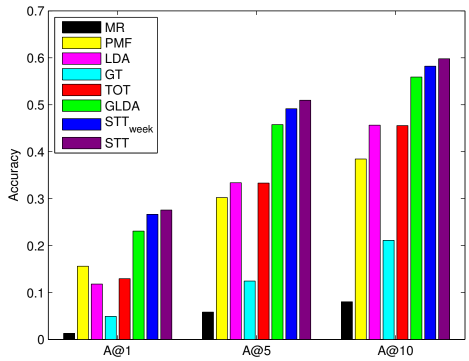
(c) Brightkite data set.

Figure 4.7: Perplexity of the comparison partners for 5,10,20,30,40,50 regions.

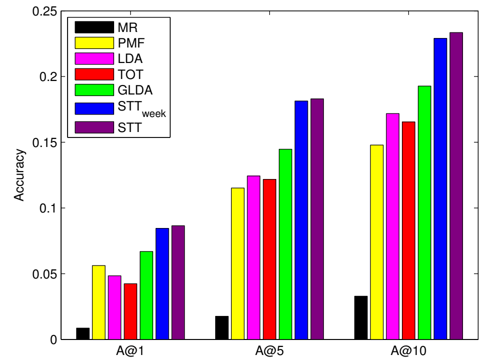
POI Recommendation

Figure 4.8(a), 4.8(b), and 4.8(c) show the precision@1,5,10 results for POI recommendation in the Twitter, Gowalla and Brightkite data sets, respectively.

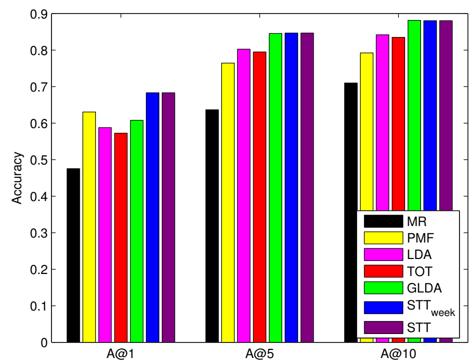
We observe that our STT and STT_{week} models consistently outperform all other models on all the three data sets. PMF and LDA yield similar results in all the data sets since they are conceptually analogous (as discussed in Chapter 2). Both MR and GT perform worse than PMF and LDA, because they do not consider the "semantic" meaning of the POIs and do not model the correlation between user movements. TOT is sometimes better or worse than PMF and LDA, because only the temporal aspect is considered, and it is dependent on only topics (not on both topics and POIs as in STT). GLDA outperform PMF and LDA, because it considers the spatial aspect of user check-ins. Compared to the most competitive



(a) Twitter data set.



(b) Gowalla data set.



(c) Brightkite data set.

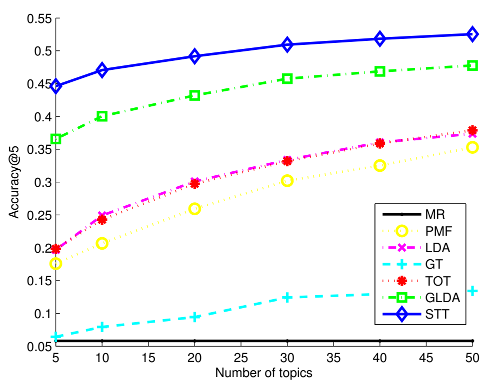
Figure 4.8: Precision@1,5,10 of POI recommendation.

and state-of-the-art method in the area of recommender systems, GLDA, STT improves the precision@1 by 19% (Twitter), 30% (Gowalla), and 12% (Brightkite). This indicates that modeling the multiple regions of users and the impact of temporal activity patterns can help improve the precision of POI recommendation. The results of STT are slightly better than those of STT_{week}, indicating that the timestamps in hours produce more distinctive temporal activity patterns than the ones in days.

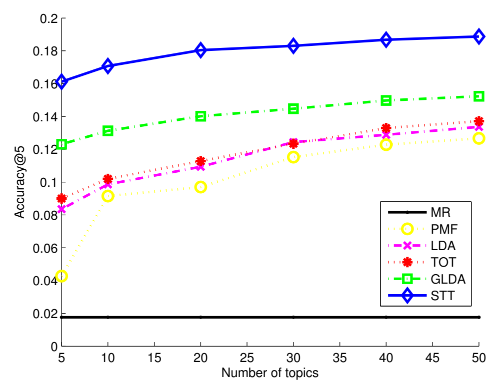
We also observe that the precision difference between STT and the other models on Twitter and Gowalla is larger than on Brightkite. We argue that as the number of POIs in the Brightkite data set is much smaller than in the other two data sets, the performance of baseline methods, such as MR, PMF, and LDA, is sufficiently good, so that the room for improvement is limited.

To analyze the impact of the input parameters, Figure 4.9 shows the precision@5 of the comparison partners for different numbers of topics, and Figure 4.10 shows the precision@5 of the comparison partners for different numbers of regions. The results for precision@1,10 are similar to the results for precision@5.

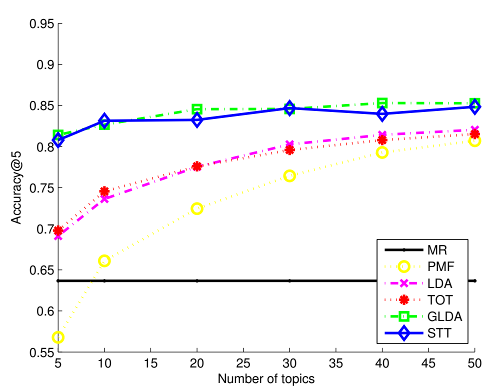
We observe that STT consistently outperforms the other comparison partners for all numbers of topics and regions. Furthermore, as the number of topics increases, the precision@5 of all the models increases and then plateaus when the number of topics reaches 20 or 30. Similarly, as the number of regions increases, the precision of STT increases. Some models, such as PMF, LDA, TOT, and GLDA, do not take the number of regions as their input, so that their precision is constant in Figure 4.10. Overall, the results of STT are fairly robust to the choice of the input parameters.



(a) Twitter data set.

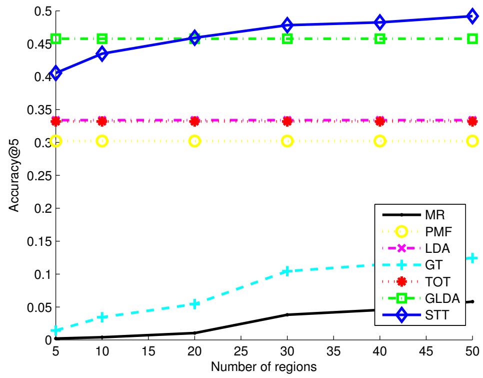


(b) Gowalla data set.

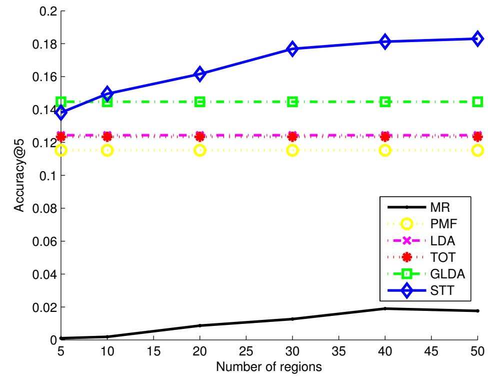


(c) Brightkite data set.

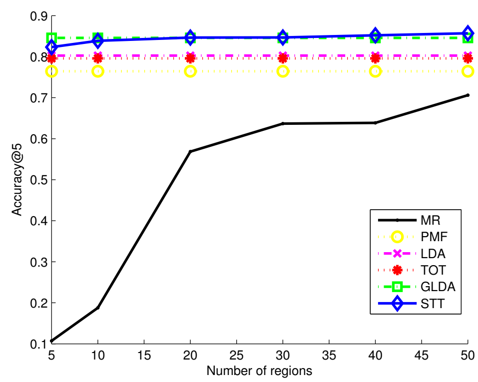
Figure 4.9: Precision@5 of POI recommendation for different number of topics.



(a) Twitter data set.



(b) Gowalla data set.

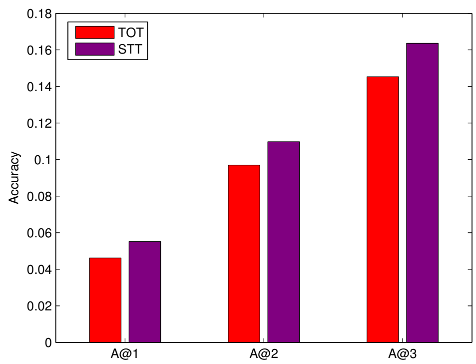


(c) Brightkite data set.

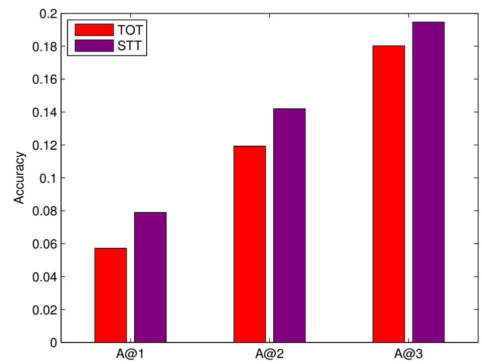
Figure 4.10: Precision@5 of POI recommendation for different number of regions.

Time Recommendation

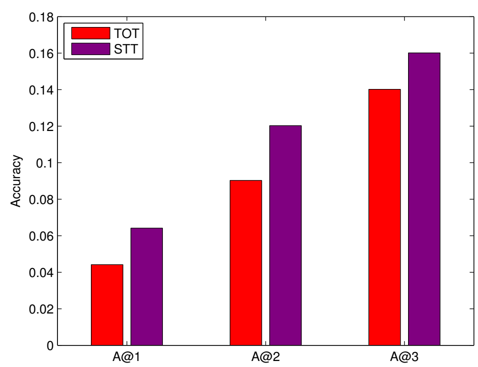
Figure 4.11(a), 4.11(b), and 4.11(c) show the precision@1,2,3 results of time recommendation in the Twitter, Gowalla and Brightkite data sets, respectively. We only compare the TOT and STT models, since all the other models cannot predict (or recommend) time. Both TOT and STT model the correlation of user movements and the temporal aspect of user check-ins. The major difference is that STT models user regions and regularizes the user POIs by their regions. This facilitates not only the improvement of POI recommendation as shown in the last section but also the improvement of time recommendation. STT improves the time recommendation precision@1 of TOT by 19% (Twitter), 38% (Gowalla), and 45% (Brightkite).



(a) Twitter data set.



(b) Gowalla data set.



(c) Brightkite data set.

Figure 4.11: Precision@1,2,3 of time recommendation.

4.4 Conclusion

In this chapter's work, we address the problem of spatio-temporal topic modeling in mobile social media for POI and time recommendation. Previous work has explored recommendation algorithms and geographical and temporal topic models that model either spatial or temporal aspects of user check-ins, but do not consider all of them together. We propose the first spatio-temporal topic model to capture the spatial and temporal aspects of check-ins, as well as user profiles (topic and region distributions) in a single probabilistic model, called Spatio-Temporal Topic (STT) model. STT exploits the interdependencies between users' regions and their POIs, and between temporal activity patterns and POIs. More specifically, STT is based on the intuitions that 1) users have visited multiple regions and their activities are restricted by these regions; 2) temporal activity patterns of POIs affect the decision of user check-ins. We argue that taking the users' region distributions, and the time-specific topic and POI distributions into account enables a more accurate discovery of relevant topics and regions. We present the graphical model of STT and a corresponding method of parameter learning. We perform an experimental evaluation on Twitter, Gowalla, and Brightkite data sets from New York City. We compare STT against state-of-the-art methods in the areas of recommender systems, and geographical and temporal topic modeling. Our experiments demonstrate substantially improved performance in POI and time recommendation.

Chapter 5

Social Topic Model

5.1 Introduction

Based on the well-known effects of social influence and selection, several social network-based methods for item recommendation [47, 33, 48, 69] have been proposed. These methods assume that friends in the social network tend to have more similar behavior patterns than non-friends. Since the problem of POI recommendation in LBSNs can be understood as item recommendation using social networks, several works [71, 72, 13, 20] applied similar social network-based methods for POI recommendation in LBSNs. In particular, for a target user, the existing social network-based approaches [71, 72, 13, 20] search his/her friends in the social network and recommend POIs visited by his/her friends. Their experimental results show that social network-based approaches outperform those approaches without social networks, but the improvement of top-k recommendation accuracy for POIs (~3%) is far smaller than for movies (~20%) [69].

We argue the reason is that the nature of POIs is quite different from the nature of items such as movies. Firstly, to know and rate a POI, a user has to physically visit that POI, and this commitment is more serious than rating a movie online. Even if a user makes the effort to visit the POI, s/he often does not check in due to privacy or safety concerns. Secondly, POIs are specific to a neighborhood or city and are not shared between users from different neighborhoods or cities. Since friends usually live in different neighborhoods or cities, co-check-ins of friends at POIs are less common than movies that friends co-like.

