Data Analysis on Location-based Social Networks

by

Fengjiao Wang
B.S., Information and Computational Science, Beihang University, 2010

Thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2017

Chicago, Illinois

Defense Committee:
Philip S. Yu, Chair and Advisor
Bing Liu
Angus Forbes
Yuheng Hu, Department of Information and Decision Sciences
Xiangnan Kong, Worcester Polytechnic Institute

This thesis is dedicated to my parents and my husband

for their endless love, support,

and encouragement.

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor Prof. Philip S. Yu for the continuous support of my PhD study and related research, for his patience, motivation, and immense knowledge. His guidance helped me all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my PhD study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Bing Liu, Prof. Angus Forbes, Prof. Yuheng hu and Prof. Xiangnan Kong, for taking their valuable time to serve on my thesis committee and their insightful comments and encouragement.

I thank my fellow colleagues for the stimulating discussions, for countless things they taught me, and for all the fun we have had in the last seven years.

Finally, I would like to give my deepest thanks to my parents, Lisheng Wang and Jianmei Tian, for their unconditional support in every possible way, for their endless love, for every moment of my life with them being my parents. Especially, I want to thank my husband, Dr. Yongzhi Qu. Without his countless support and encouragement, I would not be who I am.

FW

CONTRIBUTION OF AUTHORS

Chapter 1 is an introduction that outlines my dissertation research. Chapter 1 presents a published manuscript (1) for which I was the primary author. Dr. Guan Wang, Dr. Shuyang Lin, and my advisor Professor Philip S. Yu contributed to discussions with respect to the work and revising the manuscript. Chapter 3 presents a published manuscript (2) for which I was the primary author. Dr. Shuyang Lin, and my advisor Professor Philip S. Yu contributed to discussions with respect to the work and revising the manuscript. Chapter 4 presents a published manuscript (3) for which I was the primary author. Chun-Ta Lu, Dr. Yongzhi Qu, and Professor Philip S. Yu contributed to discussions with respect to the work and revising the manuscript. Chapter 5 presents a published manuscript (4) for which I was the primary author. Dr. Yongzhi Qu, Lei Zheng, Chun-Ta Lu, and Professor Philip S. Yu contributed to discussions with respect to the work and revising the manuscript.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

SUMMARY

As location-based services rapidly gain popularity, a large volume of check-ins is created daily. Each check-in shows who visited what place at what time. It is an online activity but reflects users' real-world lives, therefore it serves as a direct channel connecting the online and offline worlds. Effective modeling of check-ins can aid the development of many personalized and locational information services, such as the personalized advertisement, local event promotion and city management improvement.

Location-based social networks (LBSNs) are formed with check-ins as building blocks and share the basic structure of traditional social networks. LBSN data has the following distinct properties: geographical property, large-scale mobile data, accurate description of geolocations, data sparseness. Inspired by the unique characteristics of LBSN data, my research focuses on modeling and mining knowledge with regards to LBSN data to facilitate multiple applications.

In this thesis, I will introduce our latest research progress on mining and modeling tasks with regards to LBSN data. The first part of this dissertation will focus on understanding the fundamental relationship between users and POIs through modeling check-ins and side information. We propose a goal-oriented co-clustering model which tries to inject customized information into the co-clustering process (1). Moreover, we propose a collaborative co-clustering framework in which users and POIs from multiple linked social networks are modeled simultaneously to reinforce the clustering performances in every single source (2). Next, we propose a network embedding model which incorporates physical meanings in the embedding process to tackle user geolocation problem (3). In the following, we study the POI recommendation problem by proposing a deep learning framework which models different models different information sources collectively (4).

CHAPTER 1

INTRODUCTION

1.1 Thesis Outline

Check-in, one of the most popular social activities, performed by a user who shares the information in social media that he/she visited a physical place. The physical place being checked in by the user is called Point of Interest (POI). A POI is a specific point location that someone may find interesting and be willing to check in. Check-in is the essential behavior in *Location-based Social Networks* (LBSN). Most common LBSNs are Foursquare, Gowalla, Brightkite, and Loopt. Other traditional social networking sites also incorporate location-based services, such as Facebook, Twitter, Google+, Yelp, and Flickr. These LBSNs not only provide services which can meet socialization needs of users but also allow users to post whereabouts or attach their posted text or image with meaningful location information. Therefore, These social sites are actually very attractive to users and they can maintain a large number of active users regularly.

Check-in is a unique type of social activities as it shows users' real life in the virtual world. In this way, check-in connects online and offline worlds. Therefore, it provides a great opportunity to study users' behaviors. Moreover, studying users' check-ins could potentially help many applications such as the personalized advertisement, traffic management, and event promotion.

In this thesis, we focus on the modeling and mining knowledge with regards to LBSN data to apply in various real-world applications. Our work in this thesis covers four different yet correlated research directions:

- **First**, we model the interconnected relationships of users and POIs through check-ins and other side information. A goal-oriented co-clustering framework is proposed which integrates subspace learning and co-clustering to generate effective user groups and POI clusters.
- **Second**, we model user behavior patterns and POIs' characteristics based on multiple information sources. We propose a multi-view co-clustering framework to learn meaning meaningful user groups and POI clusters from multi-sourced information networks.
- **Third**, we focus on predicting users' home locations by connecting multiple sources of signals with a unified geographical embedding framework. It virtually embeds each user and POI into the embedding space by utilizing users' friendship networks and check-ins.
- **Finally**, we focus on POI recommendation task through modeling two different source information. We propose a deep learning framework to effectively model user preference, sequential patterns, and POIs' characteristics.

1.2 Concurrent Goal-oriented Co-clustering in LBSN

(Part of the section was previously published in (1).)

Users and POIs are two essential objects in LBSN. Modeling user behavior and discovering characteristics of POIs is important to understand fundamental patterns in LBSN. Co-clustering is a classic technique which can be utilized to identify patterns existing between two closely related objects, such as users and POIs in LBSNs. User groups and POI clusters can further facilitate many applications. However, the co-clustering process is usually performed without considering customer's goals (e.g. a customer may want to explore the city to find the POI within the same neighborhood but in different categories).

In Chapter 2, we study the problem of goal-oriented co-clustering. Instead of achieving one optimal co-clustering, goal-oriented co-clustering intend to detect multiple different co-clusterings with regards to multiple goals. We devise an effective way to represent clustering goal by introducing seed feature sets. It is hard for traditional semi-supervised co-clustering algorithms to involve user-provided information. Each clustering goal is represented by a set of features. Moreover, this set of features will be automatically learned through subspace learning technique. To ensure the quality of multiple co-clusterings, we integrate spectral co-clustering technique with subspace learning which iteratively improving the quality of generated co-clusters.

1.3 Collaborative Co-clustering across Multiple Social Media

(Part of the section was previously published in (2).)

For the social media websites such as Twitter, Facebook, Google+, there are usually two types of objects, social media objects (tweet, review, video, and POI) and user objects. User objects are unique as users can perform various actions (e.g. following, voting, tagging, reviewing, etc.) across different social media websites. Sometimes, the same user may have multiple accounts. In this way, different social media websites can be linked by the same set of users. Usually, combining information from different sources can help better solve the problem.

In Chapter 3, we focus on the co-clustering problem across multiple social media. One benefit that considering multiple source information together is that information in different sources could reinforce or compensate each other. However, co-clustering across multiple social media is a challenging task, it is hard to devise an effective framework to combine multiple information sources. To solve this challenging problem, we make the following effort. First of all, we cast the problem into multi-view learning problem. Multiple heterogeneous sources are transformed accordingly to the multi-view setting. A co-regularized technique is introduced to impose a common constraint between different sources such that clusters from different sources can be unified. Experiments on real-world social network dataset demonstrate the effectiveness of the proposed model.

1.4 Collective Geographical Embedding for Geolocating Social Network Users

(Part of the section was previously published in (3).)

Understanding user behavior, summarizing user's mobility pattern, recommending interesting POIs are problems which directly stemmed from LBSNs. One of the core tasks also lie in these location-based services is inferring the physical location of users. Effectively solving user geolocation problem could potentially benefit all the above-mentioned problems in LBSN.

In Chapter 4, we study the problem of geolocating social media users. However, there are several challenges for this task. First of all, data sparsity hinders the performance as many users may tend to hide their location information due to privacy concerns. Secondly, noisy signal is another big challenge, as friendship information in online social networks may be an irrelevant information for the task. (e.g. global fans follow their celebrities.) The third challenge is scalability. To cope with these challenges, we propose a **C**ollective **G**eographical **E**mbedding (CGE) algorithm to embed users and POIs from multiple information sources into the same low dimensional space, with the property that the distance in the embedding space can reflect the physical distance in the real world. Such property is achieved by incorporating a location affinity matrix as the constraint to the process of heterogeneous network embedding. Extensive experiments performed on the real-world LBSN datasets demonstrate the effectiveness of the proposed CGE model.

1.5 Deep and Broad Learning on Content-aware POI Recommendation

(Part of the section was previously published in (4).)

In Chapter 5, we focus on the POI recommendation in LBSN. It is the most effective tool for location-based service providers to attract users. However, most of the previous works focusing on modeling multi-source information (e.g. sequential information, user preference, POI properties) in a flat manner. It means that different source information are treated indiscriminately, so they may unable to bring the full power of combining multiple source information.

We propose a **D**eep **C**ontext-aware **P**OI **R**ecommendation model (DCPR) to structurally learn user and POI properties in deep learning framework. To be specific, the proposed model contains three collaborative layers to capture multiple source information: a) a CNN layer for learning POI characteristics; b) an RNN layer for sequential modeling and user preference learning; c) a joint optimization layer to learn users' check-in behavior. Experiments on Foursquare and Yelp dataset show that DCPR model achieves significant improvement over state-of-the-art deep recommendation models.

CHAPTER 2

CONCURRENT GOAL-ORIENTED CO-CLUSTERING IN LBSN

(This chapter was previously published as "Concurrent Goal-oriented Co-clustering Generation in Social Networks", in *IEEE International Conference on Semantic Computing (ICSC '15)* (1). DOI: https://doi.org/10.1109/ICOSC.2015.7050833.)

2.1 Introduction

Co-clustering, the process of clustering two types of objects, has become a popular topic in many applications. Co-clustering can be applied to various data mining applications, for example, in text mining to identify similar documents and word clusters (5), in social recommendation system to create recommendation systems that predict movie ratings based on the co-clustering relationship between user groups and movie clusters (6), in academic networks to explore author groups and their interplay with conference clusters (7).

User's expectation (clustering goal) is a critical objective in co-clustering. Unfortunately, most existing co-clustering approaches did not consider user's expectation, since they just generate groups of similar objects. Whether the grouping results can live up to user's expectation or not is beyond the scope of existing approaches. Such approaches may lead to undesirable co-clusterings. Therefore, the traditional co-clustering algorithms are missing the following two aspects w.r.t. goals.

- **They are unable to concurrently generate multiple co-clusterings according to different goals.** User may have varying expectations (different clustering goals) for co-clustering. For example, in academic networks, a user who wants to find group of authors with the same affiliation may also be looking for groups of authors tackling similar problems. When exploring the same data, different users demand different co-clusterings.
- **They are unable to pick most relevant features for multiple co-clusterings with different goals *together*.** Different features are related to different goals. To determine which feature is more important to which goal, it is beneficial to consider multiple goals simultaneously.

In this paper, we propose a new approach, namely goal-oriented co-clustering, to solve co-clustering problem. Rather than obtaining one optimal co-clustering from the data, goal-oriented co-clustering finds different co-clusterings with regards to different goals. There are three key challenges to fulfill this purpose.

- **Devising effective ways of capturing user provided information:** Traditional unsupervised co-clustering did not consider any user provided information. Therefore, it is hard to perform co-clustering with a desired goal. Another algorithm available in literature called semi-supervised co-clustering (8) could utilize user provided information to help with co-clustering. This technique requires users to provide a large amount of information that contains "must-link" or "cannot-link" objects, which are co-clustering constraints. In order to achieve desired co-clustering results, we need reasonable large set of such constraints. It is unrealistic for users to provide a high quality set of constraints.