To understand the nature of location-based social networks, let us take an example of a Foursquare data set, which contains 9,672 users, 32,291 POIs, 375,646 check-ins, and 40,470 undirected social relations among users (mutually following relations) from a Twitter social network. We observe that only 20.6% (8,376/40,470) of pairs of friends co-check in at POIs. In other words, approximately 80% of friends do not have any POIs in common, and similar results are reported in [20, 60]. These results verify that the performance improvement of social network-based approaches over those approaches without social networks should be insignificant for POI recommendation. We conduct the same analysis on a Flixster data set used in movie recommendations [33, 69], where its undirected social network is from Facebook. We observe that approximately 40% of friends have watched common movies, which is twice the percentage of friends with common POIs in the Foursquare data set. Social networks of both data sets are acquaintance social networks, which are essentially the same but friends' behaviors are quite different. Therefore, modeling social networks for POI recommendation should be different from that for movie recommendation.

We assume that the effects of social influence and selection exist in social networks for POI recommendation. Unlike in movie recommendation, these effects cannot influence friends' check-in behaviors directly but can influence them indirectly. We hypothesize that friends living in different neighborhoods or cities have similar interests, such as watching movies, and they most likely do not check in the same POI, but (the categories of) their POIs should be semantically similar, i.e., cinema or video stores. In the Foursquare data set, each POI is associated with a set of tags, i.e., "music", "shopping", or "mexican food" etc. To test our hypothesis, we build a topic model using Latent Dirichlet Allocation (LDA) [11] of all POIs assuming that POIs are "documents" and tags are "words". For 100 topics, we observe that a very high percentage (80.3%) of friends in the social network shares topics, which is four times as high as the percentage of friends who share POIs.

Motivated by the above observations, we propose a novel social topic model for POI recommendation in social networks. The intuition of our proposed model is that friends tend to check in POIs with similar semantic meanings. Basically, a check-in is represented by a user and a POI with a bag of words (e.g., from user-generated text such as tags, posts or reviews), which are considered as observed random variables. Similar to LDA, a set of latent topics is defined. Each user is associated with a probability distribution over topics, which captures the user interests, and each topic has a probability distribution over POIs and words, which captures the semantic meaning of POIs. Topics are assumed to represent sets of POIs that have similar meaning/function such as parks, night clubs, or restaurants. Additionally, we model the social network to regularize the users' topic distribution/interests by those of their friends, and to indirectly influence the users' check in patterns. This modeling strategy is different from the one used in the existing social network-based POI recommendation approaches [71, 72, 13, 20].

We propose an EM (Expectation-Maximization) algorithm to learn the latent random variables and parameters of the social topic model that maximizes the likelihood of observed random variables. We perform experiments on large scale real life data sets from Foursquare and Yelp, and we evaluate the effectiveness of the proposed model and of state-of-the-art models in terms of the accuracy of top-k POI recommendation.

The major contributions of this chapter's work are as follows:

- We comprehensively analyze the nature of POIs and the benefits of a social network for POI recommendation.
- We propose a novel social topic model for POI recommendation using social networks.
- Through comprehensive experiments, we demonstrate that the proposed model consistently improves the average recall@1,2,...,20 significantly for POI recommendation compared to existing state-of-the-art social network-based recommendation algorithms on Foursquare and Yelp data sets. We also demonstrate that the location-based social network is very useful for improving the performance of non social network-based model when it is appropriately modeled.

5.2 Social Topic Model

In this section, we first introduce the problem definition of top-k POI recommendation, and then present our proposed **ST** (**S**ocial **T**opic) model.

5.2.1 Problem Definition

We assume that there are a set of users $U = \{u_1, u_2, ..., u_{|U|}\}$ and a set of POIs $I = \{i_1, i_2, ..., i_{|I|}\}$, and each user performs actions on a set of POIs. The actions performed by users on items are given in a matrix $D = [d_{u,i}]_{|U|\times|I|}$, where $d_{u,i}$ denotes the action of user u on POI i. Normally, each action contains a post or document on Twitter or Yelp. Since we will apply topic models which process on "documents", we must define the concept of "documents". The first option is to define each user-generated text as a document $d_{u,i}$. We could also define a document d_i and d_u as a set of all user-generated texts of a POI i and a user u, respectively. For the time being, we consider the set of all user-generated texts of a POI as a document d_i, because most of the user-generated texts such as tweets from Twitter or reviews from Yelp tend to discuss the properties of the POI. In other words, each document d_i contains a set of user-generated texts describing the POI i, and each check-in $d_{u,i}$ of user u on POI i contains a document of POI i.

In addition, there is a social network among users, which is represented as an undirected graph $G = (U, E)$, where an undirected edge $(u, v) \in E, u \in U, v \in U$ from user u to user v represents the fact that u and v are friends.

POI recommender systems can recommend a set of POIs that users may be interested in, based on the history of user check-ins. The problem of top-k POI recommendation is defined in Chapter 2.

5.2.2 Assumptions

Latent factor models [12, 59, 58, 38, 39, 40] are widely used in recommender systems to model the interactions between users and items. Based on the nature of location-based social networks that we mentioned in the introduction section, we make the following assumptions.

- Assumption 1. Most friends cannot influence a user's decisions to check in at POIs.

This is a counter-intuitive assumption, because social influence and correlation are exploited in many social network-based recommender systems. However, in the context of POI recommendation, we cannot directly adopt the modeling of social influence and correlation from traditional item recommendation. For example, unlike traditional movie recommendation in which all the interactions between users and movies only take place in a virtual network, check-ins at POIs require real-world physical commitments, which means users will check in at POIs in person. Intuitively, users are likely to check in at nearby POIs, and friends living in different neighborhoods check in at different POIs.

- Assumption 2. Friends tend to share similar interests.

This is a common assumption, which is motivated by the theory of "homophily", which states that users connect to similar users, and by the theory of "social influence" which claims that connected users become more similar to each other. These phenomenons of homophily and social influence interact with each other, and their collective effect is referred to as "social correlation". Together with assumption 1, we argue that the social network can be useful when it is modeled properly.

- Assumption 3. Users' interests play a major role in users' decisions to check in at POIs.

This is also a common assumption. However, we list it here because in our context, this phenomenon is even stronger. We assume that users' interests can be captured by analyzing user-generated texts, which are associated with POIs, and that user interests greatly influence user check-ins. For example, in Yelp, users who like sushi will check in at sushi restaurants frequently.

All the above assumptions will be tested and verified in the experiments.

5.2.3 Model

In this section we describe our proposed **ST** (**S**ocial **T**opic) model. We first introduce the notations of the ST model which are listed in Table 6.4. Our input data, i.e., POIs and user-generated texts are modeled as observed random variables, shown as shaded circles in Figure 5.1(c), and we use $i_{u,d}$ and $w_{u,d,n}$ to denote them. The topic (index) of documents is considered as a latent random variable, which is denoted as $z_{u,d}$. Users are associated with topic distributions, i.e., θ, from which the topics of documents are sampled. Topics are associated with both POI distributions ψ and word distributions ϕ. Given the sampled topic, POIs are drawn from the POI distribution and words are drawn from the word distribution of that topic.

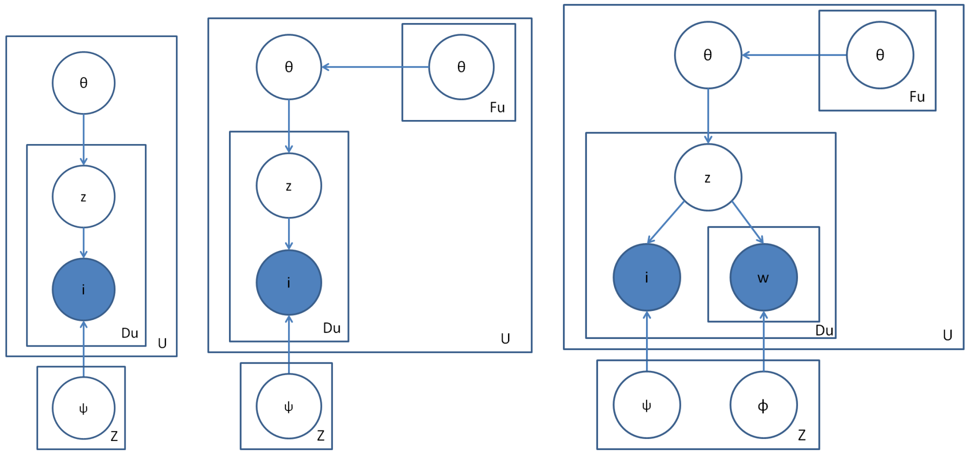
A standard latent factor model LDA [11] can be applied for modeling user check-ins. Figure 5.1(a) shows the graphical model of LDA. Specifically, LDA assumes that there is a latent topic distribution θ_u for user u, and a latent POI distribution ψ_z for topic z. The dependency between user preferences and their POIs is transferred through the topic. LDA captures the correlation between preferences of different users, such that users who have similar preferences share the same topics. Note that the topics play a similar role as the latent factors in MF (Matrix Factorization).

Table 5.1: Notations

Variable	Interpretation
$i_{u,d}$	POI index of the d^{th} document posted by the u^{th} user
$w_{u,d,n}$	n^{th} word of the d^{th} document posted by the u^{th} user
$z_{u,d}$	topic assignment of the d^{th} document posted by the u^{th} user
Z	set of topics
U	set of users
I	set of POIs
D_u	set of documents of the u^{th} user
$N_{u,d}$	set of words in the d^{th} document of the u^{th} user
V	set of the vocabulary
θ^0	topic distribution of the background
θ_u	topic distribution of the u^{th} user
ψ^0	POI distribution of the background
ψ_z	POI distribution of the z^{th} topic
ϕ^0	word distribution of the background
ϕ_z	word distribution of the z^{th} topic
F_u	set of friends of the u^{th} user

Equivalent to [13, 48], SLDA (Social LDA) extends LDA as shown in Figure 5.1(b), where F_u represents the set of friends of user u. SLDA models both user check-ins and the social network, and the intuition is that a user's topic distribution is regularized by the topic distribution of his/her friends.

Different from the existing latent factor models such as LDA and MF, we propose a novel ST (Social Topic) model as shown in Figure 5.1(c), which associates a word distribution with a topic so that it can describe the latent user and POI factors. Intuitively, collaborative filtering assumes that POIs A and B should both have high probabilities in POI distributions of some topic(s) if many users frequently co-occur in both A and B. POIs A and B do not necessarily have the same functionality. However, ST further assumes that POIs with high probabilities for the same topic should be cohesive in their functions, e.g., a topic with high probabilities for words like "coffee", "Java", and "latte" should have high probability only for coffee shops. This design enables ST to detect users with similar interests and POIs with similar functions, and enables ST to better deal with "cold start" users, i.e., users who have very few check-ins, since the words of their few check-ins are more informative than their POIs. This model design is based on assumption 3 that users' interests influence their check-ins.



(a) LDA. (b) Social LDA. (c) Social Topic.

Figure 5.1: The graphical models. The background parameters θ^0, ψ^0, ϕ^0 are omitted.

As formulated in assumption 1, we observe that POIs are hardly shared by friends who live in different neighborhoods or cities due to the nature of location-based social networks. It means that the SLDA model does not capture the nature of location-based social networks, and theoretically cannot boost the recommendation accuracy over LDA by much. Therefore, in our proposed ST model an important difference compared to existing social network-based models such as SLDA is that topics are associated not only with POIs but also with user-generated texts, which implies that the user topic distribution θ takes both the user check-in behavior and user interest into account. The topics z of documents are sampled from the users topic distributions θ. POIs i and words w are sampled from the POI distribution and the word distribution, respectively, of the sampled topic z. Based on assumption 2, we keep the social regularization part as in SLDA, and the social network should be boosting the recommendation accuracy as it influences both the user check-in behavior and user interests directly.

Next, we describe the generative process of the ST model for a single check-in $d_{u,i}$ of a given user u:

- Compute a topic distribution of user u
  - $\theta_u \sim p(\theta_u | F_u, \theta)$
- Draw a topic index $z_{u,d}$
  - $z_{u,d} \sim p(z_{u,d} | u, \theta^0, \theta)$

- Draw a POI index $i_{u,d}$, given the topic index $z_{u,d}$
  - $i_{u,d} \sim p(i_{u,d}|z_{u,d}, \psi^0, \psi)$
- Draw each word in d given the topic index $z_{u,d}$
  - $w_{u,d,n} \sim p(w_{u,d,n}|z_{u,d}, \phi^0, \phi)$

The ST model first computes a topic distribution of user u by averaging the topic distributions from u's friends f_u.

$$p(\theta_u|F_u, \theta) = \frac{\sum_{v \in F_u} \theta_v}{|F_u|} \tag{5.1}$$

where F_u represents the set of friends of user u, and $|F_u|$ denotes the number of friends.

For each check-in by user u, the ST model then samples a topic from the set of topics. To generate a topic z, the model uses multinomial distributions of the background and user's topic together as follows:

$$p(z|u, \theta^0, \theta) = \frac{\exp(\theta_z^0 + \theta_{u,z})}{\sum_{zz=1}^{|Z|} \exp(\theta_{zz}^0 + \theta_{u,zz})} \tag{5.2}$$

where θ^0 is the background topic distribution, and θ is the topic distributions of user u. To simplify the notations, we use $p(z|u, \theta^0, \theta) = \alpha_{u,z}$. This approach employs the sparse coding technique introduced in the SAGE (Sparse Additive Generative) model [19]. The major advantage of SAGE is that it does not require additional latent "switching" variables when the model needs to take multiple factors into account. For example, in order to model a mixture topic distribution of background and user factors, based on the background topic distribution θ^0, it models the difference in log-frequencies, e.g., θ from the background topic distribution θ^0, instead of the log-frequencies themselves.