- **Utilizing multiple features to represent goals:** We should keep in mind that each type of object could associate with multiple mutually unrelated features. The objects can be co-clustered according to different goals. Each goal is depicted by a bunch of features. In order to get goal-oriented co-clusterings, we need to assign features specifically casted to that goal. How to make full use of these features towards different goals is a big challenge. If all features are utilized indiscriminately, they could interfere with each other, resulting in wasted features as well as poor co-clustering output.

- **Concurrent co-clustering on different goals:** To ensure the quality of co-clusterings, it is beneficial to learn subspace and iteratively improve the associated feature set for each goal. Current co-clustering approaches did not provide any learning mechanism during co-clustering process. By integrating co-clustering and subspace learning technique, these two tasks could reinforce each other and achieve better results.

Location-based social network has attracted many attentions recently. Mobile users share the places they visited by "check-in" to places. To encourage mobile users to explore new places, location recommendation service is an essential function to website providers. For this reason, location recommendation has emerged as a hot topic recently. Co-clustering technique has been proved to be a powerful technique in social recommendation (7). It is encouraged to explore goal-oriented co-clustering technique to help location recommendation.

In Figure 1, we use Foursquare data as an example to explain the motivation of multiple goal co-clusterings. The left box represents Foursquare data set which contains check-in information. We have two users with different goals. The first one wants to search a group of places in the

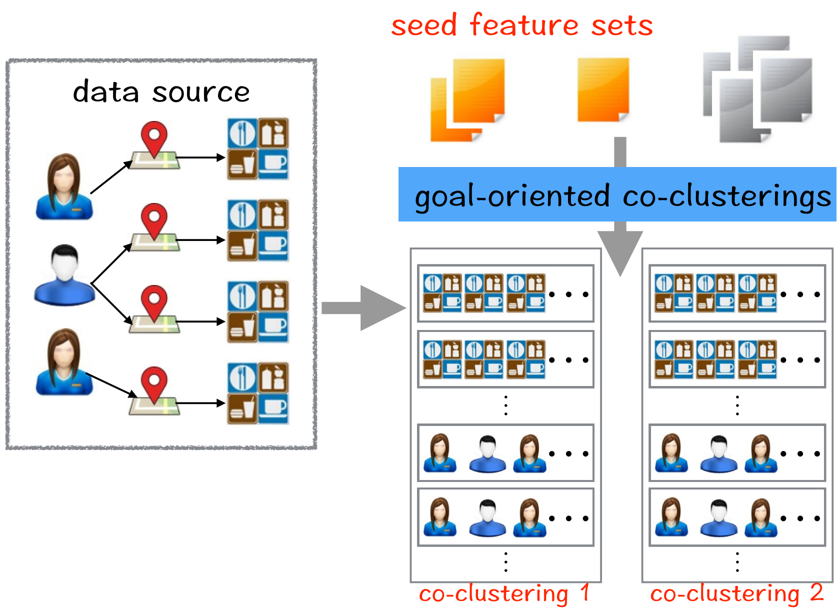


Figure 1. Example of goal-oriented co-clustering model.

same neighborhood and he selects the city and zip code as seed features (orange file), while the second one wants to search a group of places of one category and she uses keywords like 'Entertainment' as seed feature (orange file). Each set of features (could be only one) selected by users is defined as a seed feature set. Inputs of goal-oriented co-clustering model include seed feature sets (in this scenario, two seed feature sets) and other features which are not specified by users. Eventually, the model creates two co-clusterings. In co-clustering 1, places in the same neighborhood will be clustered into one group, and users always checking in at the same neighborhood will be clustered into one group. In co-clustering 2, places with similar functions will be grouped into one cluster, and users always checking in at places with similar functions will be grouped into one cluster.

To summarize, our goal-oriented co-clustering models are novel in four folds.

- **goal-based approach:** We introduce a novel framework to consider goal-oriented idea in the setting of co-clustering.
- **seed feature expansion to capture goal:** We devise an approach to utilize user provided information to select goal-related features.
- **subspace and spectral learning:** We integrate subspace learning technique with spectral learning based co-clustering to avoid learning unrelated co-clustering.
- **location-based social network application:** We apply goal-oriented co-clustering model on location-based social network data to cluster users and places.

The experimental evaluation shows that the proposed model can result in better performance with regards to clustering quality. Additional, three case studies are performed to demonstrate the effectiveness of co-clustering for users and places. We also illustrate a possible way of making recommendation in social networks through one case study.

2.2 Preliminaries

In this section, we formally define the problem of goal-oriented co-clusterings. First of all, we introduce some notation conventions. Capital-based letters such as $\mathbf{E}$, $\mathbf{K}$, $\mathbf{D}$ are used as matrices and script letters such as $\mathcal{V}_r$, $\mathcal{V}_c$ as vertex sets. $\mathbf{E_{ij}}$ denotes the (i, j)-th element of $\mathbf{E}$.

Co-clustering allows clustering of the rows and columns of a matrix simultaneously. Spectral co-clustering is one popular approach among co-clustering algorithms that transforms co-clustering problem as a partition problem on a bipartite graph. Since the proposed models are based on spectral co-clustering, graph based notations will be used.

Denote the bipartite graph as $G = (\mathcal{V}_r, \mathcal{V}_c, \mathbf{E})$. It contains two sets of vertices $\mathcal{V}_r$ and $\mathcal{V}_c$. In this paper, set $\mathcal{V}_r$ is the set of places (businesses) and set $\mathcal{V}_c$ the set of users (reviewers). For convenience of discussion, we call the vertices in $\mathcal{V}_r$ as "place vertices", while vertices in $\mathcal{V}_c$ as "user vertices". Matrix $\mathbf{E}$ is composed of elements that represent edges between place vertices and user vertices. Each element, $\mathbf{E_{ij}}$, is a check-in (review) performed by a user $v_{ci} \in \mathcal{V}_c$ in one place (business) $v_{rj} \in \mathcal{V}_r$. Therefore, the adjacency matrix of the bipartite graph, denoted as $\mathbf{K}$, can be written as

$$\mathbf{K} = \begin{bmatrix} \mathbf{0} & \mathbf{E} \\ \mathbf{E}^T & \mathbf{0} \end{bmatrix} \tag{2.1}$$

2.3 Proposed Model

Spectral Co-Clustering Spectral co-clustering is a co-clustering algorithm that transforms co-clustering problem as a partition problem on a bipartite graph. "Row vertices" and "column vertices" in the context of bipartite graph refer to original rows and columns of matrix in co-clustering problem. Each edge in the bipartite graph corresponds to an element of matrix in co-clustering problem. The partition problem on a bipartite graph aims to simultaneously partition row vertices $\mathcal{V}_r$ into k place clusters and column vertices $\mathcal{V}_c$ into k user clusters. Spectral co-clustering tends to find minimum cut vertex partitions in a bipartite graph between row vertices and columns vertices. The optimal solution of this graph partitioning problem can be solved by calculating eigenvectors of a system.

Goal-oriented co-clustering framework We propose a framework for goal-oriented co-clustering which contains two components. One component is to generate multiple co-clusterings, which is the main purpose of the framework. The other component is a subspace learning technique. The subspace learning technique is optional for the purpose of goal-oriented co-clustering. However, this technique could significantly enhance the goal-oriented co-clustering results. We propose two models, simple goal-oriented co-clustering model (SGCC) and full goal-oriented co-clustering model (FGCC). SGCC model only contains the first component. It takes seed feature sets as input and directly produce co-clusterings. Note that the seed feature sets may have not covered all features. Those features not yet included in the seed feature sets may also be helpful for improving co-clustering quality. Therefore, we propose FGCC model to handle additional features. For the specific location-based social network co-clustering problem we study in this paper, the aforementioned goals are all about places and the features are also for places. Although A few features for user are also utilized in the proposed algorithms, features for user are not necessary for goal-oriented co-clustering. Therefore, user features are included for all goals and they are not processed in subspace learning. The feature in the following context will refer to place features exclusively. The following Sections 2.3.1 and 2.3.2 present SGCC and FGCC models in details.

2.3.1 Simple Goal-oriented Co-clustering Model (SGCC)

In this section, we will introduce simple goal-oriented co-clustering model (SGCC), the first proposed model under the goal-oriented co-clustering framework. SGCC model takes seed feature sets as input and directly produces co-clusterings. Since seed feature sets are provided by users, they contain information related to user's clustering goals. Therefore, the seed feature sets can be used to supervise co-clustering towards user's clustering goals.

As defined in Section 2.2, given a bipartite graph $G = (\mathcal{V}_r, \mathcal{V}_c, \mathbf{E})$, not only the edge weight between different type of objects, such as place vertices and user vertices are considered, similarities between the same type of objects utilizing object's features are also considered. Thus, with these information taken into account, the adjacency matrix becomes

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_r & \mathbf{E} \\ \mathbf{E}^T & \mathbf{K}_c \end{bmatrix} \tag{2.2}$$

where matrix $\mathbf{K}_r$ is the similarity matrix of place vertices and matrix $\mathbf{K}_c$ is the similarity matrix of user vertices. In this case, the graph is no longer a bipartite graph since it includes links between any two vertices of the same kind. Since place features for different goal are different, matrix $\mathbf{K}_r$ will vary. Matrix $\mathbf{K}_c$ remains the same for different co-clustering. The objective function of multiple co-clusterings is defined as:

$$\begin{aligned} \underset{U_q}{\text{minimize}} \quad & \sum_q \mathrm{Tr}(\mathbf{U}_q^T \mathbf{L}_q \mathbf{U}_q) \\ \text{subject to} \quad & \mathbf{U}_q(k)^T \mathbf{U}_q(s) = \mathbf{0}, \text{ if } k \neq s \\ & \mathbf{U}_q^T \mathbf{U}_q = \mathbf{I} \end{aligned} \tag{2.3}$$

where $\mathbf{U}_q = \begin{bmatrix} \mathbf{U}_{qr} \\ \mathbf{U}_{qc} \end{bmatrix}$, $\mathbf{L}_q = \mathbf{D} - \mathbf{K}_q$, $\mathbf{D} = \begin{bmatrix} \mathbf{D}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_c \end{bmatrix}$, $\mathbf{K}_q = \begin{bmatrix} \mathbf{K}_{qr} & \mathbf{E} \\ \mathbf{E}^T & \mathbf{K}_c \end{bmatrix}$, $[\mathbf{D}_r]_{ii} = \sum_j \mathbf{E}_{ij}$, $[\mathbf{D}_c]_{ii} = \sum_j \mathbf{E}_{ji}$, $\mathbf{U}_q$ is the q-th co-clustering solution. $\mathbf{U}_{qr}$ is the place vertex partition matrix of the q-th co-clustering and $\mathbf{U}_{qc}$ is the user vertex partition matrix of the q-th co-clustering. The entry $[\mathbf{U}_{qr}]_{ij} = 1$ if and only if place vertex v_{ri} belongs to j-th place cluster. $\mathbf{U}_q(k)$ and $\mathbf{U}_q(s)$ are k-th and s-th columns of $\mathbf{U}_q$ respectively. $\mathbf{K}_q$ is the adjacency matrix corresponding to q-th co-clustering. $\mathbf{K}_{qr}$ is the similarity matrix of place vertices corresponding to q-th co-clustering. Matrix $\mathbf{L}_q$ is a laplacian matrix. Laplacian matrix is a matrix representation of the graph. According to spectral graph theory, we can study the property of bipartite graph by studying the fundamental characteristics of laplacian matrix, such as eigenvalues and eigenvectors, etc. The constraint is selected to satisfy the following criterion: for a specific co-clustering goal, a single object cannot belong to multiple clusters. Each co-clustering solution is achieved by selecting k left and k right eigenvectors of the matrix $(\mathbf{D}_r - \mathbf{K}_{qr})^{-1/2}\mathbf{E}_q(\mathbf{D}_c - \mathbf{K}_c)^{-1/2}$.

2.3.2 Full Goal-oriented Co-clusterings Model (FGCC)

In Section 2.3.1, the proposed SGCC model only takes seed feature sets as input. This could omit other useful information. In order to take into consideration of other features, we further incorporate subspace learning technique. The goal of subspace learning is to find several low-dimensional subspaces of features. Each low-dimensional subspace corresponds to one goal. In the proposed model, subspace learning determines whether features not included in seed feature sets should be fully or partially tied to each co-clustering.