Each POI index i is drawn depending on the sampled topic z as follows:

$$p(i|z, \psi^0, \psi) = \frac{\exp(\psi_i^0 + \psi_{z,i})}{\sum_{ii=1}^{|I|} \exp(\psi_{ii}^0 + \psi_{z,ii})} \tag{5.3}$$

where ψ^0 is the global distribution of POIs and ψ_z is the POI distribution of topic z. To simplify the notations, we use $p(i|z, \psi^0, \psi) = \beta_{z,i}$.

Similarly, for generating the word index, we use a multinomial distribution as follows:

$$p(w|z, \phi^0, \phi) = \frac{\exp(\phi_w^0 + \phi_{z,w})}{\sum_{ww=1}^{|V|} \exp(\phi_{ww}^0 + \phi_{z,ww})} \tag{5.4}$$

where ϕ^0 is the global distribution of words and ϕ_z is the word distribution of topic z. To simplify the notations, we use $p(w|z, \phi^0, \phi) = \delta_{z,w}$.

5.2.4 Parameter Learning

Our goal is to learn parameters that maximize the marginal log-likelihood of the observed random variables $\mathbf{i}, \mathbf{w}$. The marginalization is performed with respect to the latent random variable $\mathbf{z}$, and it is hard to be maximized directly. Therefore, we apply the MCEM (Monte Carlo Expectation Maximization) algorithm to maximize the complete data likelihood $p(\mathbf{z}, \mathbf{w}, \mathbf{i}|\mathbf{u}, \Theta)$ in Equation 5.5, where $\Theta = \{\theta^0, \theta, \psi^0, \psi, \phi^0, \phi\}$.

$$
\begin{aligned}
&p(\mathbf{z}, \mathbf{w}, \mathbf{i}|\mathbf{u}, \Theta) \\
&\qquad = p(\mathbf{z}|\mathbf{u}, \theta^0, \theta) \times p(\mathbf{w}|\mathbf{z}, \phi^0, \phi) \times p(\mathbf{i}|\mathbf{z}, \psi^0, \psi) \\
&\qquad = \textstyle\prod_{u=1}^{|U|} \prod_{d=1}^{|D_u|} \alpha_{u,z_{u,d}} \times \prod_{u=1}^{|U|} \prod_{d=1}^{|D_u|} \beta_{z_{u,d},i_{u,d}} \\
&\qquad \times \textstyle\prod_{u=1}^{|U|} \prod_{d=1}^{|D_u|} \prod_{n=1}^{|N_{u,d}|} \delta_{z_{u,d},w_{u,d,n}}
\end{aligned}
\tag{5.5}
$$

where $\alpha_{u,z}, \beta_{z,i}, \delta_{z,w}$ are shown in Equation 5.2, 5.3, and 5.4.

According to the MCEM method, we sample the latent variables $\mathbf{z}$ in the E step and maximize the parameters Θ in the M step. To sample a single variable $z_{u,d}$ given all other variables fixed, we use Equation 5.6.

$$
p(z_{u,d}|\mathbf{z}_{-u,d}, \mathbf{i}, \Theta) \propto \alpha_{u,z_{u,d}} \times \beta_{z_{u,d},i_{u,d}} \times \prod_{n=1}^{|N_{u,d}|} \delta_{z_{u,d},w_{u,d,n}} \tag{5.6}
$$

In the M step, we maximize the log likelihood of Equation 5.5 with respect to the parameters Θ with the fixed latent variable $\mathbf{z}$ that is sampled in the E step. To update the parameters, we use the gradient descent learning algorithm PSSG (Projected Scaled Sub-Gradient) [61], which is designed to solve optimization problems with L1 regularization on the parameters. More importantly, PSSG is scalable because it uses the quasi-Newton strategy with line search that is robust to common functions.

According to the limited-memory BFGS [46] updates for the quasi-Newton method, we get the derivative functions of the parameters $\theta^0, \theta, \psi^0, \psi, \phi^0, \phi$ in the following equations in Figure 5.2, where $|d(u, z)|$ represents the number of check-ins assigned to topic z by user u, and $|d(z, i)|$ represents the number of check-ins assigned to topic z of POI i, $|d(z, w)|$ represents the number of words assigned to topic z, $|d(u)|$ represents the number of check-ins of user u, and $|d(z)|$ represents the number of words assigned to topic z. In Equation 5.8, the model regularizes θ_u by friends θ_{F_u}, and η is the learning rate.

Good initializations of parameters can speed up the learning process towards the convergence of the objective function by reducing the number of iterations. We set the background

$$\frac{\partial L}{\partial \theta_z^0} = \sum_{u=1}^{|U|} |d(u,z)| - \sum_{u=1}^{|U|} \Big(|d(u)| \times \alpha_{u,z} \Big) \tag{5.7}$$

$$\frac{\partial L}{\partial \theta_{u,z}} = |d(u,z)| - |d(u)| \times \alpha_{u,z} - \eta \times \sum_{v}^{F_u} (\theta_{u,z} - \theta_{v,z}) \tag{5.8}$$

$$\frac{\partial L}{\partial \psi_i^0} = \sum_{z=1}^{|Z|} |d(z,i)| - \sum_{z=1}^{|Z|} \Big(|d(z)| \times \beta_{z,i} \Big) \tag{5.9}$$

$$\frac{\partial L}{\partial \psi_{z,i}} = |d(z,i)| - |d(z)| \times \beta_{z,i} \tag{5.10}$$

$$\frac{\partial L}{\partial \phi_w^0} = \sum_{z=1}^{|Z|} |d(z,w)| - \sum_{z=1}^{|Z|} \Big(|d(z)| \times \delta_{z,w} \Big) \tag{5.11}$$

$$\frac{\partial L}{\partial \phi_{z,w}} = |d(z,w)| - |d(z)| \times \delta_{z,w} \tag{5.12}$$

Figure 5.2: The derivative equations for parameters θ^0, θ, ψ^0, ψ, ϕ^0, and ϕ in the ST model.

POI distribution ψ_i^0 to the log-frequency of check-ins of the POI i, and the background word distribution ϕ_w^0 to the log-frequency of the word w.

5.2.5 Top-k POI Recommendation

The ST model can be employed for top-k POI recommendation as follows. Given a user, our task is to recommend top-k new POIs, that user will visit in the future. More precisely, given the user u and all POIs with user-generated texts, the probability that user u visits POI i is computed in Equation 5.13:

$$\begin{aligned} p(i, \mathbf{w}|u, \Theta) &\propto \sum_{z}^{Z} p(i, z, \mathbf{w}|u, \Theta) \\ &= \textstyle\sum_z^Z p(z|\theta^0, \theta^{user}) \times p(i|z, \psi^0, \psi) \\ &\quad \times \textstyle\prod_n p(w_n|z, \phi^0, \phi) \end{aligned} \tag{5.13}$$

5.3 Experiments

In this section, we experimentally evaluate the effectiveness of our proposed ST (Social Topic) model. We are in particular interested in 1) how the proposed ST model performs in comparison with state-of-the-art social network-based POI recommenders; and 2) how the social network contributes in the ST model. We report the experimental results on Foursquare and Yelp data sets, and we evaluate the comparison partners using the recall@k for top-k POI recommendation.

5.3.1 Experimental Setup

Data Sets

In this chapter's work, we take two location-based social networking websites Foursquare and Yelp, which contain users, POIs, and a social network, as our case studies.

Foursquare is one of the most popular location-based social networking websites, where users "check in" at venues. We crawled a data set from Foursquare using its API[1], and we collect public Foursquare check-in data (tweets) from Sept. 2010 to Jan. 2011 through Twitter with the same crawling strategy as proposed in [20, 60]. We also collect the user friendships from the Twitter follower-followee relationships. Note that this data set is available on one author's homepage[2].

Yelp was launched in 2005, and it has quickly become a popular website providing services for writing reviews on businesses. We use a publicly available data set from the competition of a Yelp data set challenge[3]. It is a deep data set from a US city – Phoenix, and it covers a square region of 50×50 km around the center of Phoenix.

For both data sets, in the pre-processing user-generated texts (e.g., tweets from Foursquare and reviews from Yelp) are processed by tokenizing on whitespace and punctuations, while we remove the URLs starting with "http" and user names starting with "@". Then we remove all texts with non-latin characters, followed by removing stop words, and the words with occurrences less than 100. The statistics about the data sets are presented in Table 6.1. Note that our data sets are larger or comparable to the ones used in the literature [71, 72, 20, 13].

The user-POI check-in matrix is very sparse, i.e., it has a sparsity of 99.99% on both data sets. On average, each user checked in only at a very small fraction of all the POIs, and each POI is checked in only by a very small fraction of all the users. Figure 5.3 and Figure 5.4 show the histograms of number of check-ins in the Foursquare and Yelp data sets

[1] https://developer.foursquare.com

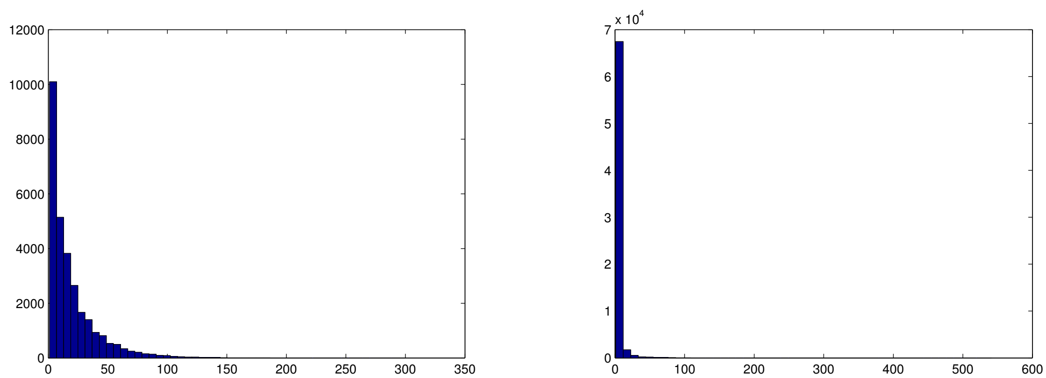
[2] http://www.sfu.ca/˜boh

[3] https://www.yelp.com/dataset_challenge/

Table 5.2: Statistics of data sets from on Foursquare and Yelp.

#	Foursquare	Yelp
Users	29,117	70,817
POIs	364,259	15,585
Check-ins	785,249	335,022
Friendships	89,693	151,516
Vocabulary	3,417	3,456
Avg. check-ins/user	26.96	4.73
Avg. check-ins/POI	2.15	21.49
Avg. friendships/user	3.08	2.14

by users and POIs, respectively. We observe that they follow the Power-law distribution, which means that a few users/POIs get most of the check-ins while a large number of users/POIs get few check-ins. Users/POIs with few check-ins (less than 10 check-ins) are referred as "cold start" or "long-tail" users/items in recommender systems. We are in particular interested in cold start users/POIs because more than 90% of users/POIs are cold start.



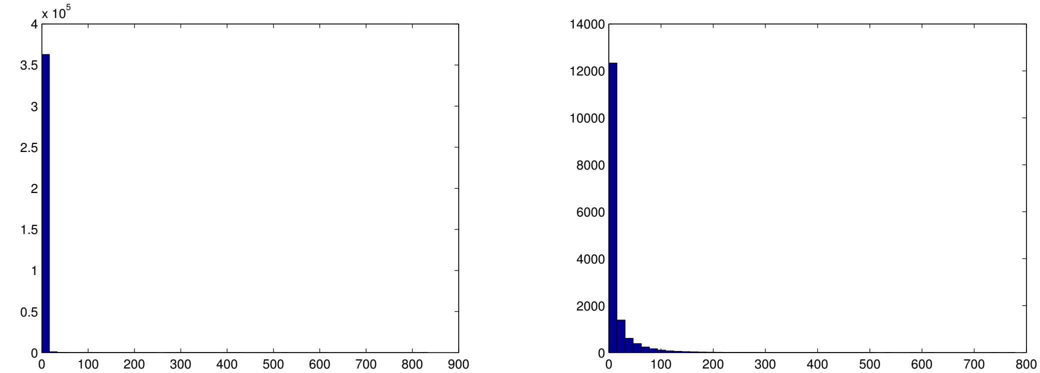
(a) Foursquare.

(b) Yelp.

Figure 5.3: Histograms of number of check-ins of users.

Comparison Partners

In our experiments, we evaluate and compare our proposed ST model with the following comparison partners.



(a) Foursquare. (b) Yelp.

Figure 5.4: Histograms of number of check-ins of POIs.

- Popularity (POP). This is a naive baseline that ranks POIs according to their popularity, i.e., the number of check-ins in the given training set. The more check-ins the POI has, the higher its position in the recommendation list. Note that it is a non-personalized recommendation approach: for any given user, the recommendations are always the same.

- Probabilistic Matrix Factorization (PMF). This is a well-known matrix factorization model proposed in [58]. The users' and POIs' factors are obtained by factorizing the user-POI check-in matrix, and the recommendation probability of a user on a given POI is multiplied by the given user and POI factors.

- **Latent Dirichlet Allocation** (LDA). This is an LDA model, where the only observed variable is the index of POI $\mathbf{i}$ and the only latent variable is the topic $\mathbf{z}$. Note that this model is equivalent to the original LDA model that generates words instead of indexes of POIs, and LDA is equivalent to PMF according to the model section as we described.