The learning of the subspace in each co-clustering is done by integrating dimensionality reduction with spectral co-clustering. In each co-clustering, the kernel similarity matrix K is computed in subspace. Each element of the kernel similarity matrix is calculated based on kernel function $k(\mathbf{W}_q^T v_{ri}, \mathbf{W}_q^T v_{rj})$, where $\mathbf{W}_q \in R^{d \times l_q}$ is a transformation matrix for each co-clustering that transforms $v_{ri} \in R^d$ from the original space to a lower-dimensional space l_q. Hilbert-Schmidt Independence Criterion (HSIC) is used to measure non-linear dependencies between features in different co-clusterings. HSIC was introduced as a penalty term which aims at finding subspaces as different as possible for different goal-oriented co-clusterings. Assume we have a set of n places $\mathcal{V}_r = \{v_{r1}, ..., v_{rn}\}$ and a set of m users $\mathcal{V}_c = \{v_{c1}, ..., v_{cm}\}$. Each v_{ri} is a column vector in $\mathbb{R}^d$ that contains all features of a place. Each v_{cj} is a column vector in $\mathbb{R}^s$ that includes all features of a user. HSIC measures the dependency between two random variables. In this paper, HSIC measures dependency between two different subspaces. HSIC is defined using kernel similarity matrix $\mathbf{K}$, as follows,

$$HSIC = (n-1)^{-2} tr(\mathbf{K}_1 \mathbf{H} \mathbf{K}_2 \mathbf{H}) \tag{2.4}$$

where $\mathbf{K}_1, \mathbf{K}_2 \in R^{n\times n}$ are kernel similarity matrix, $[\mathbf{K}_1]_{ij} = k(\mathbf{W}_{q1}^T v_{ri}, \mathbf{W}_{q1}^T v_{rj})$, $[\mathbf{K}_2]_{ij} = k(\mathbf{W}_{q2}^T v_{ri}, \mathbf{W}_{q2}^T v_{ri})$, and $[\mathbf{H}]_{ij} = \delta_{ij} - n^{-1}$, δ_{ij} is the indicator function which takes 1 when $i = j$ and 0 otherwise. Matrix $\mathbf{H}$ centers the kernel similarity matrix to have zero mean in the feature subspace.

Now we denote the co-clustering solution matrix as $\mathbf{U}$ with regard to one co-clustering goal. $\mathbf{U} = \begin{bmatrix} \mathbf{U}_r \\ \\ \mathbf{U}_c \end{bmatrix}$, where $\mathbf{U}_r$ is the place vertex partition matrix, and $\mathbf{U}_c$ is the user vertex partition matrix. The entry $[\mathbf{U}_r]_{ij} = 1$ if and only if the place vertex v_i belongs to j-th place cluster. The object function is as follows:

$$\begin{aligned}
\underset{U_q, W}{\text{minimize}} \quad & \sum_q \mathrm{tr}(\mathbf{U}_q^T \mathbf{L}_q \mathbf{U}_q) \\
& + \lambda \sum_{q1 \neq q2} \mathrm{HSIC}(\mathbf{W}_{q1}^T v_r, \mathbf{W}_{q2}^T v_r) \\
\text{subject to} \quad & \mathbf{U}_q(k)^T \mathbf{U}_q(s) = \mathbf{0}, \text{ if, } k \neq s \\
& \mathbf{W}_q^T \mathbf{W}_q = \mathbf{I} \\
& \mathbf{U}_q^T \mathbf{U}_q = \mathbf{I}
\end{aligned} \tag{2.5}$$

where $\mathbf{L}_q = \mathbf{D} - \mathbf{K}_q$, $\mathbf{D} = \begin{bmatrix} \mathbf{D}_r & \mathbf{0} \\ \\ \mathbf{0} & \mathbf{D}_c \end{bmatrix}$, $\mathbf{K}_q = \begin{bmatrix} \mathbf{K}_{qr} & \mathbf{E} \\ \\ \mathbf{E}^T & \mathbf{K}_c \end{bmatrix}$, $[\mathbf{K}_q]_{ij} = k_q(\mathbf{W}_q^T v_{ri}, \mathbf{W}_q^T v_{rj})$, $[\mathbf{D}_r]_{ii} = \sum_j \mathbf{E}_{ij}$, $[\mathbf{D}_c]_{ii} = \sum_j \mathbf{E}_{ji}$, $\mathbf{U}_q(k)$ and $\mathbf{U}_q(s)$ are k-th and s-th columns of $\mathbf{U}_q$ respectively, $v_{ri} \in R^d$, $\mathbf{W}_{q1} \in R^{d\times l_q}$, and $\mathbf{W}_{q2} \in R^{d\times l_q}$. $\mathbf{W}_{q1}$ and $\mathbf{W}_{q2}$ are two transformation matrices. The first term in the objective function $\sum_q \mathrm{Tr}(\mathbf{U_q}^T \mathbf{L_q} \mathbf{U_q})$ is the relaxed spectral clustering objective for each co-clustering and it helps to optimize cluster quality. The second term, $\lambda \sum_{q1 \neq q2} \text{HSIC}(\mathbf{W}_{q1}^T x_r, \mathbf{W}_{r2}^T x_r)$, is designed to penalize for overlaps of subspaces. Simply optimizing one of these criteria is not sufficient to produce multiple high-quality co-clusterings. The parameter λ is a regularization parameter that controls the trade-off between these two criteria.

2.3.3 Full Goal-oriented Co-clusterings Algorithm

Now, we describe the procedure to optimize the proposed objective function. We get the solution by iteratively optimizing $\mathbf{U}_q$ and $\mathbf{W}_q$. The optimization process contains two steps:

Step 1: Assume all subspace matrix $\mathbf{W}_q$ are fixed, optimize $\mathbf{U}_q$ in each co-clustering. With the projection matrix W_q fixed, the problem is similar to SGCC model. The solution for $\mathbf{U}_q$ is

$$\mathbf{U_q} = \begin{bmatrix} (\mathbf{D_r} - \mathbf{K_{qr}})^{-1/2}\mathbf{u} \\ (\mathbf{D_c} - \mathbf{K_{qc}})^{-1/2}\mathbf{v} \end{bmatrix} \tag{2.6}$$

$$\mathbf{A} = (\mathbf{D_r} - \mathbf{K_{qr}})^{-1/2}\mathbf{E}(\mathbf{D_c} - \mathbf{K_{qc}})^{-1/2} \tag{2.7}$$

Matrix $\mathbf{u}$ equals to the first c_q left eigenvectors of matrix $\mathbf{A}$, and matrix $\mathbf{v}$ equals to first c_q right eigenvectors of matrix $\mathbf{A}$. c_q is cluster number.

Step 2: Assume all $\mathbf{U}_q$ are fixed, optimize $\mathbf{W}_q$ for each co-clustering. Matrix $\mathbf{W}_q$ is optimized by applying gradient descent on the Stiefel manifold (9; 10) to satisfy the orthonormality constraints, $\mathbf{W}_q^T \mathbf{W}_q = \mathbf{I}$. For convenience of notation, we use f to denote objective function in Equation (5). The gradient of the objective function is projected onto the tangent space, $\Delta W_{Stiefel} = \frac{\partial f}{\partial W_q} - W_q(\frac{\partial f}{\partial W_q})^T W_q$. Thus, W_q can be updated in the direction of the tangent space as follows:

$$W_{new} = W_{old}\exp(\tau W_{old}^T \Delta W_{Stiefel}), \tag{2.8}$$

where exp means matrix exponential and τ is the step size. We apply a backtracking line search to find the step size according to Armijo rule (11) to assure improvement of the objective function at every iteration. Since the object function is a summation of two parts, the derivatives of the two parts can be computed separately. Denote first part of the object function as f_1, $f_1 = \sum_q \mathrm{tr}(\mathbf{U}_q^T \mathbf{L}_q \mathbf{U}_q)$. According to chain rule, the derivative of f_1 can be computed as $\frac{\partial f_1}{\partial \mathbf{W}_q} = \frac{\partial f_1}{\partial \mathbf{K}_q}\frac{\partial \mathbf{K}_q}{\partial \mathbf{W}_q}$. To compute the derivative $\frac{\partial f_1}{\partial \mathbf{K}_q}$, we first expand trace of matrix and then compute the derivative. Then, $\frac{\partial \mathrm{tr}(\mathbf{U}_q^T \mathbf{L}_q \mathbf{U}_q)}{\partial \mathbf{K}_q r,i,j}$ can be written as

$$\frac{\partial \mathrm{tr}(\mathbf{U}_q^T \mathbf{L}_q \mathbf{U}_q)}{\partial k_{qr,i,j}} = \sum_{s=1}^{n} u_{qr,i,s}^2 - \sum_{s=1}^{n} u_{qr,i,s} u_{qr,j,s}, \tag{2.9}$$

According to equation (4), the HSIC term can be written as,

$$\mathrm{HSIC}(\mathbf{W_{q1}}^T v_r, \mathbf{W_{q2}}^T v_r) = (n-1)^{-2}\mathrm{tr}(\mathbf{K}_{q1}\mathbf{H}\mathbf{K}_{q2}\mathbf{H}) \tag{2.10}$$

where, $\mathbf{H} = [\mathbf{H}_{ij}]_{n\times n}, \mathbf{H}_{ij} = \delta_{ij} - 1/n$, δ_{ij} is the indicator function which takes 1 when $i = j$ and 0 otherwise. We use linear kernel function, which means $\mathbf{K}_q = \mathbf{X}^T \mathbf{W}_q \mathbf{W}_q^T \mathbf{X}$. Then derivative of $\mathbf{k}_{qr,ij}$ and derivative of $\mathrm{tr}(\mathbf{K}_{q1}\mathbf{H}\mathbf{K}_{q2}\mathbf{H})$ with respect to $\mathbf{W}_q$ can be represented as,

$$\frac{\partial \mathbf{k}_{qr,ij}}{\partial \mathbf{W}_q} = 2 \triangle x_{ij} \triangle x_{ij}^T \mathbf{W}_q \tag{2.11}$$

$$\frac{\partial \mathrm{tr}(\mathbf{K}_{q1}\mathbf{H}\mathbf{K}_{q2}\mathbf{H})}{\partial \mathbf{W}_q} = 2\mathbf{X}\mathbf{H}\mathbf{K}_{q2}\mathbf{H}\mathbf{X}^T \mathbf{W}_q \tag{2.12}$$

Therefore, combine equations (9), (11), and (12), the derivative of objective function f with respect to $\mathbf{W}_q$ can be written as

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{W}_q} &= \sum_q \frac{\partial tr(\mathbf{U}_q^T \mathbf{L}_q \mathbf{U}_q)}{\partial \mathbf{K}_q} * \frac{\partial \mathbf{K}_q}{\partial \mathbf{W}_q} \\ &+ \lambda \sum_{q \neq r} (n-1)^{-2} \frac{\partial \mathrm{tr}(\mathbf{K}_{q1}\mathbf{H}\mathbf{K}_{q2}\mathbf{H})}{\partial \mathbf{W}_q} \end{aligned} \tag{2.13}$$

Finally, $\Delta W_{Stiefel}$ can be calculated with Equation (13), and $\mathbf{W}_q$ is optimized. The algorithm is summarized in Algorithm 1.

2.4 Experiment

2.4.1 Dataset

The proposed algorithms were tested with two real world social network datasets, Foursquare dataset and Yelp dataset. Each dataset contains two objects, users and businesses (places), and the relationship information between them, check-ins (reviews).

Algorithm 1 Goal-oriented co-clusterings

Input: Data v_r for place vertex, v_c for user vertex, cluster number c_q, checkin matrix E, and number of views m

1: //Initialization step
Initialize Wq by clustering the features

2: **repeat**

3: For each co-clustering q, project data on subspaces $W_q, q = 1, ..., m$. Calculate the kernel similarity matrix K_q. Calculate the top c_q left eigenvectors of $\mathbf{D}_r^{-1/2}\mathbf{E}\mathbf{D}_c^{-1/2}$ as $\mathbf{u}$ and top c_q right eigenvectors of $\mathbf{D}_r^{-1/2}\mathbf{E}\mathbf{D}_c^{-1/2}$ as $\mathbf{v}$. Follow previous definition to compute matrix $\mathbf{U}_q$. Normalize rows of $\mathbf{U}_q$ to have unit length

4: Given all $\mathbf{U}_q$, update $\mathbf{W}_q$ based on gradient descent on the Stiefel manifold. Until τ_j satisfies Armijo condition $f(\mathbf{x}_k + \alpha\mathbf{p}_k) \leq f(\mathbf{x}_k) + c_1\alpha_k\mathbf{p}_k^T\nabla f(\mathbf{x}_k)$
update $W_{new} = W_{old}exp(\tau W_{old}^T \triangle W_{stiefel})$, where $\triangle W_{stiefel} = \frac{\partial f}{\partial W_q} - W_q(\frac{\partial f}{\partial W_q})^T W_q$

5: **until** Convergence

Foursquare dataset: The Foursquare dataset contains 780 places, 881 users, and $10,285$ check-ins. Check-in information included in this dataset is obtained from a Foursquare dataset provided by Cheng et al. (12). This Foursquare dataset itself does not contain any place information. It only provides web addresses of places in Foursquare. Foursquare ID of places can be extracted from web address. We also obtained place information through Foursquare API by place's Foursquare ID. For each place, we crawled its name, coordinate (latitude and longitude), postalcode, city, state, category, country, number of check-ins, number of users, and number of tips through Foursquare API. To get user information from Foursquare, we mapped user's Twitter account back to their Foursquare account, since Foursquare provides a service that allows users to link their Twitter accounts with Foursquare accounts. We crawled Foursquare users' home city information and number of tips.

Note that Foursquare has its own category hierarchy. The hierarchy contains two levels of categories. There are 9 categories in the top level and each top level category has a number of second level categories. For example, category *Arts & Entertainment* is a top level category, it has second level categories such as *Aquarium*, *Art Gallery*, and *Casino*. In Foursquare, each place can have multiple categories, for example, Willis Tower has categories *Building*, *Event Space*, and *Historic Site*. And these category can belong to different top-level categories. In the Willis Tower example, category *Building* belongs to category *Professional & other Places*, and category *Historic Site* belongs to category *Arts & Entertainment*. It's hard to assign a single top-level category to each place. Therefore, ground truth of category is not available. Ground truth of location is not available either, since different granularity (city, state, and country) of location can result in different ground truth.

Yelp dataset: The Yelp dataset consists of 5000 businesses 5000 users and 150,328 reviews. This data is sampled from Yelp Dataset Challenge[1]. Although Yelp dataset did provide check-in information, they did not specify which user checked in which place. Therefore, we utilized review data, since each review record contains user information and business information. For each business, we utilized its name, postalcode, city, state, location, category, number of reviews, number of stars. Similar with Foursquare data, ground truth for business and user clusters is not readily available.

---

[1]Yelp dataset can be found at http://www.yelp.com/dataset_challenge

Six approaches are applied to present the experiment results on two data sets. The comparison models include three state-of-the-art approaches: the information-theoretic co-clustering (14), euclidean co-clustering, and minimum squared residue co-clustering.

The common way of evaluating clustering results is using ground truth to compute cluster purity or normalized mutual information. Since there is no such suitable ground truth available in this paper, we proposed two indirect ways to evaluate the proposed models.

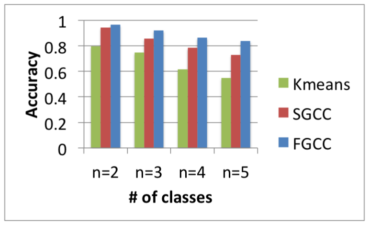
2.4.2 Evaluation of SGCC and FGCC Models

In order to evaluate the overall quality of the proposed algorithms, two metrics are selected to quantify the results. The first metric is classification based. It is an indication of the matching degree between clusters and goals. This tells us the performance of the algorithms toward multiple goals. The second one is based on KL divergence. It measures the divergence of different clusters toward a single goal. This evaluates the fundamental clustering quality. Both evaluation methods are presented in the following.

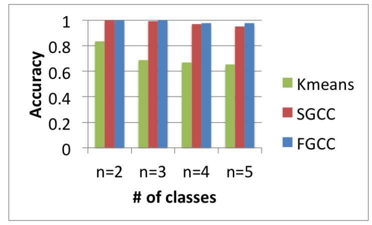
2.4.2.1 Classification based evaluation

In this section, we use classification-based method (13) to evaluate the clustering performance of SGCC and FGCC models. As mentioned early, users only define goals for places. Thus, only the quality of place clusters is evaluated versus the known goals. The idea of this evaluation method is to test whether utilizing clustering results of the proposed models could improve results of classification.

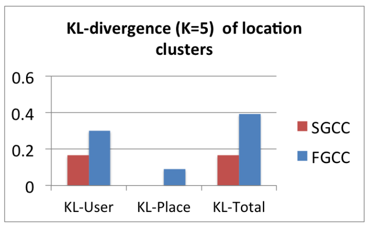
The labeled classes will be used as standard to measure the matching degree between clusters and goals. If the proposed models successfully accomplished the purpose of “Goal Orientation”,



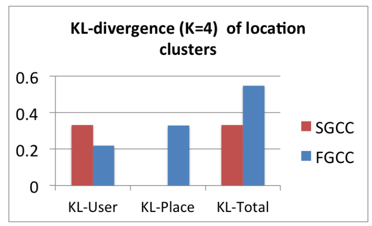
(a) location in Foursquare



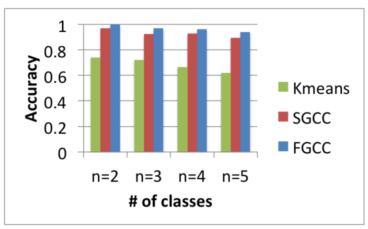
(b) location in Yelp



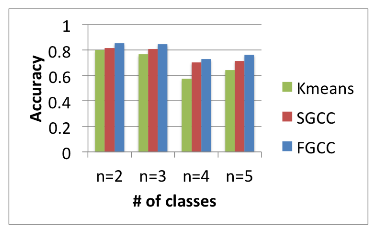
(c) co-clustering with regards to location in Foursquare



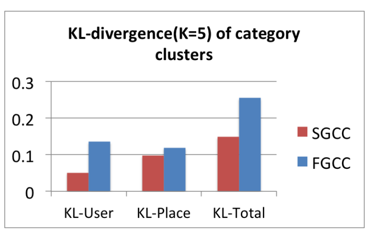
(d) co-clustering with regards to location in Yelp



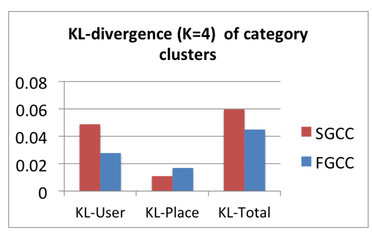
(e) category in Foursquare



(f) category in Yelp



(g) co-clustering with regards to category in Foursquare



(h) co-clustering with regards to category in Yelp

Figure 2. User Place Clusters

the co-clustering results generated by the proposed models would have a better classification performance compared with non-goal oriented clustering results. Two comparisons will be made to justify the proposed methods. First, we compared the SGCC and FGCC clustering results with a baseline clustering results without goal-oriented scheme. Specifically, K-mean clusters are selected as the baseline results. The other comparison is made between SGCC and FGCC. Since FGCC incorporates the subspace learning methods, it is expected that FGCC will produce results that have even higher relevancy to the goals than SGCC.

In this evaluation, we applied the decision tree to build classification models and conduct 10-fold cross validation to evaluate the accuracy. n is number of class. In Figures 2(a) and 2(b), the class labels are produced from place's location information (city); similarly, in Figures 2(e) and 2(f), the class labels are generated from place's category information. All four figures show that the proposed FGCC and SGCC models achieve significant improvement over K-means. It proves that when goal-related features are considered in the proposed models, it could fulfill user's expectation. Also, in all four figures, the FGCC model outperforms the SGCC model, since it incorporated subspace learning technique to use information discriminatively. We can draw the conclusion from these four figures that the proposed SGCC and FGCC models achieve higher quality co-clusterings with respect to location and category.

2.4.2.2 KL divergence

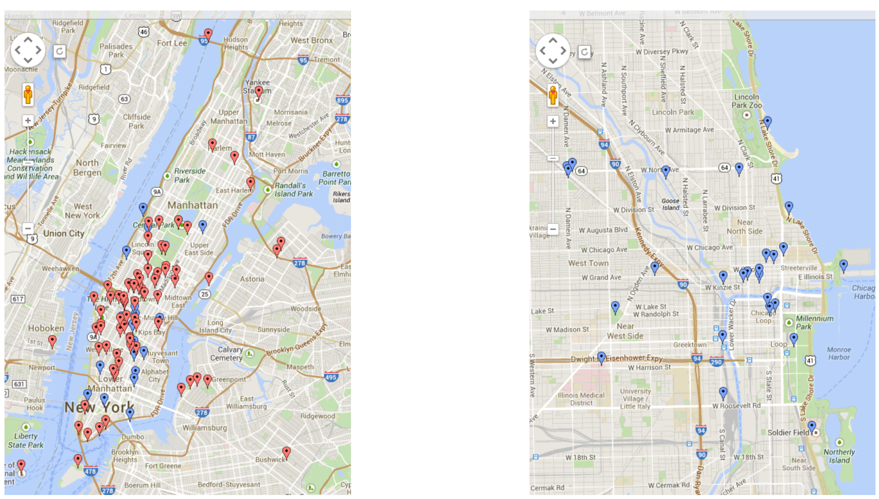
In this section, we evaluate the quality of co-clusterings with KL divergence (13).

Figures 2(c), 2(d), 2(g), and 2(h) show the KL divergence of SGCC and FGCC models. In Figures 2(c) and 2(d), KL divergence values of place clusters of SGCC are 0. For the Foursquare

Figure 3. Word cloud of **office**

dataset, in co-clusterings with regards to both location and category, FGCC model achieves higher KL divergence value. For the Yelp dataset, in co-clustering with regards to location, FGCC achieves higher KL divergence in total. In co-clustering with regards to category, FGCC still achieves higher KL divergence on place, the primary objective of the goal. However, SGCC achieves slightly higher KL divergence in total measure because it gives slightly better clustering results for user. Possible reason might be that the Yelp dataset has a weaker relationship between users and place location and category compared with Foursquare data since Yelp users will not write reviews for every places they have visited. In general, FGCC obtains higher quality clustering results in almost all cases.

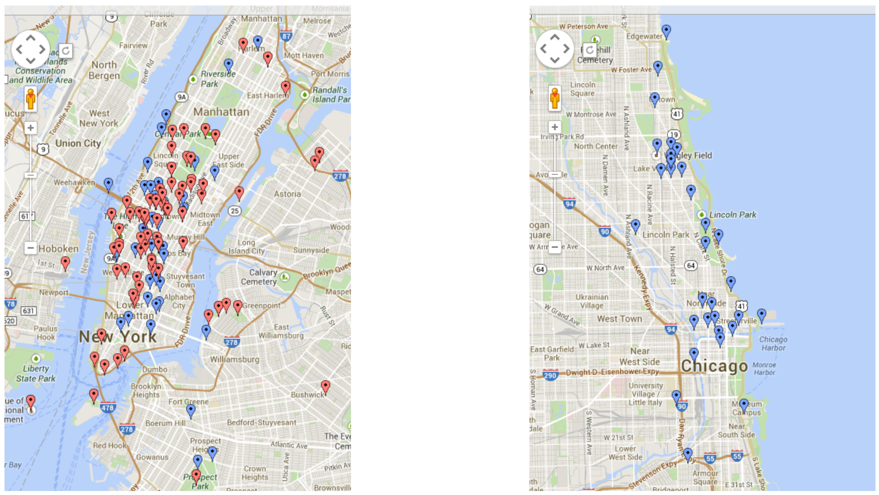
Figure 4. Word cloud of **fitness**



(a) NYC

(b) Chicago

Figure 5. User Clusters With Regards to Location



(a) NYC (b) Chicago

Figure 6. User Recommendation

2.4.3 Case Study

In this section, we use two case studies to show how user and place clusters produced from FGCC model can match goals. We use another case study is to show the possibility of utilizing co-clustering results to do social recommendations.