- **Probabilistic Matrix Factorization with Social Regularization** ($PMFSR$). This is an extension of the Probabilistic Matrix Factorization model fused with regularization in the social network as proposed in [13, 48], which achieves the best top-k item recommendation performance among social network-based recommendation algorithms in [69].

- **Social Latent Dirichlet Allocation** (*SLDA*). This is an extension of the LDA model, where social regularization is included, which is described in Section 6.4.3 and is equivalent to the social regularization of MF in [13, 48].

- **Social Topic with Text only** (*STT*). This is a simplified version of the ST model, where we remove the social network part. The observed variables are the index of POI $\mathbf{i}$ and the index of word $\mathbf{w}$, and the latent variable is the topic $\mathbf{z}$.

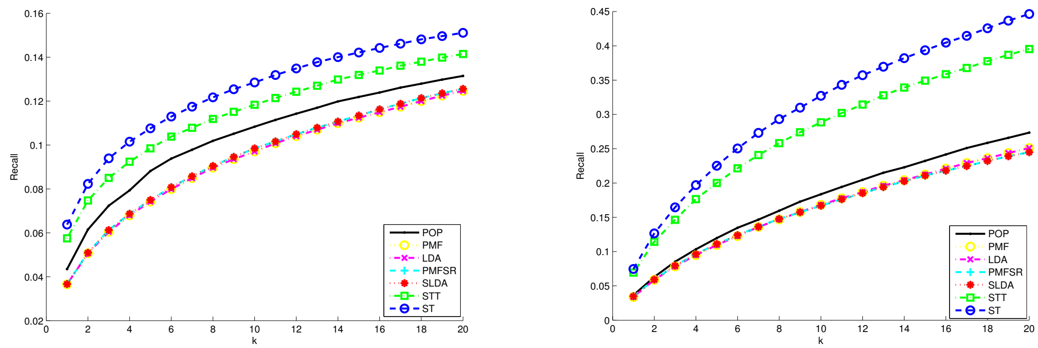
- **Social Topic** (*ST*). This is the full social topic model proposed in this chapter's work.

Note that we do not compare against [33, 47, 49, 72], since they are similar to PMFSR, which achieves the best recommendation accuracy on top-k item recommendation in the experiments of [69].

5.3.2 Experimental Results

For all comparison partners, we present the performance results with well tuned parameters. Figure 5.5(a) and 5.5(b) show the recall@[1,2,...,20] results for POI recommendation for all users using 10 topics/latent factors in the Foursquare and Yelp data sets, respectively. Similarly, Figure 5.6(a) and 5.6(b) show the results using 20 topics/latent factors. Note that the POP results are identical in both figures since POP is independent of the number of topics/latent factors. Larger values of k are usually not that important for the top-k recommendation task. There is not a big difference whether a POI is placed within the top 100 or the top 200, because neither of them will be presented to the user.

We observe that our ST models consistently outperform all other models on both Foursquare and Yelp data sets. PMF and LDA (as well as PMFSR and SLDA) yield similar results in both data sets since they are conceptually equivalent. Both PMFSR and SLDA perform the same or slightly better than PMF and LDA respectively, because their assumption is that friends check in at the same POIs. This assumption is invalid in location-based social networks as discussed before, since a large percentage of friends live in different neighborhoods or cities and usually do not check in at the same POIs. POP performs slightly better than PMF, LDA, PMFSR, and SLDA, which is consistent with the literature on item recommendation [4], and surprisingly has never been studied in the literature on POI recommendation [13, 44, 29, 73]. POP is a tough baseline method, but



(a) Foursquare.

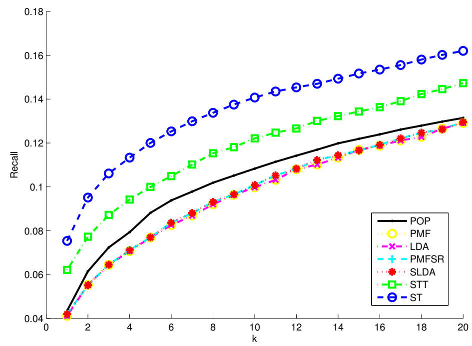
(b) Yelp.

Figure 5.5: Recall@k of comparison partners (# topics = 10) for top-k POI recommendation for all users on Foursquare and Yelp data sets.

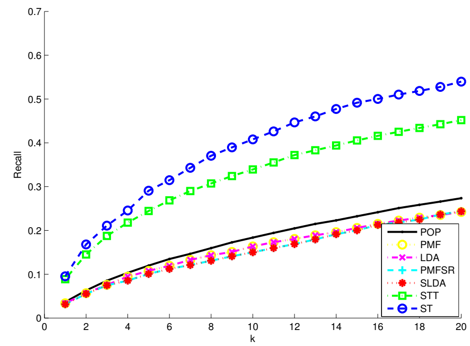
our ST models clearly outperform POP, in particular on the Yelp data set. Compared to the most competitive state-of-the-art method for social network-based item recommendation (PMFSR or SLDA), STT improves the recall@10 on average by approximately 30% (Foursquare) and 100% (Yelp), which indicates that modeling the user interests can help improving the accuracy of top-k POI recommendation significantly. Furthermore, the ST model performs clearly and consistently better than STT, and this indicates that the social network is able to improve the recommendation accuracy if the model captures the nature of location-based social networks as ST does, i.e., the interests of friends are mutually influenced and eventually affect the users check-ins. Finally, we observe that all comparison partners achieve slightly better performance with 20 topics/latent factors than with 10 topics/latent factors. We also tested different numbers of topics/latent factors such as, [30,40,...,100], and the results are similar to the results for 20 topics.

Figure 5.7(a) and 5.7(b) show the recall@[1,2,...,20] results for POI recommendation of cold start users in the Foursquare and Yelp data sets, respectively. The recall@[1,2,...,20] values of all comparison partners for cold start users are lower than the ones for all users in Figure 5.5(a) and 5.5(b), which confirms that cold start users are harder to predict than regular users since there are fewer check-ins in the training data set. However, the relative performance of all comparison partners is consistent for all users and for cold start users.

In real life POI recommender systems, user experience is the ultimate metric for the system performance. For example, if a list of popular POIs is recommended in Yelp, it is



(a) Foursquare.

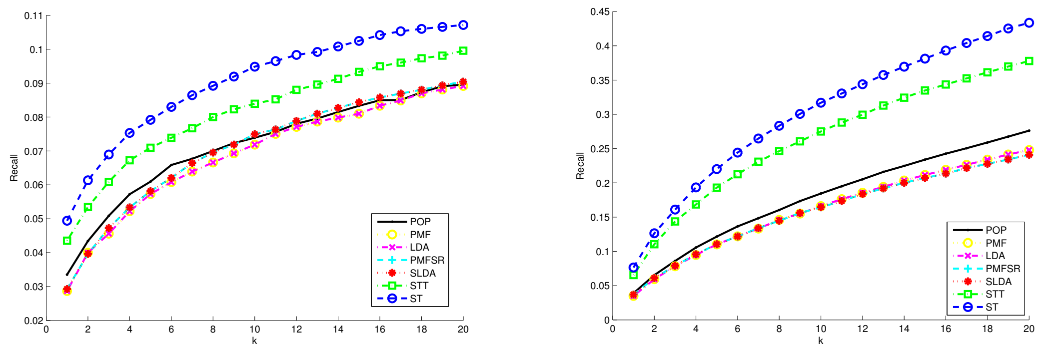


(b) Yelp.

Figure 5.6: Recall@k of comparison partners (# topics = 20) for top-k POI recommendation for all users on Foursquare and Yelp data sets.

quite probable that a user will like (rate highly) the recommended POIs. However, such a recommendation is clearly not very useful because it lacks novelty, since popular POIs are relatively well-known, and a recommender system is probably not even required. In that sense cold start (unpopular) POIs are often more important than non-cold start (popular) POIs. Figure 5.8(a) and 5.8(b) show the recall@[1,2,...,20] results for POI recommendation of cold start POIs to all users in the Foursquare and Yelp data sets, respectively. Not surprisingly, we observe that POP performs worst among all comparison partners since all test POIs are unpopular. The relative performance of all comparison partners is consistent for all POIs and cold start POIs. We notice that the recall gains for the ST models relative to the other comparison partners are significantly larger for cold start POIs than for all POIs, because the ST models exploit user-generated texts which are more informative than POIs alone.

We conclude that all the above experimental results verify our assumptions in Section 5.2.2. The proposed ST model performs consistently and substantially better than all other comparison partners for all users, all POIs, cold start users, and cold start POIs. The ST model is the only comparison partner that outperforms the popularity baseline method for all users and all POIs.



(a) Foursquare.

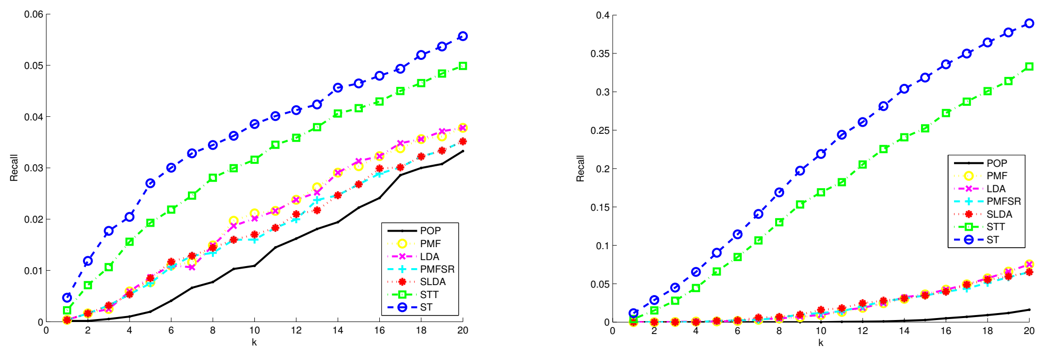
(b) Yelp.

Figure 5.7: Recall@k of comparison partners (# topics = 20) for top-k POI recommendation for cold start users on Foursquare and Yelp data sets.

5.4 Conclusion

In this chapter's work, we address the problem of recommending top-k Point-of-Interest (POI) for users in location-based social networks. State-of-the-art work has explored location-based social networks based on the assumption that friends check in at the same POIs. To the best of our knowledge, we are the first to propose different theories that 1) friends' check-in behaviors are not mutually influenced when they live in different neighborhoods or cities; 2) friends' interests are influenced by each other and affect their check-in behaviors. Based on the above theories, we propose a model to capture the nature of location-based social networks as well as user profiles (topic distributions) in a single probabilistic model, called Social Topic (ST) model. ST exploits the interdependencies between the interests of friends and between user interests and POIs.

We present the ST model and an EM learning algorithm. We perform an experimental evaluation on Foursquare and Yelp data sets, and compare our proposed ST models against the existing state-of-the-art methods. Our experiments demonstrate substantially improved performance in top-k POI recommendation for all users, especially cold start users, and justify the assumptions or theories we proposed.



(a) Foursquare.

(b) Yelp.

Figure 5.8: Recall@k of comparison partners (# topics = 20) for top-k POI recommendation for cold start POIs on Foursquare and Yelp data sets.

Chapter 6

Location Recommendation for New Stores

6.1 Introduction

Determining the optimal location for a new store has been studied in a wide range of research areas (e.g., land economy and urban settlement) for the past few decades. Traditional methods [53, 5] are limited to big retail store chains, due to the fact that data collection and customer surveys are time and money consuming. Moreover, as shown in [35], the accuracy of these predictive models decreases as the size of stores decreases, because smaller store chains have far less collected information from customers, stores, and locations, which greatly compromises the prediction accuracy.

In recent years, with the rapid growth of mobile networks, more and more information useful for determining store locations is becoming available online. Mobile networks enable users to post on social media services, e.g., Yelp, Foursquare and Twitter, from anywhere and anytime. Creating large scale data sets of customers, stores, and locations requires relatively small costs. The scale and finer granularity of these data sets, i.e., a large number of individual user check-ins on stores, has the potential to support more accurate location prediction models for new stores.

The pioneering work of [36] proposes to take advantage of the user check-ins from online location-based social services. A set of locations, i.e., circular areas with 200 meters radius on the map, is generated in advance and considered as potential locations for new stores. Based on the explicit geographic features of locations, such as the number of stores etc, and the mobility features, such as the intensity of user check-ins' inside the location, a regression model is presented to predict the optimal location for new stores of a given store chain, i.e., the location where a store of this chain will attract the highest number of user check-ins.

In spite of its effectiveness on retail store chains, the proposed method has the following limitations. Since the proposed method learns separate regression models for each store chain independently from each other, its success strongly depends on having a sufficient number of stores in each store chain. Therefore, the method is likely to fail for a newly founded and growing store chain with a small number of stores, which is known as the "cold start" problem in the literature [2, 33]. Given the fact that more than 75% of store chains in the Yelp and Foursquare data sets (refer to Section 6.3) have less than 10 stores, this issue is critical and needs to be handled.

We argue that store chains share some implicit patterns of the number of check-ins with other chains. Therefore, we formulate the determination of the optimal location for a new store as a recommendation problem. We propose a latent factor model with different variations to model the implicit patterns of the store chains and the locations that collaboratively govern the interactions between store chains and locations.