2.4.3.1 Place clusters

In this case study, we want to intuitively show how place clusters produced from FGCC model can meet category goal-oriented clustering. To help the reader get an idea of what kind of clustering results it outputs, two cluster results are given in this section. To show whether places in the same cluster have similar category, we use word cloud of places' names to represent place cluster. Usually, the name of the place can indicate the category of this place. If the words appears most frequently in one cluster are all indicating places with similar category, then we have confidence that places in this cluster have similar category. Two word clouds of these clusters are shown in Figure 3 and 4. It can be seen in Figure 3, most places are about firm or corporate offices, because words such as Building, HQ, Group, Corporate, office, center are the dominant keywords in this cluster. The second place cluster showed in Figure 4 are all related with fitness. Because words such as Fitness, Club, Sports, Equinox appeared most frequently. It is important to notice that different city names appeared in the same word cloud. It means that co-clustering with regards to category produces place clusters containing places with similar category but in different locations. It can be very helpful in actual recommendation when a user is traveling to a new place.

2.4.3.2 User Clusters

As a co-clustering algorithm, FGCC model not only produce place clusters with regards to goals, it also create user clusters with regards to clustering goals. In this case study, we want to show that how users could be clustered with regards to location. Figure 5 shows two user clusters (red and blue groups). We pick 10 users from each cluster and map these 20 users' check-in places into one map with clusters labels. Red group users' check-ins are all located in New York City. Most of Blue group users' checkins are located in Chicago. The figure shows that different clusters of users checked in at different locations. These checkin's city information matches the city information that user provided in their twitter account. Moreover, clustering user according to locations could facilitate the purpose of delivering targeted advertisements.

2.4.3.3 Recommendation for a Single User

Figure 6 shows an example that the proposed algorithms could be used to recommend locations for user. We pick one user, noted as **A**, from Foursquare Dataset. He resides in New York and most of his check-ins are in New York city. Then we pick the user clusters containing user **A** from two co-clusterings, location based (red group) and category based (blue group). Specifically, the red check-ins are posted by other users in the location cluster, while the blue check-ins are posted by users in the category cluster. One recommendation scenario is when user **A** goes out of his resident city to a new place such as Chicago, we can recommend places checked in by other users in user **A**'s category cluster. The reason is that users in user **A**'s category cluster have similar taste of places with user **A** and they lived in Chicago. Therefore, places checked in by these users in Chicago are more likely to attract user **A**.

2.5 Related Work

Co-clustering algorithm earns a lot of attentions from research communities in data mining and machine learning due to its ability of clustering two types of objects simultaneously (14; 15; 16; 17; 18). The co-clustering algorithm also proves itself as a powerful data mining techinique on practical applications such as bioinformatics (19; 20), natural language processing (21; 22), text mining (14; 23; 24). Dhillon et al. (14) utilizes graph partitioning technique for the co-clustering of the bipartite graph of documents and words. Graph partitioning technique attracts a lot of attentions, since the intuitive objective function and elegantly approximate problem as eigenvector problem. Therefore, a lot of researchers develop new algorithms based on spectral graph partitioning (8). Shi et al. (8) integrate prior information as constraints to spectral co-clustering framework. Only considering the relation between rows and columns may not be enough, people start to think about incorporating side information of rows and columns (13), and indeed achieves certain improvement. Traditional clustering algorithm identifies a single clustering. With complex data, researchers start to design multiple clustering algorithms which create multiple interpretations of the data. Niu et al. (25) design an algorithm which aims at finding multiple non-redundant clusterings.

Caruana et al. (26) devise an approach, which did consider the user's needs. However, the user is only allowed to select produced clusterings of the data. This will cause a problem when there is no suitable clusterings generated from algorithm for user to select. Also, this paper focused on clustering one type of objects. It is not able to handle two types of closely related objects. In this paper, we propose to utilize user provided data–goal-oriented features to create co-clusterings. Considering user provided information before running the algorithms could provide guidance for co-clusterings. Another paper (27) also utilizes user provided information, however their main focus is how to choose information in relational tables to help clustering on one type of objects.

CHAPTER 3

COLLABORATIVE CO-CLUSTERING ACROSS MULTIPLE SOCIAL MEDIA

(This chapter was previously published as "Collaborative Co-clustering across Multiple Social Media", in *IEEE International Conference on Mobile Data Management (MDM '16)* (2). DOI: `https://doi.org/10.1109/MDM.2016.31`.)

3.1 Introduction

It becomes prevalent that multiple types of objects co-exist in social networks. Social media websites such as Twitter, Foursquare naturally contain at least two types of objects. One is social media objects, for example, tweet, review, video, point of interest (POI), while the other is users. The user objects are unique compared to other social media objects as they conduct a series of actions in social media websites. Specifically, they create text, image, video in social media websites; they interact with each other by following or liking; they also interact with social media objects, for example, commenting on posts, retweeting tweets, checking in POIs, and sharing videos and photos. Co-clustering model groups two types of closely related objects at the same time. Utilizing co-clustering model to analyze social media data has several advantages. The co-clustering model is able to preserve meaningful relationships between social media objects and users. It also enhances recommendation which is the most important task

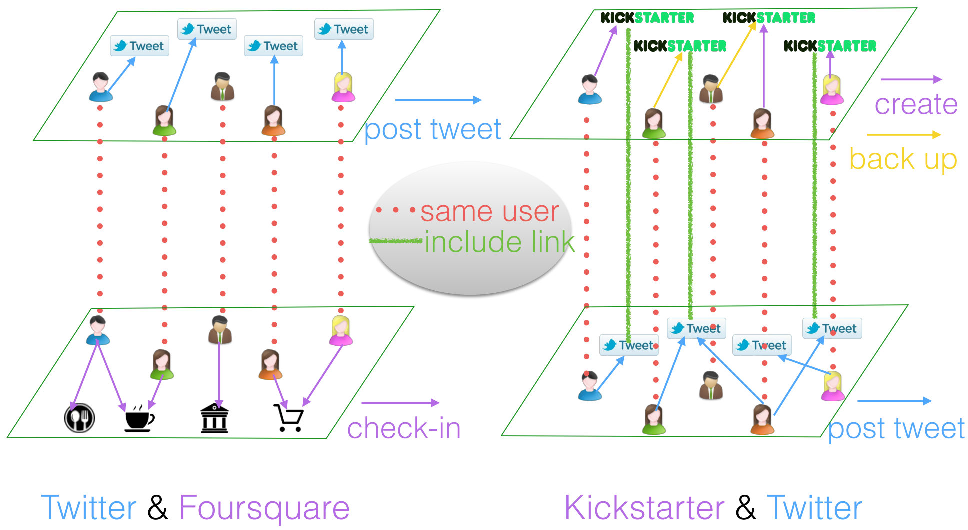


Figure 7. Two examples of Connected social networks

in social media services by producing groups of social media objects and groups of users with similar preference.

Linking different accounts together has become a routine in social networks. In Figure 7, we show two examples of connected social networks. The left example shows that Twitter network and Foursquare network are connected as users link their Twitter accounts with Foursquare accounts. Therefore, users can check in at venues in Foursquare and also post tweets in Twitter for similar events. The right example illustrates the linked Twitter network and Kickstarter network. Similarly to the left example, Twitter network and Kickstarter network are connected by users. Interestingly, they are also connected by projects as tweets can include links of Kickstarter projects. In this case, the co-clustering objects (users and projects) in Kickstarter also co-exist in Twitter. Utilizing these shared common objects and the corresponding attributes associated, one could gain more insight into a co-clustering problem in any of the connected networks. For example, users' profile, tweets, following users, and followers in Twitter could provide complementary information for a co-clustering task on Foursquare. Users who follow twitter account of publishers are more likely to have different tastes of places compared with users who follow restaurants.

Generally, information collected in real world is incomplete in each single source. Combining information from multi-sources, especially for social networks, could potentially allow us to construct a panorama for social media objects. Information from different sources could be both complementary and conflicted from different perspectives. It is thus interesting to design a smart way to unify and leverage multi-source information.

In this paper, we aim to co-cluster two types of objects in one social network by leveraging information from multiple social networks. Traditional co-clustering algorithms only works on relationship matrix which can not utilize additional information of each type of objects. More recent semi-supervised co-clustering algorithm considers side information as hard constraints to force co-clusters agree with these constraints. This could introduce more noise since constraints may not always be correct. Some other co-clustering algorithms consider information such as object features indiscriminately. They implicitly make an assumption that these features are equally useful to the co-clustering, which may not be the case when we considering information from multiple sources. Therefore, to avoid introducing too much noise and jeopardize the co-clustering performance, a technique needs to be developed to treat information from different sources discriminatively.

In order to handle multiple source information, a special multi-view formulation is proposed in this paper. Unlike traditional multi-view learning which only considers one type of objects, our problem is to handle more than one types of objects on multiple social networks. In the traditional multi-view learning, different views are constructed with regards to the same type of objects since there is only one type of objects. However, for the proposed problem, different views may contain information in terms of different types of objects. Moreover, one view may contain information in terms of more than one type of objects (e.g. relationship matrix). To sum up, traditional multi-view learning only handles one dimension of variation, namely different views, while in our problem, we need to handle two dimensions of variations, namely views and objects. It is a challenge to design a framework that handles two dimensions of variations.

In the proposed multi-view formulation, the relationship matrices, which are usually exploited in co-clustering, are utilized to construct the *relationship view* that contains information with regards to two types of objects. Features of each individual objects from different sources are utilized to build *feature views*, where each view contains information in terms of one type of object. Finally, a collaborative co-regularization co-clustering (**Co-CoClust**) framework is proposed. Co-regularization technique is introduced to learn co-clusters from both the *relationship view* and *feature views*. Specifically, "co-clusters" from spectral co-clustering and "clusters" from spectral clustering are co-regularized using eigenvector matrix in terms of the same type of objects. Iterative optimization is applied to iteratively update co-cluster.

The rest of the paper is organized as follows: Section 2 gives the details of the problem formulation and the proposed algorithms. Section 3 shows the experimental results as well as the discussion and comments. Section 4 presents a review for state-of-the-art research status. Section 5 concludes the paper.

3.2 Co-regularized Collaborative Co-clustering

In this section, we first briefly review spectral co-clustering algorithm. Then we introduce the proposed collaborative co-clustering formulation and the co-regularized collaborative co-clustering model.

3.2.1 Spectral Co-clustering

Spectral co-clustering is a co-clustering algorithm that transforms co-clustering problem into a partition problem on a bipartite graph. "Row vertices" and "column vertices" in the context of bipartite graph refer to original rows and columns of relationship matrix in co-clustering problem. Each edge in the bipartite graph corresponds to an element of relationship matrix in co-clustering problem. Denote the bipartite graph as $G = (\mathcal{V}_r, \mathcal{V}_c, \mathbf{E})$. It contains two sets of vertices $\mathcal{V}_r$ and $\mathcal{V}_c$. Each element of matrix $\mathbf{E}$ represents an edge between two vertices. The partition problem on a bipartite graph aims to simultaneously partition row vertices $\mathcal{V}_r$ into k clusters and column vertices $\mathcal{V}_c$ into k groups. Spectral co-clustering is designed to find minimum cut vertex partitions in a bipartite graph between row vertices and columns vertices.

The solution of this graph partitioning problem can be solved by calculating eigenvectors of the matrix $\mathbf{D}_r^{-1/2}\mathbf{E}\mathbf{D}_c^{-1/2}$, $[\mathbf{D}_r]_{ii} = \sum_j \mathbf{E}_{ij}$, and $[\mathbf{D}_c]_{ii} = \sum_j \mathbf{E}_{ji}$.