In the classic recommendation framework, there is a user-item matrix and each element in the matrix represents the user's rating of that item. To put it in the context of location recommendation for a new store, store chains can be viewed as "users", locations can be viewed as "items", and "user-item ratings" can represent the store specific information, e.g., the number of check-ins at this location for a specific store chain. Different from the simple area definition of locations in [36], we cluster all the stores based on their coordinates and define each cluster as a location. This is much more meaningful when different areas, e.g., downtown and residence areas have greatly varying densities of stores.

Given a store chain (e.g., Starbucks) and a list of locations, our goal is to recommend top-k locations where a new store of the given chain will attract the highest number of check-ins. Without domain knowledge or content analysis, a location recommender system can recommend a set of locations to a store chain by predicting the number of check-ins of stores of the given chain at all locations and recommending top-k locations with the highest number of check-ins. This approach exploits implicit patterns shared by store chains of the same type as well as different types. For example, a small coffee chain can benefit from this setting, because it can learn from other competitors, such as Starbucks. It is also beneficial for Starbucks that it resorts to other types of store chains, such as Walmart, since their popularity may be correlated. Moreover, we include the observed features of locations to overcome the "cold start" problem on locations, i.e., locations with few stores. For example, locations have explicit geographical features, such as the number of stores of the location.

Our experiments on two real life data sets from Yelp and Foursquare demonstrate the ability of our proposed models to accurately recommend locations for different types and sizes of store chains. We also show the robustness of our model to both sparsely (Phoenix) and densely (Manhattan) populated areas. The major contributions of this chapter's work are as follows:

- We formulate the problem of determining the optimal location for a new store as a recommendation problem, and we propose a Latent Factor (LF) model for its solution. This approach enables us to tackle store chains with different types and different sizes.

- We also propose a Feature based Latent Factor (FLF) model, which incorporates available features of locations. In the FLF model, we propose to model the observed location features that strengthens the latent factors of "cold start" locations.

- Through comprehensive experiments, we demonstrate that our proposed models consistently improves the average top-k NDCG@1 (Normalized Discounted Cumulative Gain) of the existing methods by 17% on Yelp and 27% on Foursquare.

6.2 Related Work

In this section, we briefly review related work. There are two lines of related work, which are (a) optimal store placement, which addresses the same problem, and (b) recommender systems, which employs similar methods.

Optimal Store Placement. The problem of optimal store placement has been studied in the research area of land economy for the past decades. Traditional methods [53, 5] are limited to big retail store chains, because data collection and customer surveys are time and money consuming. Moreover, as shown in [35], the accuracy of these predictive models decreases as the size of stores decreases, because smaller store chains have far less collected information from customers, stores, and locations, which greatly compromises the prediction accuracy.

Benefiting from the increasing number of mobile users, a recent work of [36], collects the history of user check-ins from Twitter and Foursquare for three store chains, i.e., Starbucks, McDonald's, and Dunkin Donuts, which are the top 3 biggest retail store chains in the Manhattan area of New York City. A set of locations, i.e., circular areas with 200 meters radius on the map, is generated in advance and considered as potential locations for new stores. Based on the explicit geographic features of locations, such as, location density (the number of stores) and location entropy (the diversity of stores) etc, and the mobility features, such as, area popularity (the total number of check-ins) and transition density (the intensity of user check-ins' transitions inside the location) etc, a regression model is trained, which predicts the number of user check-ins for a new store of a given chain at a given location. The optimal location is determined as the one with the highest predicted number of check-ins. Since the features of successful locations vary for different store chains, separate prediction models need to be built for each store chain. The method of [36] assumes that these models are independent of each other, and builds a set of regression models.

To the best of our knowledge, we are the first to formulate optimal store placement as a recommendation problem.

6.3 Data

In this section, we will first introduce our data sets, and then show some statistic analysis results of the data sets.

6.3.1 Data Collection

Yelp was launched in 2005, and it has quickly become a popular website providing services for writing reviews on stores. We use a publicly available data set from the competition of a Yelp data set challenge[1]. It is a deep data set from a US city – Phoenix, and it covers a square region of 50×50 km around the center of Phoenix. In the Yelp data set, there are totally 43,873 users, 11,537 stores, and 229,907 reviews. Each store is associated with a unique id and a unique pair of latitude and longitude coordinates as well as the store name, type (category), and number of user check-ins.

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

Foursquare is one of the most popular location-based social networking websites, where users "check in" at venues. We crawled a data set from Foursquare using its venue API[2], which is based a snapshot of data collected on August 15^{th} 2013. Our goal was to collect a data set comparable to the one used in [36], which contains a set of all the stores within a square region of 10×10 km around the Manhattan area (the center of New York City). Since the venue search API returns up to 50 venues per query, we first compose a set of grid cells with the size of 50 meters on the map. For each grid cell, we query the API for 50 venues with the coordinates of that cell and a 25 meters radius. After removing unverified and duplicate venues, there are totally 11,627 stores from Foursquare. Based on our estimates, we note that more than 90% of all the publicly available stores from Foursquare are included in this data set.

6.3.2 Statistics

Many factors influence the popularity (number of user check-ins) of a store. In addition to the location factor, the properties of stores, such as, brand reputation, quality, price, and environment etc, are important. However, such information is often unavailable. In order to focus on the location factor, we standardize all other factors by considering only stores from store chains, such as, Starbucks (coffee shop), Panda Express (American Chinese cuisine), and LA Fitness (health club) etc, and compare the number of check-ins of different stores of the same store chain at different locations.

Particularly, a store chain is denoted as a store brand (name) that has at least two stores at different locations in the data sets. We consider two stores are in the same chain if they share the same store name. After manually combining similar store names, like "Starbucks" and "Starbucks coffee shop", there are 3,514 stores from 697 store chains in the Yelp data set, and 3,747 stores from 528 store chains in the Foursquare data set. Some statistics about the data sets are presented in Table 6.1.

We further analyze the types (categories) of store chains. In the Yelp data set, there are multiple category labels on each store. For example, a Starbucks shop is associated with two category labels "Food" and "Coffee & Tea", which correspond to the category and subcategory of the store type. For the sake of calculating the statistics, we only use

[2]https://developer.foursquare.com/overview/venues

Table 6.1: Statistics of the Yelp and Foursquare data sets.

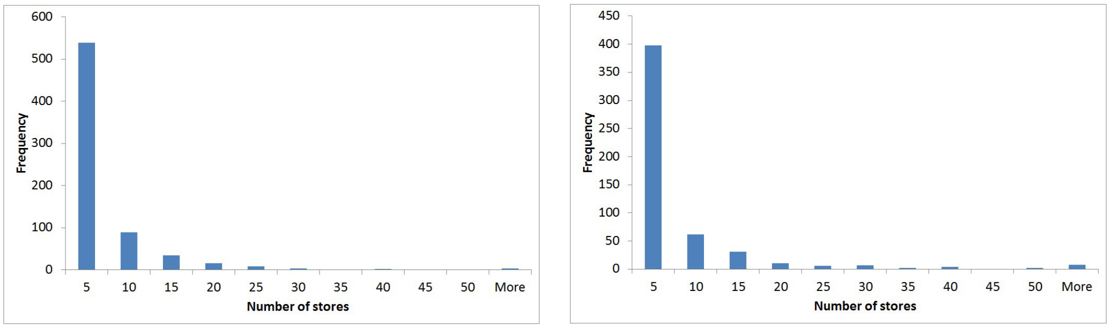
#	Yelp	Foursquare
Stores	11,537	11,627
Stores from chains	3,514	3,747
Store chains	697	528
Average stores/chain	5.04	7.09
Starting date	2005-11-23	NA
Ending date	2013-01-05	2013-08-15

the least commonly used label, which normally corresponds to the subcategory label, i.e., "Coffee & Tea" in this case. We believe that subcategory labels are more meaningful to describe stores than higher level category labels. We process the Foursquare data set in the same way. Table 6.2 presents the top 10 most popular types of store chains from Yelp and Foursquare. We observe that all kinds of restaurants and fast food shops dominate the popular store chains in the Yelp data set, which is not surprising because more than 80% of reviews from Yelp are food related. In addition to restaurants, as a fashion city, New York City, especially the Manhattan area, contains many clothing and cosmetics stores.

Table 6.2: Top 10 most popular types of store chains in the Yelp and Foursquare data sets. "#" represents the number of chains.

Yelp	#	Foursquare	#
Mexican	61	Clothing Store	44
Restaurant	40	Coffee Shop	20
Fast Food	37	Building	19
Pizza	34	Bank	18
American	33	Women's Store	17
Grocery	32	Misc Shop	17
Burgers	28	Cosmetics Shop	17
Department Store	27	Bakery	17
Sandwiches	24	Sandwich Place	17
Italian	23	American	17

Next, we present some statistic results on the size of store chains. Figure 6.1 shows histograms of the number of stores of different store chains in the Yelp and Foursquare data sets. We observe that most store chains from both data sets have few stores, i.e., more than 75% (539 out of 697 on Yelp and 398 out of 531 on Foursquare) store chains have less than 10 stores. Since regression models might fail due to having insufficient training data, the problem of how to improve the prediction accuracy on these store chains becomes significant and challenging.



(a) Phoenix from Yelp.

(b) Manhattan from Foursquare.

Figure 6.1: Histogram of the number of stores of different store chains from Yelp and Foursquare.

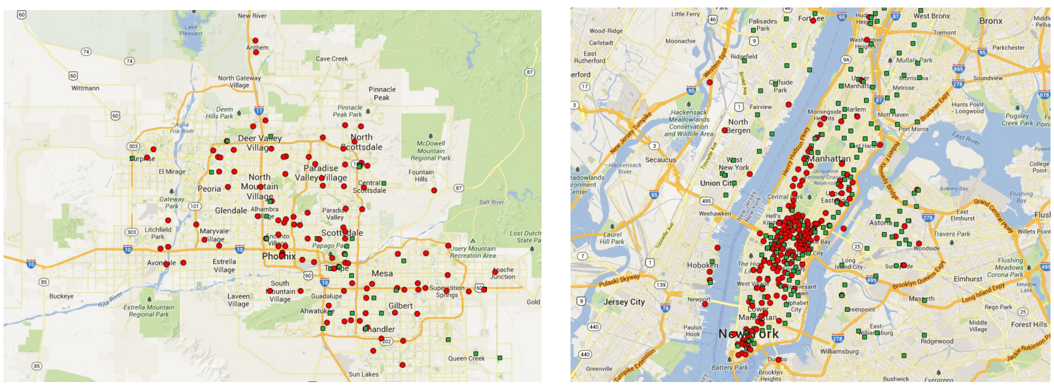
Furthermore, Table 6.3 presents the top 10 biggest store chains from Yelp and Foursquare. We observe that the biggest store chains are Starbucks and Dunkin Donuts, which contain 109 and 191 stores in the Yelp and Foursquare data sets, respectively. Note that some store chains, such as, Starbucks, McDonald's, and Dunkin Donuts, occur in both data sets. In general, the chains from Foursquare contain more stores than the ones from Yelp since the Foursquare (Manhattan) data set covers a more densely populated area and contains more stores for chains.

Table 6.3: Top 10 biggest store chains from Yelp and Foursquare.

Yelp	#	Foursquare	#
Starbucks	109	Dunkin Donuts	191
Subway	65	Starbucks	189
McDonald's	58	Duane Reade	167
Taco Bell	39	Chase Bank	138
Discount Tire	36	Citi Bank	84
Dunkin Donuts	31	McDonald's	84
Walgreens	29	Citi Bike Station	61
Panda Express	29	Rite Aid	56
Warlmart	28	TD Bank	47
Chipotle Mexican Grill	27	7-Eleven	46

6.3.3 Geographical Analysis

To illustrate geographical features of Phoenix and Manhattan in the Yelp and Foursquare data sets, we take an example of stores from Starbucks and Dunkin Donuts, which are popular and big store chains in both data sets. Figure 6.2 shows the geographical distribution of Starbucks and Dunkin Donuts stores in Phoenix and Manhattan. Note that the covered area (Phoenix) in the Yelp data set is approximately 25 times the size of the one (Manhattan) in the Foursquare data set. We can conclude that stores in Foursquare are much more crowded than the ones on Yelp because there are similar numbers of stores in both data sets. This is supported by the observation in Figure 6.2 that the average distance between Starbucks stores in Manhattan is far shorter than in Phoenix.

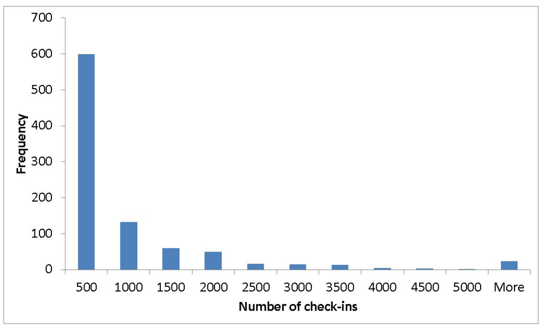


(a) Phoenix from Yelp.   (b) Manhattan from Foursquare.

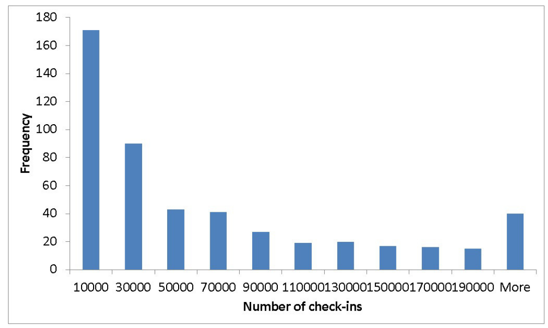
Figure 6.2: The geographical distribution of Starbucks and Dunkin Donuts stores from Yelp and Foursquare. Each red circle represents a Starbucks shop, and each green square represents a Dunkin Donuts shop.