3.2.2 Collaborative Co-clustering Formulation

To co-cluster two types of objects on multiple source information, we formulate it in a multi-view fashion. *Relationship view* is constructed from relationship matrix of two types of objects. *Feature views* are constructed from features of each individual type of objects from different sources. In this work, we assume there is only one relationship view and there are multiple feature views in the problem formulation. In case of multiple relationship matrices scenario, only the relationship matrix in the target social network is used to construct the relationship view, while other relationship matrices in supporting social networks are transformed into two feature matrices with regards to two types of objects separately. For example, in the right example of Figure 7, there are two relationship matrices between users and projects in Kickstarter and Twitter networks, respectively. When solving a co-clustering problem in Kickstarter, we will only consider the relationship matrix from Kickstarter and transfer the other one into two feature matrices. We follow the notations of spectral co-clustering. Assume there are m object A and n object B, the relationship matrix between object A and object B is $\mathbf{E}$. Feature matrices in *feature views* are computed using Gaussian kernel. We denote kernel matrix of object A in view w as $\mathbf{K}_A^{(w)}$. We also denote kernel matrix of object B in view p as $\mathbf{K}_B^{(p)}$. Our goal is to find co-cluster result $\mathbf{X}$, where $\mathbf{X}$ can be solved using $\mathbf{E}$, $\mathbf{K}_A^{(w)}$, and $\mathbf{K}_B^{(p)}$.

3.2.3 Co-regularized Collaborative Co-clustering Model

In the proposed multi-view setting, objects are grouped into co-cluster and clusters under different views. In order to achieve better co-cluster by leveraging *relationship view* and *feature views*, we can consider this problem as finding maximum agreement between different views with regards to the same types of objects. The dissimilarity between co-cluster and clusters is then measured by a co-regularization term, which is performed on the eigenvector matrices of co-cluster and clusters. The reasons why eigenvector matrices are utilized are listed below. First, Eigenvector matrices in spectral clustering/co-clustering represent the graph partition rules, which is essentially the discriminative information of clusters. Second, an eigenvector matrix for two types of objects in spectral co-clustering could be decomposed into two eigenvector matrices corresponding to each type of objects. This would allow us to construct the co-regularization term for each type of objects under different views.

Frobenius norm is employed to realize this co-regularization. Assume two eigenvector matrices are $\mathbf{U}^{(a)}$ and $\mathbf{U}^{(b)}$, the Frobenius norm measures the distance between them, noted as $D(\mathbf{U}^{(a)}, \mathbf{U}^{(b)})$, where

$$D(\mathbf{U}^{(a)}, \mathbf{U}^{(b)}) = -\mathrm{tr}(\mathbf{U}^{(a)}\mathbf{U}^{(a)T}\mathbf{U}^{(b)}\mathbf{U}^{(b)T})$$

Subsequently, maximizing agreement between two views is to minimize $-\mathrm{tr}(\mathbf{U}^{(a)}\mathbf{U}^{(a)T}\mathbf{U}^{(b)}\mathbf{U}^{(b)T})$.

We formulate the collaborative co-clustering problem as an optimization problem which tries to find optimized graph partitioning for spectral co-clustering and spectral clustering in multiple views and maximize agreement between *relationship view* and *feature views*. That being said, we will first perform spectral co-clustering on the relationship view, from which we can obtain a partition result represented as $\mathbf{U}^{(v)}$. Also, spectral clustering will be performed on each *feature view* for A and B, where $\mathbf{U}_A^{(w)}\ for\ 1 \leq w \leq W$, and $\mathbf{U}_B^{(p)}\ for\ 1 \leq p \leq P$ can be obtained. Then, $\mathbf{U}^{(v)}$, $\mathbf{U}_A^{(w)}$, and $\mathbf{U}_B^{(p)}$ will be used as initial value for the optimization algorithm. The objective function for optimization is shown in Equation Equation 3.1. It is composed of three parts, i.e., the objective function of co-clustering in *relationship view*, the objective function in *feature views*, and the objective function for the co-regularization. In particular, the first item is the spectral co-clustering objective function. The second and third items stand for the objective function under features views for A and B, respectively. Note that multiple feature views exist for A and B, so they are both in a sum manner. As can be seen, the fourth and fifth items are in the shape of Frobenius norm, and therefore represent the objective function for co-regularization. It is worth to mention that since we combine the objective function together as shown, it brings us the merit of simultaneous clustering, co-clustering, and collaboratively optimization.

$$
\begin{aligned}
&\min_{\mathbf{U}^{(v)},\mathbf{U}_A^{(w)},\mathbf{U}_B^{(p)}} \mathrm{tr}(\mathbf{U}^{(v)T}\mathbf{L}^{(v)}\mathbf{U}^{(v)}) + \sum_{1\le w\le W} \mathrm{tr}(\mathbf{U}_A^{(w)T}\mathbf{L}_A^{(w)}\mathbf{U}_A^{(w)}) \\
&\quad + \sum_{1\le p\le P} \mathrm{tr}(\mathbf{U}_B^{(p)T}\mathbf{L}_B^{(p)}\mathbf{U}_B^{(p)}) \\
&\quad - \lambda \sum_{1\le w\le W} \mathrm{tr}(\mathbf{T}_1\mathbf{U}^{(v)}\mathbf{U}^{(v)T}\mathbf{T}_1{}^T\mathbf{U}_A^{(w)}\mathbf{U}_A^{(w)T}) \\
&\quad - \lambda \sum_{1\le p\le P} \mathrm{tr}(\mathbf{T}_2\mathbf{U}^{(v)}\mathbf{U}^{(v)T}\mathbf{T}_2{}^T\mathbf{U}_B^{(p)}\mathbf{U}_B^{(p)T}) \qquad (3.1) \\
&\text{where} \quad \mathbf{U}_A^{(w)T}\mathbf{U}_A^{(w)} = \mathbf{I}, \text{for } 1\le w\le W \\
&\quad \mathbf{U}_B^{(p)T}\mathbf{U}_B^{(p)} = \mathbf{I}, \text{for } 1\le p\le P \\
&\quad \mathbf{L}^{(v)} = \begin{bmatrix} \mathbf{D}_r^{(v)} & -\mathbf{E} \\ -\mathbf{E}^T & \mathbf{D}_c^{(v)} \end{bmatrix} \\
&\quad \mathbf{L}_A^{(w)} = \{\mathbf{D}_A^{(w)}\}^{-1/2}\mathbf{K}_A^{(w)}\{\mathbf{D}_A^{(w)}\}^{-1/2}, \text{for } 1\le w\le W \\
&\quad \mathbf{L}_B^{(p)} = \{\mathbf{D}_B^{(p)}\}^{-1/2}\mathbf{K}_B^{(p)}\{\mathbf{D}_B^{(p)}\}^{-1/2}, \text{for } 1\le p\le P
\end{aligned}
$$

$\mathbf{U}^{(v)}$ is eigenvector matrix in view v related to two types of objects A and B. $\mathbf{U}_A^{(w)}$ and $\mathbf{U}_B^{(p)}$ are eigenvector matrices in view w and view p related to object A and object B respectively. $\mathbf{L}^{(v)}$ is Laplacian matrix of co-clustering in view v. $\mathbf{L}_A^{(w)}$ and $\mathbf{L}_B^{(p)}$ are Laplacian matrices of clustering in view w and p. Matrices $\mathbf{D}_r^{(v)}$, $\mathbf{D}_c^{(v)}$, $\mathbf{D}_A^{(w)}$, and $\mathbf{D}_B^{(p)}$ are diagonal matrices, $[\mathbf{D}_r^{(v)}]_{ii} = \sum_j \mathbf{E}_{ij}$, $[\mathbf{D}_c^{(v)}]_{ii} = \sum_j \mathbf{E}_{ji}$, $[\mathbf{D}_A^{(w)}]_i = \sum_j [\mathbf{K}_A^{(w)}]_{ij}$, and $[\mathbf{D}_B^{(p)}]_i = \sum_j [\mathbf{K}_B^{(p)}]_{ij}$. In co-regularization, to make eigenvector matrices $\mathbf{U}^{(v)}$ and $\mathbf{U}_A^{(w)}/\mathbf{U}_B^{(p)}$ compatible with each other, we define transition matrix $\mathbf{T}_1 = \begin{bmatrix} \mathbf{I}_{m\times m} & \mathbf{0}_{m\times n} \end{bmatrix}$ and $\mathbf{T}_2 = \begin{bmatrix} \mathbf{0}_{n\times m} & \mathbf{I}_{n\times n} \end{bmatrix}$ to transfer $\mathbf{U}^{(v)}$ to another matrix which only contains information in terms of the same type of objects with $\mathbf{U}_A^{(w)}/\mathbf{U}_B^{(p)}$. Eigenvector matrix $\mathbf{U}^{(v)}$ can be split into two matrices $\mathbf{U}_A^{(v)}$ and $\mathbf{U}_B^{(v)}$ by equation $\mathbf{U}^{(v)} = \begin{bmatrix} \mathbf{U}_A^{(v)} \\ \\ \mathbf{U}_B^{(v)} \end{bmatrix}$, where matrix $\mathbf{U}_A^{(v)}$ and $\mathbf{U}_B^{(v)}$ are eigenvector matrices corresponding to object A and object B in view v respectively. Hyperparameter λ is defined to tradeoff original clusterings and co-regularization term.

3.2.4 Optimization

The proposed problem is a non-convex optimization problem since two non-convex terms for co-regularization are introduced in the objective function. We use alternating minimization technique to solve this problem, since alternating minimization provides a useful framework for iterative optimization in non-convex problems. In details, every eigenvector matrix will be updated alternatively with other eigenvector matrices being held fixed in each iteration. Since an analytical solution can be found for each eigenvector matrix during alternating minimization, repeating this process iteratively will converge asymptotically in general. However, it is not the aim of this paper to prove this property. We also provide an intuitive interpretation of the proposed algorithm. Take object A as an example, the final clusters of object A should preserve original relationship with another type of object B and also be refined by clusters in other views. To avoid a clustering result that is either too close to original co-cluster or too close to clusters in other views which may not preserve the full information, we utilize the iterative algorithm to balance co-clustering and feature view clustering. In such a fashion, we will be able to improve the results of co-clustering for two types of objects.

Next, we will explain how to alternatively update each eigenvector matrix, namely, $\mathbf{U}^{(v)}$, $\mathbf{U}_A^{(w)}$, and $\mathbf{U}_B^{(p)}$. Note that, there are multiple $\mathbf{U}_A^{(w)}$, for $1 \leq w \leq W$, each $\mathbf{U}_A^{(w)}$ will be updated in the same iteration with similar equation. Therefore, we will only need to derive an equation for a single item $\mathbf{U}_A^{(w)}$, and other items will follow accordingly. $\mathbf{U}_B^{(p)}$ is updated in the same manner. It can be seen in the objective function (Equation Equation 3.1), co-regularization takes place between $\mathbf{U}^{(v)}$ and $\mathbf{U}_A^{(w)}$, also between $\mathbf{U}^{(v)}$ and $\mathbf{U}_B^{(p)}$. To update $\mathbf{U}_A^{(w)}$ or $\mathbf{U}_B^{(p)}$, only the result of $\mathbf{U}^{(v)}$ is needed. However, when updating $\mathbf{U}^{(v)}$, both $\mathbf{U}_A^{(w)}$ and $\mathbf{U}_B^{(p)}$ are needed. Initially, $\mathbf{U}^{(v)}$ is computed from relationship matrix, while $\mathbf{U}_A^{(w)}$ or $\mathbf{U}_B^{(p)}$ is computed from feature matrix.

Given eigenvector matrix $\mathbf{U}^{(v)}$, to update $\mathbf{U}_A^{(w)}$ is to solve the following optimization problem with all unrelated items removed from Equation Equation 3.1,

$$\begin{aligned} &\min_{\mathbf{U}_A^{(w)}} \operatorname{tr}\{\mathbf{U}_A^{(w)T}(\mathbf{L}_A^{(w)} - \lambda \mathbf{T}_1 \mathbf{U}^{(v)} {\mathbf{U}^{(v)}}^T {\mathbf{T}_1}^T)\mathbf{U}_A^{(w)}\} \\ &\text{s.t.} \quad \mathbf{U}_A^{(w)T}\mathbf{U}_A^{(w)} = \mathbf{I} \end{aligned} \tag{3.2}$$

The above optimization problem is similar to the optimization problem in spectral clustering when we consider the whole term $\mathbf{L}_A^{(w)} - \lambda \mathbf{T}_1 \mathbf{U}^{(v)} {\mathbf{U}^{(v)}}^T {\mathbf{T}_1}^T$ as graph Laplacian matrix. Therefore, we denote ${\mathbf{L}_A^{(w)}}'$ as "new" graph Laplacian, where ${\mathbf{L}_A^{(w)}}' = \mathbf{L}_A^{(w)} - \lambda \mathbf{T}_1 \mathbf{U}^{(v)} \mathbf{U}^{(v)T} {\mathbf{T}_1}^T$. The solution of $\mathbf{U}_A^{(w)}$ is given by the top-k eigenvectors of "new" graph Laplacian ${\mathbf{L}_A^{(w)}}'$.

Given $\mathbf{U}^{(v)}$, updating $\mathbf{U}_B^{(p)}$ is similar to update $\mathbf{U}_A^{(w)}$.

$$\begin{aligned} &\min_{\mathbf{U}_B^{(p)}} \operatorname{tr}\{\mathbf{U}_B^{(p)T}(\mathbf{L}_B^{(p)} - \lambda \mathbf{T}_2 \mathbf{U}^{(v)} \mathbf{U}^{(v)T} {\mathbf{T}_2}^T)\mathbf{U}_B^{(p)}\} \\ &\text{s.t.} \quad \mathbf{U}_B^{(p)T}\mathbf{U}_B^{(p)} = \mathbf{I} \end{aligned} \tag{3.3}$$

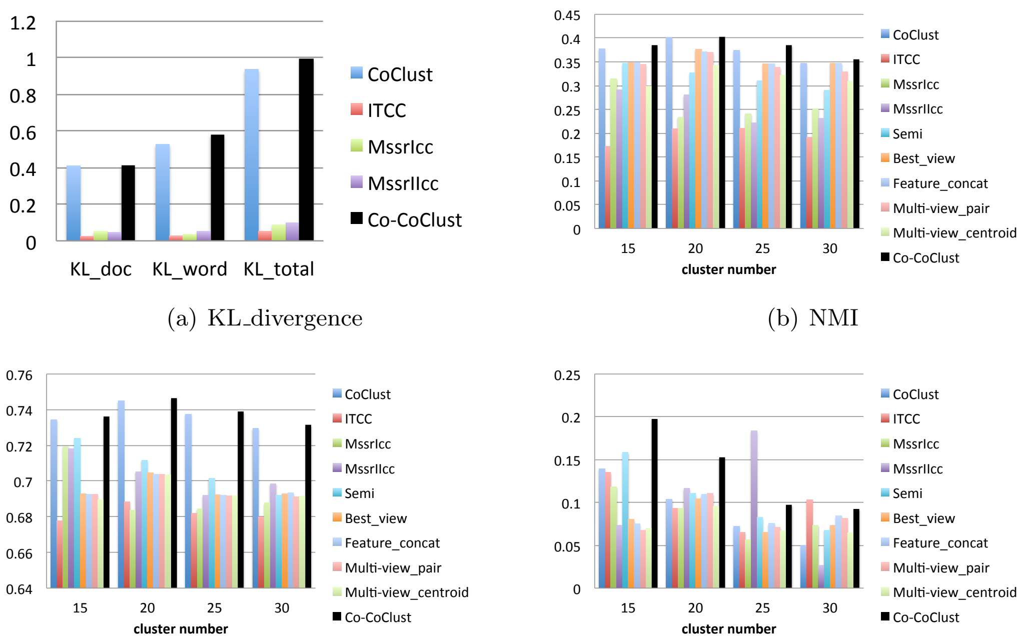
We denote a "new" graph Laplacian as $\mathbf{L}_B^{(p)'}$, where $\mathbf{L}_B^{(p)'} = \mathbf{L}_B^{(p)} - \lambda \mathbf{T}_2 \mathbf{U}^{(v)} \mathbf{U}^{(v)T} {\mathbf{T}_2}^T$. The solution of updated $\mathbf{U}_B^{(p)}$ is given by the top-k eigenvectors of "new" graph Laplacian $\mathbf{L}_B^{(p)'}$.

Given all eigenvector matrices $\mathbf{U}_A^{(w)}$, for $1 \leq w \leq W$ and $\mathbf{U}_B^{(p)}$, for $1 \leq p \leq P$, updating $\mathbf{U}^{(v)}$ is to solve the following optimization problem,

$$\begin{aligned} &\min_{\mathbf{U}^{(v)}} \operatorname{tr}\{\mathbf{U}^{(v)T}(\mathbf{L}^{(v)} - \lambda \sum_{1 \leq w \leq W} {\mathbf{T}_1}^T \mathbf{U}_A^{(w)} \mathbf{U}_A^{(w)T} \mathbf{T}_1 \\ &- \lambda \sum_{1 \leq p \leq P} {\mathbf{T}_2}^T \mathbf{U}_B^{(p)} \mathbf{U}_B^{(p)T} \mathbf{T}_2) \mathbf{U}^{(v)}\} \end{aligned} \tag{3.4}$$

Recall in spectral co-clustering, this can be solved by computing u and v as left and right eigenvectors of a matrix $\mathbf{F}$, denoted as

$$\begin{aligned} &\mathbf{F} = (\mathbf{D}_r - \lambda \sum_{1 \leq w \leq W} \mathbf{U}_A^{(w)} \mathbf{U}_A^{(w)T})^{-1/2} \mathbf{E} (\mathbf{D}_c - \\ &\lambda \sum_{1 \leq p \leq P} \mathbf{U}_B^{(p)} \mathbf{U}_B^{(p)T})^{-1/2} \end{aligned} \tag{3.5}$$



(a) KL_divergence (b) NMI

(c) RI (d) Accuracy

Figure 8. Reuters Dataset

Then the solution of matrix $\mathbf{U}^{(v)}$ is given as follows,

$$\mathbf{U}^{(v)} = \begin{bmatrix} (\mathbf{D}_r - \lambda \sum_{1 \le w \le W} \mathbf{U}_A^{(w)} \mathbf{U}_A^{(w)T})^{-1/2} u \\ \\ (\mathbf{D}_c - \lambda \sum_{1 \le p \le P} \mathbf{U}_B^{(p)} \mathbf{U}_B^{(p)T})^{-1/2} v \end{bmatrix} \tag{3.6}$$

For details of spectral co-clustering, please refer to (5). The proposed algorithm is summarized in Algorithm 2.

Algorithm 2 Multi-view Co-CoClust Algorithm

Input: Relationship matrix $\mathbf{E}$ in view v, feature matrix (or Kernel) $\mathbf{K}_A^{(w)}$ of object A, feature matrix (or Kernel) $\mathbf{K}_B^{(p)}$ of object B, iteration number $iter$, number of clusters k

Output: Row partition matrix $\mathbf{X}_A$; column partition matrix $\mathbf{X}_B$

//Initialization step

Initialize $\mathbf{L}^{(v)} = \begin{bmatrix} \mathbf{D}_r^{(v)} & -\mathbf{E} \\ -\mathbf{E}^T & \mathbf{D}_c^{(v)} \end{bmatrix}$ $\mathbf{L}_A^{(w)} = \{\mathbf{D}_A^{(w)}\}^{-1/2}\mathbf{K}_A^{(w)}\{\mathbf{D}_A^{(w)}\}^{-1/2}$, for $1 \le w \le W$

$\mathbf{L}_B^{(p)} = \{\mathbf{D}_B^{(p)}\}^{-1/2}\mathbf{K}_B^{(p)}\{\mathbf{D}_B^{(p)}\}^{-1/2}$, for $1 \le p \le P$

Compute $\mathbf{U}^{(v)}$ by spectral co-clustering with $\mathbf{E}$

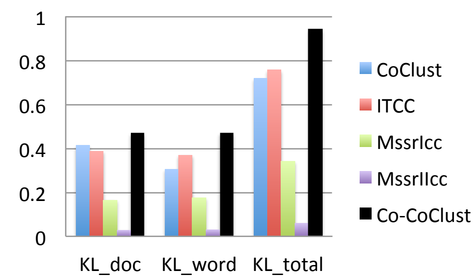
for $1 \le w \le W$ Compute $\mathbf{U}_A^{(w)}$ by spectral clustering with $\mathbf{K}_A^{(w)}$

for $1 \le p \le P$ Compute $\mathbf{U}_B^{(p)}$ by spectral clustering with $\mathbf{K}_B^{(p)}$

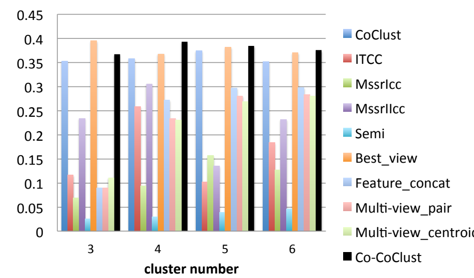
```
while i ≤ iter do
    Update U_A^(w)
    for w = 1 to W do
        compute top k eigenvectors of graph Laplacian L_A^(w)' (Equation 3.2)
    end for
    Update U_B^(p)
    for p = 1 to P do
        compute top k eigenvectors of graph Laplacian L_B^(p)' (Equation 3.3)
    end for
    Update U^(v)
    compute left and right eigenvectors of the 2nd to (k + 1)-th eigenvalues of matrix F
    (Equation (Equation 3.5))
    update U^(v) (Equation (Equation 3.6))
    i ← i + 1
end while
```

3.3 Experiment

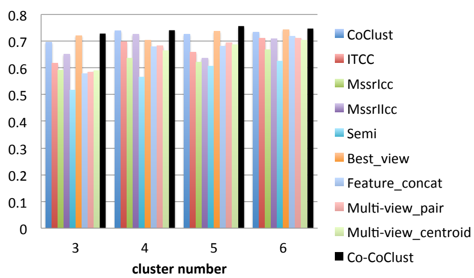
In this section, we first introduce three sets of datasets. Then, the proposed **Co-Coclust** algorithm is justified along with state-of-the-art algorithms: single view clustering algorithms, co-clustering algorithms, and multi-view clustering algorithms.



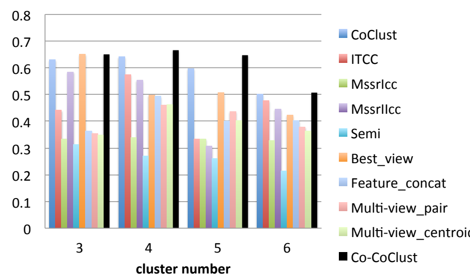
(a) KL_divergence



(b) NMI



(c) RI



(d) Accuracy

Figure 9. Cornell Dataset

The first two datasets contain labels, and therefore ground truth could be obtained for evaluation purpose. Different algorithms are compared with regards to quality of the co-clusters and clusters. The last dataset, Foursquare and Twitter dataset, does not have any ground truth. In order to show the performance of the proposed algorithm, the dataset was made partially observable on purpose. Comparison was made between the proposed algorithm and baseline algorithms using partial information against K-means clustering results using full information, which is assumed to be the ground truth.

3.3.1 Setup

3.3.1.1 Datasets

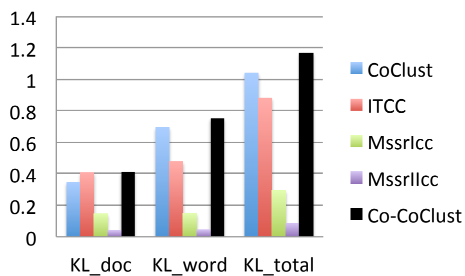
- **Reuters Multilingual dataset** This dataset contains 1200 documents with each document belonging to one of 6 categories with regards to topics. Those documents are written in 5 different languages, and therefore contain 5 views. Originally, the documents were written in English, and were translated into French, German, Italian, and Spanish. We can co-cluster documents and words in this example. For the multi-view setting, *relationship view* is constructed using document-word matrix from the documents in English, where each entry of the matrix indicates whether or not a certain word appears in a certain document. 4 *Feature views* of documents are constructed from 4 translated version of documents in French, German, Italian, and Spanish.