Based on the different geographical features in the Yelp and Foursquare data sets, the definition of locations in [36], i.e., circular areas with a 200 meters radius, is not appropriate, because the locations in Yelp might contain far less stores and cannot convey their "semantic" meaning, i.e., a representative of a community or neighborhood. Therefore, we propose to cluster all the stores in the data sets based on their coordinates, and define each cluster as a location. The average radius of locations (clusters) varies inversely with the varying number of locations (clusters). Particularly, the average radius of locations is 500 and 200 meters in Phoenix and Manhattan when the number of clusters is set to 1000 and 500, respectively. The 200 meter radius in Manhattan from Foursquare is consistent to the one used in [36]. We observe that the areas of locations in the downtown of Phoenix are much smaller than the ones in the residence areas. Figure 6.2 also shows that many store chains (e.g., Starbucks and Dunkin Donuts) co-occur in different locations in both data sets.

Figure 6.3 shows the histogram of the popularity (number of check-ins) of different locations from Yelp and Foursquare. We note that it is consistent in both data sets that most locations attract small numbers of check-ins while a few locations attract large numbers of check-ins.



(a) Phoenix from Yelp.



(b) Manhattan from Foursquare.

Figure 6.3: Histogram of the popularity (number of check-ins) of different locations from Yelp and Foursquare.

6.4 Location Recommendation for New Stores

In this section, we first formulate our research problem of determining the optimal location for a new store in the context of recommendations. After defining the research problem, we introduce some existing regression models addressing the problem, propose a latent factor model that captures both the observed and latent features of store chains and locations, and then propose a feature based latent factor model with regularization to tackle the problem of store chains having few stores. Finally, a stochastic gradient descent algorithm is proposed to learn the model parameters.

6.4.1 Problem Definition

We formally define a store and a location as follows:

Definition 3 (Store) *A store is denoted by a set of attributes: store id (unique), name, a pair of latitude and longitude coordinates, type (category), and number of user check-ins.*

Definition 4 (Location) *A location is defined by a cluster that contains a set of stores, which are geographically cohesive to each other.*

After clustering all the stores, each store is assigned to a specific location. In particular, an example of a publicly available store is as follows:

- "MkbsldEHTRUwDed–bMHIg" (store id), "33.507, -112.275" (coordinates), "1" (location id), "Starbucks" (name), "Food, Coffee & Tea" (type), "1088" (# of check-ins).

which states that a Starbucks shop at the location "1" has attracted 1088 user check-ins.

Moreover, we define a store chain as follows:

Definition 5 (Store chain) *A store chain is defined as a set of stores (at least two stores), which have the same store name.*

Note that stores with a unique name is not considered as a store chain. Moreover, we assume that there is only one store of the given store chain at a specific location. Occasionally, big store chains, e.g., Starbucks, open two and more stores at the same location. In this case, we consider the stores of the given store chain at the same location as one store, and the corresponding number of check-ins is averaged.

We first introduce the notations needed in our problem definition. We assume that all the store chains are from a fixed set S with size $|S|$ and all the locations are from a fixed set L with size $|L|$. Note that we use capital letters to represent the sets and the $|.|$ sign to represent the cardinality (size) of the sets. We denote the number of user check-ins by store chain s at the location l as c_{sl}. All the notations are listed in Table 6.4.

Based on the above definitions and notations, we formalize our research problem as follows:

Problem 6 *Location Recommendation for a New Store (LRNS)*. *Given a store chain s, and a list of candidate locations, our task is to predict the number of check-ins c_{sl} for a new store of chain s at one of these locations l. In other words, the goal is to recommend top-k locations where a new store of the given chain s will attract the highest number of check-ins.*

Table 6.4: Notations of parameters

Variable	Interpretation
s	store chain index
l	location index
c_{sl}	number of check-ins of the s^{th} store chain at the l^{th} location
$\hat{c}_{sl}$	estimate of c_{sl}
S	set of store chains
L	set of locations
C	set of numbers of check-ins given store chains and locations
$\boldsymbol{f}_s$	observed feature vector of the s^{th} store chain
$\boldsymbol{f}_l$	observed feature vector of the l^{th} location
b_s	bias factor of the s^{th} store chain
b_l	bias factor of the l^{th} location
$\boldsymbol{\phi}_s$	latent factor vector of the s^{th} store chain
$\boldsymbol{\phi}_l$	latent factor vector of the l^{th} location
$\boldsymbol{W}$	weight coefficient of observed features for the locations

6.4.2 Regression Model

In this subsection, we discuss a simple regression model to tackle the problem of LRNS introduced in [36]. For a given store chain s, we have a set of numerical responses, i.e., the number of check-ins $C_s = \{c_{sl} | l \in L_s\}$ of the stores of the given chain s at different locations L_s, where L_s is the set of locations where the store chain s already has a store. Since we are recommending a new location to a store chain, the key factors affecting the number of check-ins of each store are the features of the store location.

We use a vector $\boldsymbol{f}_l$ to represent the observed features of location l. Thus, for each chain s, the method of [36] treats the problem of LRNS of the given chain s as a regression problem, which trains a model based on the features $\boldsymbol{f}_l$ of the location l and responses c_{sl} in the training data set, and predicts the number of check-ins c_{sl} of the store chain s at a new location l based on the regression model and the features of the new location. Some representative features proposed by [36] are: 1) density, i.e., the number of stores of the location; 2) neighbor entropy, i.e., an entropy measurement for the frequency of store types of the location; 3) popularity, i.e., the total number of user check-ins from all the stores of the location. Other features include "competitiveness", "quality by Jensen", "transition density", "incoming flow", and "transition quality" (please refer to [36]). All these location features will be used later in our proposed models.

A Linear Regression (LR) model that can be applied to the problem of LRNS for the store chain s is as follows:

$$\hat{c}_{sl} = \alpha_s^T \boldsymbol{f}_l \tag{6.1}$$

Note that α_s is a chain specific coefficient vector that corresponds to the feature vector f_l and will be learned from the training data set.

6.4.3 Latent Factor Model

Although these regression models are simple and easy to be applied to the problem of LRNS, the assumption behind them, i.e., the independence of feature coefficients on store chains, lacks modeling latent factors of locations that can be learned from the interactions between store chains and locations, and compromises the prediction accuracy as shown in the experiment section.

Latent factor models are widely used in recommender systems [22, 9, 38, 39, 40]. A latent factor model can model the interactions between different types of entities, such as "user-item" in recommendation problems, to discover their latent factors and relationships. In this subsection, we propose a latent factor model to address our problem of LRNS. Specifically, we assume that there is a set of latent factor vectors $\boldsymbol{\phi}_s$ for store chain s and $\boldsymbol{\phi}_l$ for location l, and the length of latent factor vectors is a model parameter with typical values ranging from 10 to 50. We also assume that there is a bias factor, i.e., b_s and b_l for the store chain s and the location l, respectively, which shows the different levels of the popularity of store chains and locations. Formally, latent factor representations are defined as follows:

- Each store chain s is mapped into a latent factor vector $\boldsymbol{\phi}_s$ from a $|K|$-D real number space $\mathbb{R}^{|K|}$.
- Each location l is mapped into a latent factor vector $\boldsymbol{\phi}_l$ from a $|K|$-D real number space $\mathbb{R}^{|K|}$.
- Each store chain s is mapped into a latent bias factor b_s from a 1-D real number space $\mathbb{R}^1$.
- Each location l is mapped into a latent bias factor b_l from a 1-D real number space $\mathbb{R}^1$.

Based on the above definitions, we propose our first Latent Factor (LF) model as follows:

$$\hat{c}_{sl} = b_s + b_l + \boldsymbol{\phi}_s^T \boldsymbol{\phi}_l \tag{6.2}$$

where $\hat{c}_{sl}$ denotes the estimated number of check-ins.

Equation 6.2 presents the exact form of the traditional matrix factorization [40] for collaborative filtering, where the response of the number of user check-ins c_{sl} depends on the inner product between the respective latent factor vectors $\boldsymbol{\phi}_s$ for the s^{th} store chain and $\boldsymbol{\phi}_l$ from the l^{th} location, and the bias factors b_s for the s^{th} store chain and b_l for the l^{th} location.

Different from the regression approach, our proposed LF model does not establish separate models for each store chain, and collaboratively learns the parameters from all store chains. The intuition of LF is that if two store chains are similar in the training data set, i.e., numbers of user check-ins are correlated at some locations, they act in a correlated manner in the testing data set.

We define the objective function of the model as follows:

$$\mathcal{O} = \sum_{s}^{|S|} \sum_{l}^{|L|} \mathcal{L}(c_{sl} - \hat{c}_{sl}) + \sum_{s}^{|S|} \sum_{l}^{|L|} \mathcal{R}(b_s, b_l, \boldsymbol{\phi}_s, \boldsymbol{\phi}_l) \tag{6.3}$$

where $\mathcal{L}(.)$ represents the loss function and we use the least square loss function in our experiments. Note that we do not use a ranking loss function because the differences between the corresponding results are trivial according to the work of [68]. The $\mathcal{R}(.)$ function represents the regularization function of the bias and latent factors, and we normally use the L^2-norm.

6.4.4 Feature based Latent Factor Model

The proposed latent factor model does not incorporate observed location features used in the regression models, which have proven helpful for predicting the response of the number of check-ins. Therefore, we introduce a Feature based Latent Factor (FLF) model similar to [17] to capture not only the latent factors of store chains and locations but also their observed features. The basic idea is that the response of the number of check-ins depends on both observed and latent features (factors) of store chains and locations. We define it formally in the following equation:

$$\begin{aligned}\hat{c}_{sl} &= b_s + b_l + \boldsymbol{\phi}_s^T(\boldsymbol{\phi}_l + \boldsymbol{W}\boldsymbol{f}_l) \\ &= b_s + b_l + \boldsymbol{\phi}_s^T\boldsymbol{\phi}_l + \boldsymbol{\phi}_s^T\boldsymbol{W}\boldsymbol{f}_l\end{aligned} \tag{6.4}$$

where $\boldsymbol{W}$ is a weight coefficient matrix of the size of $|K| \times |F|$ for locations, and $|K|$ is the length of latent factors and $|F|$ is the number of observed location features. The purpose of these coefficients is to transform and re-weight the observed location features from the observed feature space to the latent feature space. A new objective function is obtained when we plug Equation 6.4 into Equation 6.3 as follows:

$$\begin{aligned}\mathcal{O} = &\sum_s^{|S|}\sum_l^{|L|}\mathcal{L}(c_{sl} - b_s - b_l - \boldsymbol{\phi}_s^T\boldsymbol{\phi}_l - \boldsymbol{\phi}_s^T\boldsymbol{W}\boldsymbol{f}_l) \\ &+ \sum_s^{|S|}\mathcal{R}(b_s) + \sum_l^{|L|}\mathcal{R}(b_l) + \sum_s^{|S|}\mathcal{R}(\boldsymbol{\phi}_s) + \sum_l^{|L|}\mathcal{R}(\boldsymbol{\phi}_l)\end{aligned} \tag{6.5}$$

The major difference between FLF and LF is the fourth term in the second row of Equation 6.4, i.e., $\boldsymbol{\phi}_s^T(\boldsymbol{W}\boldsymbol{f}_l)$, which basically models the interactions between observed features of locations and latent factors of store chains. The advantage of FLF is that "cold start" locations can resort to their observed features when they cannot establish reliable latent factors.

Compared to the separate linear regression model in Equation 6.1, the part of FLF, i.e., $\boldsymbol{\phi}_s^T(\boldsymbol{W}\boldsymbol{f}_l)$, is similar to a model that we establish a separate regression model for each chain, and then add their objective functions up and learn the parameters together. The major difference is that FLF offers to convert the observed location features into a latent factor space using $(\boldsymbol{W}\boldsymbol{f}_l)$, which is shared with all store chains. Conceptually, knowledge can be transferred from one chain to the other through this feature space conversion. The idea is that "cold start" store chains can "borrow" the latent factors from other store chains.

6.4.5 Parameter Learning

Our goal is to learn the parameters $\Theta = \{b_s, b_l, \boldsymbol{\phi}_s, \boldsymbol{\phi}_l, \boldsymbol{W}\}$ that minimizes the objective function $\mathcal{O}(\Theta, C)$. We use a stochastic gradient descent algorithm to learn the parameters at t^{th} iteration as follows:

$$\Theta^t = \Theta^{t-1} - \tau * \frac{\partial\mathcal{O}}{\partial\Theta} \tag{6.6}$$

where the values of parameters at current iteration Θ^t is based on the values of parameters at the last iteration Θ^{t-1}, and the partial derivative of the objective function $\frac{\partial \mathcal{O}}{\partial \Theta}$ with respect to the specific parameter multiplied by the learning rate τ.

We present the partial derivative functions of the parameters b_i, v_i and w in Equation 6.5 as follows:

$$\frac{\partial \mathcal{O}}{\partial b_s} = -(c_{sl} - \hat{c}_{sl}) + \lambda b_s \tag{6.7}$$

$$\frac{\partial \mathcal{O}}{\partial \boldsymbol{\phi}_s} = -(c_{sl} - \hat{c}_{sl})(\boldsymbol{\phi}_l + \boldsymbol{W}\boldsymbol{f}_l) + \lambda \boldsymbol{\phi}_s \tag{6.8}$$

$$\frac{\partial \mathcal{O}}{\partial \boldsymbol{W}} = -(c_{sl} - \hat{c}_{sl})(\boldsymbol{\phi}_s \boldsymbol{f}_l^T) + \lambda \boldsymbol{W} \tag{6.9}$$

Similarly, we get derivative functions for the remaining parameters, which are omitted because of the page limit.

6.5 Experiments

In this section, we experimentally evaluate the effectiveness of our proposed latent factor models. We also compare it against some baseline methods, e.g., a randomization method and an unsupervised method, and an existing state-of-the-art model [36], i.e., linear regression model. We report our experimental results on Yelp and Foursquare data sets, and we evaluate them using the top-k average accuracy for location recommendation.

6.5.1 Experimental Setup

Data Split. In our data set, we randomly select 10 stores for each store chain as the test data, and the remaining as the training data. This train and test split guarantees a sufficient number of test stores, which is critical for the top-k NDCG (Normalized Discounted Cumulative Gain) computation because a too small number of test data misleads to an unreasonably high NDCG. For example, to rank a list of length 2, the top-2 NDCG (ranges from 0 to 1) is 0.7 even if the list is randomly permuted. Another advantage of that split is that we can compare the NDCG results across all the store chains, because the length of their test data is the same. Note that this split strategy is different from the one in [36], because we evaluate all store chains while they consider only three large retail store chains.

We train models in the training data set. For every 10 stores from the store chains in the test set, the ground truth of ranked list of locations is ordered by the number of check-ins on corresponding stores. To make a recommendation, we rank the locations in decreasing order of their predicted number of check-ins. To evaluate the performance of the comparison partners, we compare the ranked list of recommended locations against the ground truth of ranked list of locations.

Evaluation Metrics. NDCG@k (top-k average NDCG) is used to evaluate the models. NDCG@k is by far the most commonly used metric to measure the performance of recommender systems [22]. The top-k NDCG for a test store chain is computed by comparing the predicted ranked list of locations and the ground truth of ranked list of locations.

To compute DCG@k of the recommended list of location for a specific store chain s, we use the following equation:

$$DCG@k(s) = \sum_{n=1}^{k} \frac{2^{rel(l_n)} - 1}{\log_2(n+1)} \tag{6.10}$$

where l_n represents the recommended location index that is ranked at n^{th} position, and $rel(l_n)$ represents the relevance of the location. We use its relative position in the ground truth ranked list as used in [36]:

$$rel(l_n) = \frac{|L| - rank(l_n) + 1}{|L|} \tag{6.11}$$

Note that the relevance value is 1 when the location is ranked first and decreases to 0 when the ranking goes down. The DCG@k is normalized by the iDCG@k (ideal DCG@k) as follows:

$$NDCG@k(s) = \frac{DCG@k(s)}{iDCG@k(s)} \tag{6.12}$$

Finally, the NDCG@k is computed by averaging over all store chains:

$$NDCG@k(S_{test}) = \frac{\sum_{s}^{|S_{test}|} NDCG@k(s)}{|S_{test}|} \tag{6.13}$$

Comparison Partners. In our experiments, we evaluate the following comparison partners:

- *Rand* (Randomization). This is a baseline method that randomly permutes the locations in the test data set.

- *Pop* (Popularity). This is an unsupervised method that ranks the locations by their total number of check-ins in the training data set.

- *LR* (Linear Regression). This is a well-known Linear Regression method proposed in [36], which builds an independent linear regression model for each store chain.

- *LF* (Latent Factor). This is a well-known matrix factorization model, and we adopt it to handle the location recommendation for new stores in this chapter's work.

- *FLF* (Feature based Latent Factor). The FLF model is a featured based latent factor model proposed in this chapter's work, and we use the same location features used in [36].

6.5.2 Experimental Results

Table 6.5 and 6.6 show the NDCG@1,5,10 of the comparison partners in the Yelp and Foursquare data sets, respectively. The number of latent factors is set to 10 for LF and FLF. All the values of parameters, i.e., $\Theta = \{b_s, b_l, \boldsymbol{\phi}_s, \boldsymbol{\phi}_l, \boldsymbol{W}\}$, are initialized by a Gaussian distribution with zero mean and 0.01 variance. All results are obtained by taking the average results from repeating the training and testing 100 times.

Table 6.5: NDCG@1,5,10 on store chains in Phoenix. Best results are in bold.

Models	@1	@5	@10
Rand	0.49	0.61	0.81
Pop	0.66	0.71	0.86
LR	0.58	0.66	0.83
LF	0.64	0.71	0.85
FLF	**0.68**	**0.73**	**0.87**

Table 6.6: NDCG@1,5,10 on store chains in Manhattan. Best results are in bold.

Models	@1	@5	@10
Rand	0.49	0.61	0.81
Pop	0.67	0.75	0.87
LR	0.58	0.68	0.85
LF	0.65	0.71	0.86
FLF	**0.74**	**0.78**	**0.90**

We observe that the results from both data sets are consistent with each other, and that the performance of all models improves as k increases. The performance gain of all models over the baseline method, Rand, decrease with increasing k, because Rand is sufficiently good at a large k so that the room for improvement is limited. We note that the popularity method (Pop) works very well and outperforms Rand by a big margin. Some what surprisingly, the LR model performs significantly worse than Pop. We argue that this is due to the fact that the LR model has insufficient training data for cold start store chains, and the majority of chains is cold start.

On the contrary, our proposed latent factor models address the cold start problem. The experimental results show that LF lifts the performance of LR in NDCG@1 by 10% on Yelp and by 12% on Foursquare, and FLF improves LR in NDCG@1 by 17% on Yelp and by 27% on Foursquare. As expected, the gains in NDCG@5 and NDCG@10 are smaller. These results confirm that modeling the latent factors of store chains and locations can improve the accuracy of location recommendation, and additionally modeling the observed location features can further boost the accuracy.

As pointed out already, Pop is a surprisingly strong baseline method. Our hypothesis is that store chains open their first stores in popular locations so that location recommendation for cold start chains can be made quite accurately using the popularity feature only. It turns out that LF cannot outperform it because of the large number of cold start chains and locations. However, FLF consistently outperforms Pop. This demonstrates the power of the location features, that have been proposed in the linear regression model, in the latent factor model. To conclude, our FLF model consistently outperforms all other models in NDCG@1,5,10 on both data sets.

We note that the accuracy gain of FLF compared to the other models is larger on Foursquare than on Yelp. We believe that the values of the location features in the Foursquare data set are more robust and reliable than the ones in the Yelp data set, because there are far more users on Foursquare than on Yelp and consequently more stores have been verified on Foursquare than reviewed on Yelp.

Table 6.5 and 6.6 report the results averaged over all store chains. To analyze the impact of the size of the store chain (number of stores of that chain), Table 6.7 and 6.8 show the NDCG@1, i.e., finding the optimal location, of the comparison partners for store chains with different sizes in the Yelp and Foursquare data sets. The tables represent the average performance for chains with a size (in the training data) of 1 to 5, 6 to 10, 11 to 15, 16 to 20, and 20 and more, respectively. The number in brackets denotes the number of store chains in that size category. Note that the results for NDCG@5,10 are similar to the results for NDCG@1.

Table 6.7: NDCG@1 on store chains with different sizes in Phoenix. Best results are in bold.

Models	1-5 (35)	6-10 (16)	11-15 (8)	16-20 (4)	20+ (6)
Rand	0.49	0.49	0.49	0.49	0.49
Pop	0.62	**0.74**	0.54	**0.87**	0.69
LR	0.55	0.55	0.67	0.50	0.76
LF	0.59	0.61	**0.80**	0.60	0.81
FLF	**0.63**	0.70	0.77	0.73	**0.81**

Table 6.8: NDCG@1 on store chains with different sizes in Manhattan. Best results are in bold.

Models	1-5 (31)	6-10 (11)	11-15 (6)	16-20 (7)	20+ (16)
Rand	0.49	0.49	0.49	0.49	0.49
Pop	0.69	0.70	0.73	0.51	0.66
LR	0.58	0.53	0.59	0.62	0.66
LF	0.59	0.61	0.70	0.61	**0.76**
FLF	**0.70**	**0.87**	**0.82**	**0.68**	0.73

We note that the results for Rand depend only on the values of k, and are identical for all size categories. Pop achieves similar performance across the different sizes, because this method is unsupervised so that its performance is independent of the size of the store chains. On the contrary, the NDCG@1 of LR shows an overall increasing trend with increasing size. The low NDCG@1 values of LR compared to the other models for cold start store chains (1-5 and 5-10) support our hypothesis that the LR model fails when there is not sufficient training data. Similarly, the performance of LF and FLF improves as the size of the store chain increases.

The performance of LF and FLF for cold start store chains shows that our proposed methods are fairly robust to the number of training stores. Finally note that the performance gain of FLF over Pop on large chains (11-15, 16-20, 20+) is greater than on small chains. To conclude, we observe that FLF consistently outperform the other comparison partners for all sizes in the Foursquare data set but not in the Yelp data set. Again, we believe that the quality of the generated location features in the Foursquare data set is better than the one in the Yelp data set.

[36] used only three "non cold start" store chains (Starbucks, Dunkin Donuts and McDonald's) in their experiments and reported that for these store chains LR outperformed Pop. This is consistent with our results for "non cold start" chains. They also reported experiments for Pop and showed that it can outperform LR, which is also consistent in our experiments.

6.6 Conclusion

In this chapter's work, we address the problem of finding the optimal location for a new store of the given store chain. Previous work has explored establishing separate regression models for each store chain. To the best of our knowledge, we are the first to formulate this problem as a recommendation problem, i.e., recommending locations to a new store of the given store chain. Hence, store chains can be viewed as "users" and locations can be viewed as "items" in the context of recommendations. The advantage of this problem setting is that 1) it can collaboratively learn the model parameters for all store chains; 2) it can capture the latent factors from the interactions between store chains and locations.

As a result, we propose the first Latent Factor (LF) model and Feature based Latent Factor (FLF) model to capture the latent factors of store chains and locations as well as the observed location features, and consider all store chains together in a single probabilistic model. Specifically, the intuitions behind the models are 1) the number of check-ins of different store chains are correlated to each other at different locations; 2) observed location features can strengthen the latent factors for both store chains and locations. We present the model of LF and FLF, and a stochastic gradient descent method of parameter learning. We perform an experimental evaluation on Yelp and Foursquare data sets. We compare our proposed latent factor models against the existing state-of-the-art methods. Our experiments demonstrate substantially improved performance in location recommendation for new stores.

Chapter 7

Conclusion

Data mining in location-based social networks has become a fascinating research area due to the availability of a huge volume of user-generated content empowered by mobile phones, e.g., reviewing or checking in Point-Of-Interests (POIs). Recommendation in location-based social networks, which aims to recommend items such as POIs in location-based social networks, is a relatively new sub-area that attracted a great deal of attention recently. In this thesis, we focused on this problem because of its key role in the area of location-based social networks. The extracted users' and POIs' topics not only help the POI recommender system but also can be applied to other recommender systems. In Chapter 2, we defined this problem formally and reviewed the state-ofthe-art approaches presented in the literature.

In this thesis, we proposed several probabilistic methods for recommendation in location-based social networks. In Chapter 3, we introduced a spatial topic method [29] for top-k POI recommendation problem. The proposed spatial topic model finds users' topic and region distributions by mining a set of topics and regions from user check-ins with posts and location coordinates, and models coordinates of checked in POIs using a two dimensional Gaussian distribution. Previous works just extract the user preferences on POIs, the proposed model further extracts user preferences on regions and topics. Evaluation of results showed that the proposed model can effectively improve the accuracy of top-k POI recommendation.

Time information associated with check-ins is normally ignored in the existing works, as a result, in Chapter 4 we proposed a spatio-temporal topic model, called STT [31], to learn a set of spatio-temporal topics from the user check-in data. In comparison to the previous works which ignore the time information, STT jointly identifies topics from both the spatial and temporal aspects. In addition, STT captures the geographical influence between user regions and POIs, and temporal activity patterns of different topics and POIs. The experimental evaluation on three real life data sets from Twitter, Gowalla, and Brightkite shows the superiority of STT over the existing state-of-the-art recommendation algorithms and geographical and temporal topic models in terms of likelihood of the test data set and accuracy of top-k POI and time recommendations.

In Chapter 5, we argued that all existing social network-based POI recommendation models cannot capture the nature of location-based social network for top-k POI recommendation. We comprehensively analyzed the nature of POIs and the benefits of a social network for POI recommendation. Then we addressed this problem by proposing a social topic model, called ST [30], which effectively exploits a location-based social network for POI recommendation. In particular, ST models the check-ins with posts and a social network and extracts a set of latent topics. Users' topic distributions are mutually influenced by their friends. On two real life data sets from Foursquare and Yelp, we demonstrated that the ST model consistently improves the performance significantly for POI recommendation compared to existing state-of-the-art social network-based recommendation algorithms for all users, all POIs, cold start users, and cold start POIs.

In Chapter 6, we discussed another interesting recommendation problem in location-based social networks, i.e., determining the optimal location for a new store. To the best of our knowledge, we are the first to formulate this problem as a recommendation problem, i.e., recommending locations to a new store of the given store chain. We proposed latent factor models to solve the recommendation problem, which perform better than existing regression models.

The research of this thesis suggests many promising directions for future work. In this section, we briefly discuss such directions:

- **Cold start users or POIs**:

  Most of the current methods are effective on non-cold start users or POIs. However, they are not effective on cold start users or POIs, which play a big part in location-based social networks in terms of quantity. Although social network-based methods such as the ST model proposed in Chapter 6 address the problem, further research is still needed.

- **Sparsity problem**:

  Most check-in data of location-based social networks are from different cities. On the one hand, it is inappropriate to build a single model on top of all the data because users rarely visit multiple cities. On the other hand, building seperate models for different cities may face a sparsity problem, i.e., some cities might have few check-in data, due to various reasons e.g., data corruption. Further research is needed.

- **Additional contexts**:

  There are some additional contexts that can improve the existing methods, e.g., sentiment analysis or ratings of reviews on POIs. Particularly, the first step could test whether sentiments/ratings of reviews affect other users' check-in behavior, and the second step could model this effect.

- **Comprehensive model**:

  A comprehensive model should be explored so that each context can be used as a plug and play component.

- **User or POI dependent geographical influence**:

  Geographical influence has been proved an important factor that affects a user whether checks in a POI. The current computation of geographical influence depends on the distance between the user and POI, and is independent of users or POIs. However, different users or POIs should have different geographical influence coefficients. For instance, some users like visiting POI in the long distance. Further analysis or modeling should be investigated.

- **Evaluation on different contexts**:

  The impact of different input contexts in topic modeling approach (e.g., user-generated content, coordinates, time, and social network) should be explored. A comprehensive performance comparison is needed to clarify the impact of each input context in improving the performance.

Bibliography

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, 2005. 1, 5, 9, 10

[2] Deepak Agarwal and Bee-Chung Chen. Regression-based latent factor models. KDD, pages 19–28, 2009. 83

[3] Deepak Agarwal and Bee-Chung Chen. flda: matrix factorization through latent dirichlet allocation. WSDM, pages 91–100, 2010.

[4] Natalie Aizenberg, Yehuda Koren, and Oren Somekh. Build your own music recommender by modeling internet radio streams. In *Proceedings of the 19th International Conference on World Wide Web*, WWW, pages 1–10, 2012. 77

[5] Adee Athiyaman. Location decision making: The case of retail service development in a closed population. *Academy of Marketing Studies*, 15(1):13, 2010. 82, 84

[6] Lars Backstrom, Eric Sun, and Cameron Marlow. Find me if you can: Improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th International Conference on World Wide Web*, WWW, pages 61–70, 2010. 27, 28

[7] Marko Balabanović and Yoav Shoham. Fab: Content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, 1997. 5

[8] Jie Bao, Yu Zheng, and Mohamed F. Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL, pages 199–208, 2012. 15, 19, 26

[9] Robert Bell, Yehuda Koren, and Chris Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 95–104, 2007. 2, 7, 9, 10, 93

[10] David M. Blei. Introduction to probabilistic topic models. *Communications of the ACM*, 2011. 18

[11] David M. Blei, Andrew Ng, and Michael Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003. 12, 18, 24, 49, 65, 68

[12] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, UAI, pages 43–52, 1998. 1, 9, 10, 67

[13] Chen Cheng, Haiqin Yang, Irwin King, and Michael R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, AAAI, 2012. 15, 19, 22, 23, 24, 25, 29, 34, 40, 41, 56, 64, 66, 69, 74, 76, 77

[14] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. You are where you tweet: A content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM, pages 759–768, 2010. 27, 28, 39

[15] Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Z. Sui. Exploring millions of footprints in location sharing services. In *Proceedings of the 5th International Conference on Weblogs and Social Media*, ICWSM, 2011. 27, 28, 54

[16] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 1082–1090, 2011. 27, 28, 54

[17] Wei Chu and Seung-Taek Park. Personalized recommendation on dynamic content using predictive bilinear models. WWW, pages 691–700, 2009. 94

[18] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the 4th ACM Conference on Recommender Systems*, RecSys, pages 39–46, 2010. 7, 8, 17

[19] Jacob Eisenstein, Amr Ahmed, and Eric P. Xing. Sparse additive generative models of text. ICML, pages 1041–1048, 2011. 35, 50, 71

[20] Huiji Gao, Jiliang Tang, and Huan Liu. gscorr: Modeling geo-social correlations for new check-ins on location-based social networks. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM, pages 1582–1586, 2012. 15, 19, 20, 21, 22, 64, 65, 66, 74

[21] Brent Hecht, Lichan Hong, Bongwon Suh, and Ed H. Chi. Tweets from justin bieber's heart: the dynamics of the location field in user profiles. In *CHI*, pages 237–246, 2011. 27, 28

[22] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004. 1, 7, 8, 17, 93, 97

[23] Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI, pages 194–201, 1995. 1, 5

[24] Thomas Hofmann. Probilistic latent semantic analysis. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, UAI, 1999. 12, 18

[25] Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J. Smola, and Kostas Tsioutsiouliklis. Discovering geographical topics in the twitter stream. WWW, pages 769–778, 2012. 29, 30, 33, 38, 39, 40, 41, 54, 56

[26] Liangjie Hong, Ron Bekkerman, Joseph Adler, and Brian D. Davison. Learning to rank social update streams. SIGIR, pages 651–660, 2012.

[27] Liangjie Hong, Aziz S. Doumith, and Brian D. Davison. Co-factorization machines: modeling user interests and predicting individual decisions in twitter. WSDM, pages 557–566, 2013.

[28] Liangjie Hong, Dawei Yin, Jian Guo, and Brian D. Davison. Tracking trends: incorporating term volume into temporal topic models. KDD, pages 484–492, 2011.

[29] Bo Hu and Martin Ester. Spatial topic modeling in online social media for location recommendation. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys, pages 25–32, 2013. 77, 102

[30] Bo Hu and Martin Ester. Social topic modeling for point-of-interest recommendation in location-based social networks. In *Proceedings of the 14th IEEE International Conference on Data Mining*, ICDM, 2014. 103

[31] Bo Hu, Mohsen Jamali, and Martin Ester. Spatio-temporal topic modeling in mobile social media for location recommendation. In *Proceedings of the 13th IEEE International Conference on Data Mining*, ICDM, 2013. 102

[32] Mohsen Jamali and Martin Ester. Trustwalker: A random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 397–406, 2009. 14, 15

[33] Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the 4th ACM Conference on Recommender Systems*, RecSys, pages 135–142, 2010. 14, 15, 64, 65, 77, 83

[34] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems An Introduction.* Cambridge University Press, 2011. 5

[35] Pablo Jensen. Analyzing the localization of retail stores with complex systems tools. IDA, pages 10–20, 2009. 82, 84

[36] Dmytro Karamshuk, Anastasios Noulas, Salvatore Scellato, Vincenzo Nicosia, and Cecilia Mascolo. Geo-spotting: mining online location-based services for optimal retail store placement. KDD, pages 793–801, 2013. 82, 83, 85, 86, 89, 90, 92, 96, 97, 98, 101

[37] Noriaki Kawamae. Trend analysis model: trend consists of temporal words, topics, and timestamps. WSDM, pages 317–326, 2011.

[38] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 426–434, 2008. 2, 7, 9, 10, 11, 17, 18, 67, 93

[39] Yehuda Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 447–456, 2009. 2, 7, 9, 10, 67, 93

[40] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42:30–37, 2009. 2, 7, 9, 10, 11, 18, 67, 93, 94

[41] Yehuda Koren and Joe Sill. Ordrec: an ordinal model for predicting personalized item rating distributions. In *Proceedings of the 5th ACM Conference on Recommender Systems*, RecSys, pages 117–124, 2011. 7, 8

[42] Takeshi Kurashima, Tomoharu Iwata, Takahide Hoshide, Noriko Takaya, and Ko Fujimura. Geo topic model: joint modeling of user's activity area and interests for location recommendation. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, WSDM, pages 375–384, 2013. 15, 19, 22, 23, 29, 34, 38, 39, 40, 41, 44, 46, 49, 54, 55, 56

[43] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, January 2003. 1, 2

[44] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. Learning geographical preferences for point-of-interest recommendation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 1043–1051, 2013. 15, 19, 22, 24, 77

[45] Bin Liu and Hui Xiong. Point-of-interest recommendation in location based social networks with topic and location awareness. In *Proceedings of the 13th SIAM International Conference on Data Mining*, SDM, pages 396–404, 2013. 15, 19, 25, 27

[46] Dong C. Liu, Jorge Nocedal, Dong C. Liu, and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, pages 503–528, 1989. 37, 52, 72

[47] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. Sorec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM, pages 931–940, 2008. 14, 15, 64, 77

[48] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, WSDM, pages 287–296, 2011. 14, 15, 64, 69, 76, 77

[49] Paolo Massa and Paolo Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, pages 17–24, 2007. 14, 15, 77

[50] Bradley N. Miller, Istvan Albert, Shyong K. Lam, Joseph A. Konstan, and John Riedl. Movielens unplugged: Experiences with an occasionally connected recommender system. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI, pages 263–266, 2003. 1

[51] Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. Mining user mobility features for next place prediction in location-based services. In *Proceedings of the 12th IEEE International Conference on Data Mining*, ICDM, pages 1038–1043, 2012. 27, 28

[52] Michael J. Pazzani and Daniel Billsus. The adaptive web. chapter Content-based Recommendation Systems, pages 325–341. 2007. 9

[53] Sergio Porta, Vito Latora, Fahui Wang, Emanuele Strano, Alessio Cardillo, Salvatore Scellato, Valentino Iacoviello, and Roberto Messora. Street centrality and densities of retail and services in bologna, italy. *Environment and Planning B: Planning and Design*, 36(3):450–465, 2009. 82, 84

[54] Christian Posse. Key lessons learned building recommender systems for large-scale social networks. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 587–587, 2012. 3

[55] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW, pages 175–186, 1994. 1, 5

[56] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. *Recommender Systems Handbook*. Springer, 2011. 3, 5, 9

[57] Adam Sadilek, Henry Kautz, and Jeffrey P. Bigham. Finding your friends and following them to where you are. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, WSDM, pages 723–732, 2012. 27, 28

[58] Ruslan Salakhutdinov and Mnih Andriy. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*. 2, 7, 9, 10, 18, 40, 55, 67, 76

[59] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW, pages 285–295, 2001. 1, 7, 9, 10, 67

[60] Salvatore Scellato, Anastasios Noulas, and Cecilia 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*, KDD, pages 1046–1054, 2011. 27, 65, 74

[61] Mark Schmidt, Alexandru Niculescu-Mizil, and Kevin Murphy. Learning graphical model structure using l1-regularization paths. AAAI, pages 1278–1283, 2007. 37, 52, 72

[62] Yelong Shen and Ruoming Jin. Learning personal + social latent factor model for social recommendation. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 1303–1311, 2012. 14, 15

[63] Sergej Sizov. Geofolk: latent spatial semantics in web 2.0 social media. WSDM, pages 281–290, 2010. 29, 30, 41, 56

[64] Chong Wang, Jinggang Wang, Xing Xie, and Wei-Ying Ma. Mining geographic knowledge using location aware topic model. GIR, pages 65–70, 2007. 41, 56

[65] Jian Wang, Yi Zhang, Christian Posse, and Anmol Bhasin. Is it time for a career switch? WWW, pages 1377–1388, 2013. 55

[66] Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. KDD, pages 424–433, 2006. 54, 56

[67] Markus Weimer, Alexandros Karatzoglou, Quoc V. Le, and Alex J. Smola. Cofi rank - maximum margin matrix factorization for collaborative ranking. In *Proceedings of the 21st Annual Conference on Neural Information Processing Systems*. 7, 8

[68] Shuang-Hong Yang, Bo Long, Alex Smola, Narayanan Sadagopan, Zhaohui Zheng, and Hongyuan Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th International Conference on World Wide Web*, WWW, pages 537–546, 2011. 14, 15, 94

[69] Xiwang Yang, Harald Steck, Yang Guo, and Yong Liu. On top-k recommendation using social networks. In *Proceedings of the 6th ACM Conference on Recommender Systems*, RecSys, pages 67–74, 2012. 20, 64, 65, 76, 77

[70] Mao Ye, Xingjie Liu, and Wang-Chien Lee. Exploring social influence for recommendation: A generative model approach. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR, pages 671–680, 2012. 15, 19, 23

[71] Mao Ye, Peifeng Yin, and Wang-Chien Lee. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS, pages 458–461, 2010. 15, 19, 20, 22, 23, 29, 40, 55, 56, 64, 66, 74

[72] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR, pages 325–334, 2011. 15, 19, 22, 23, 24, 44, 56, 64, 66, 74, 77

[73] Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. Lcars: A location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 221–229, 2013. 15, 19, 25, 27, 77

[74] Zhijun Yin, Liangliang Cao, Jiawei Han, Chengxiang Zhai, and Thomas Huang. Geographical topic discovery and comparison. WWW, pages 247–256, 2011. 29, 30, 33, 41, 56

[75] Vincent W. Zheng, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative filtering meets mobile recommendation: A user-centered approach. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, AAAI, 2010. 15, 19, 25, 26

[76] Vincent W. Zheng, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th International Conference on World Wide Web*, WWW, pages 1029–1038, 2010. 15, 19, 25