- **WebKB dataset (Cornell, Texas, Washington, Wisconsin)** This dataset contains 4 sub-datasets of web pages extracted from computer science department of 4 universities: Cornell, Texas, Washington, and Wisconsin. We can co-cluster documents (web pages) and words in this dataset. The web pages (documents) were manually classified over 5 categories (student, project, course, staff, faculty). Each sub-dataset contains four views of documents including document and word, inbound, outbound, and cite views. The document and word view is utilized to construct *relationship view*, while the three other views of document are treated as *feature views* for document. The construction method for *relationship view* is similar to that in **Reuters Multilingual** dataset.

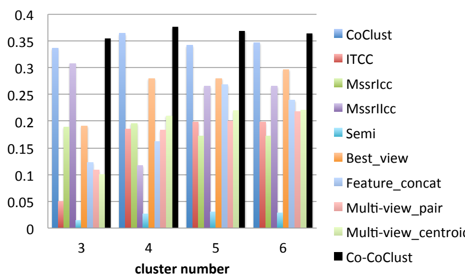
- **Foursquare+Twitter dataset** The dataset contains 881 Foursquare users who perform check-in $10,285$ ((28)) times at 780 venues. These 881 users all link their Foursquare account and Twitter account through users' Foursquare page, since Foursquare allows users to explicitly show their Twitter account on their profile page. Therefore, we are able to combine information from both networks to enhance the goal of co-clustering users and places in Foursquare network. In the multi-view setting, we construct relationship view based on check-ins in Foursquare network, since it contains relationship information between users and places. Three feature views are constructed with regards to the features of users or places. Two feature views are constructed from Foursquare, and the other from Twitter. In Foursquare, user information including names, homeCity, count of followers, and count of tips are collected as one view. Place information including names, coordinates, postal code, city, state, category, country, number of check-ins, number of users (visitors), and number of tips form the second view. The third feature view is constructed from Twitter network, where user information including location, count of friends, creation time of account, time zone, and count of tweets are utilized.

3.3.1.2 Evaluated Approaches

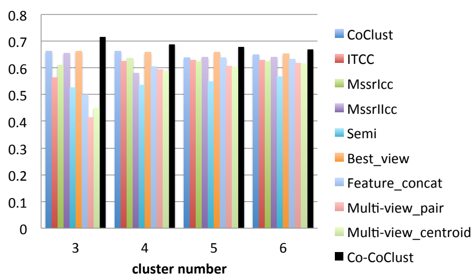
Benchmarks are nine state-of-the-art algorithms including co-clustering algorithms, spectral clustering algorithms, and multi-view clustering algorithms. All of the baselines are evaluated in every dataset. Essentially, the proposed framework is to solve a co-clustering problem, so we compare it with co-clustering algorithms first. The first baseline is spectral co-clustering algorithm (**CoClust**), which solves the co-clustering problem by spectral graph partitioning



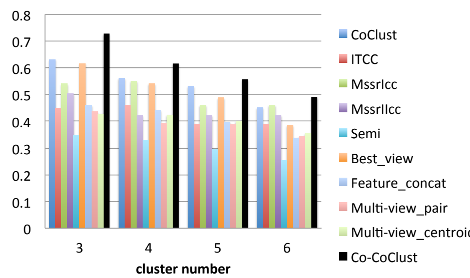
(a) KL_divergence



(b) NMI



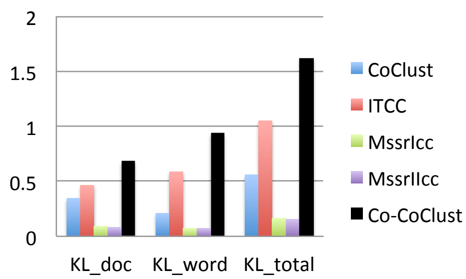
(c) RI



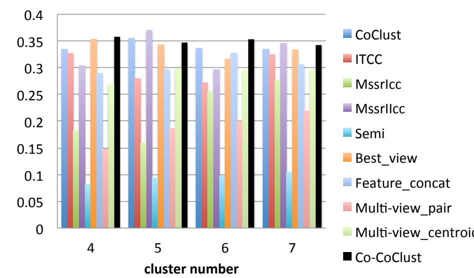
(d) Accuracy

Figure 10. Texas Dataset

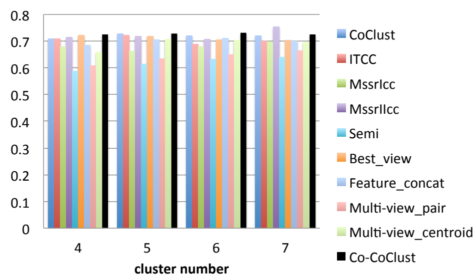
((5)). Several other typical co-clustering algorithms are also evaluated including information-theoretic co-clustering (**ITCC**), euclidean co-clustering algorithm (**MssrIcc**), and minimum squared residue co-clustering algorithm (**MssrIIcc**). Since additional information can be considered as constraints, a semi-supervised co-clustering algorithm is also included as baseline (**Semi**) ((8)). Recall that co-clustering results include the results of clustering and features regarding to a single type of objects are utilized in the proposed collaborative co-clustering framework, one would also be interested to see if the proposed algorithm performs better than clustering algorithms for one type of objects. Since there are multi-views for clustering, we



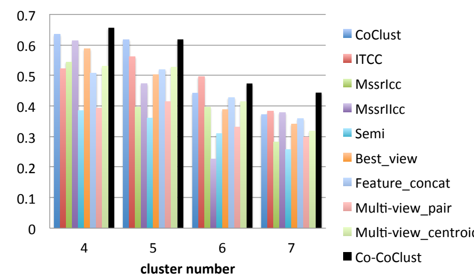
(a) KL_divergence



(b) NMI



(c) RI



(d) Accuracy

Figure 11. Washington Dataset

challenge the best results of spectral clustering among different views as well as the overall result using features from all views. We run spectral clustering algorithm on each single view and report the best results in terms of three evaluation metrics as **Best_view**. **Feature_concat** describes result of spectral clustering algorithm on feature concatenation of different views. Both of them are compared with our proposed algorithm. Two different schemes (**multi-view_pair** and **multi-view_central**) of multi-view clustering algorithm (29) are also tested on three datasets.

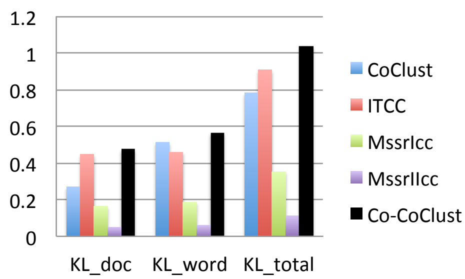
3.3.1.3 Performance Metrics

Evaluations are conducted in two aspects. In the first aspect, KL_divergence is utilized to evaluate the quality of co-clusters. Only co-clustering algorithms are compared in this part. In the second aspect, the proposed method is evaluated from the perspective of clustering quality of one type of objects. All of the baselines are compared. Three metrics, **N**ormalized **M**utual **I**nformation (**NMI**), **R**andom **I**ndex (**RI**), and **Accuracy** are utilized to test the quality of clustering results. In all of the following figures, baseline methods are represented with bright colors, while the proposed **Co-CoClust** is always represented with black color to differentiate from all baselines.

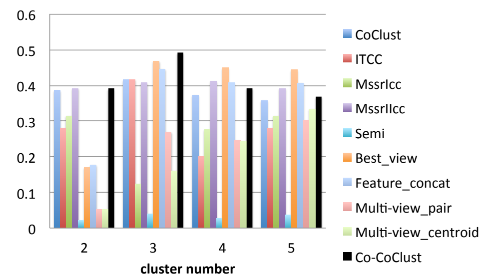
3.3.2 Experiment Results

3.3.2.1 Document-word Datasets

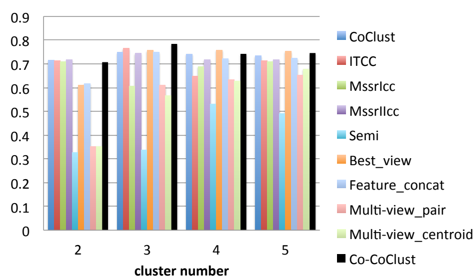
Figure 15 presents the evaluation results on Reuters dataset. Quality of co-clusters is evaluated by KL_divergence as shown in Figure 15(a). It can be seen that spectral clustering based algorithm generally obtain much better result than other co-clustering methods. Among the baselines, **CoClust** achieve the best result. However, the proposed **Co-CoClust** outperforms **CoClust** by 7%. Clustering results on documents are compared in Figure 15(b)-(d), where the x axis means different number of clusters in the results. Figure 15(b) shows that **Co-CoClust** has 20% improvement over the average of baseline results and 2% improvement over the best of the baselines. Figure 15(c) gives **RI** results, where **Co-CoClust** shows the best performance on all four cases. **Co-CoClust** achieves good overall **Accuracy**. Especially, when cluster number



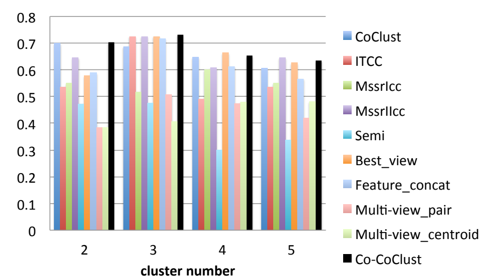
(a) KL_divergence



(b) NMI



(c) RI



(d) Accuracy

Figure 12. Wisconsin Dataset

equals to 15 and 20, **Co-CoClust** achieves 28% improvement over best of baselines. When number of clusters equals to 25, **Co-CoClust** obtains second best results.

Evaluation results on **Cornell** dataset are shown in Figure 9. Figure 9(a) shows KL_divergence of document and word clusters, where **Co-CoClust** achieves much better results compared with other baselines. Quality of document clusters is being evaluated by other metrics with varying number of clusters from 3 to 6, as shown in Figure 9(b)-(d). **Co-CoClust** achieves the best results on 3 cases and one second best result when number of clusters equals to 3 in Figure 9(b) . In Figure 9(c) and Figure 9(d), **Co-CoClust** achieves the best results on RI and Accuracy.

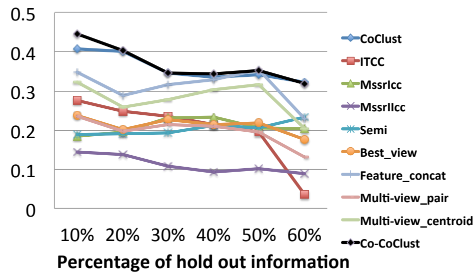
Figure 10 demonstrates evaluation results on **Texas** dataset. Again, the proposed approach performs well on KL_divergence metric in Figure 10(a). We can also observe from Figure 10(b), Figure 10(c), and Figure 10(d) that **Co-CoClust** consistently achieves better results. For instance, in Figure 10(d), **Co-CoClust** improves second best method by 10% averaging over different cluster number.

Evaluation results on **Washington** dataset is summarized in Figure 11. **Co-CoClust** achieves consistently better results than baselines by a significant margin on KL_divergence. In the evaluation of NMI, RI, and Accuracy, **Co-CoClust** performs better than co-clustering based methods and multi-view based methods in most of cases.

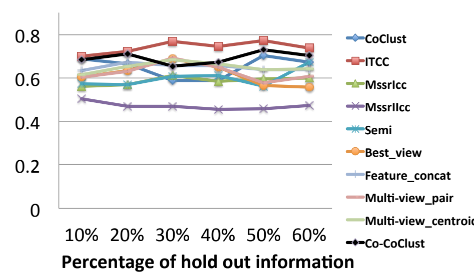
Figure 12 illustrates performance of the proposed approach and 9 baselines on **Wisconsin** dataset. **Co-CoClust** obtains the best results in KL_divergence in Figure 12(a). **Co-CoClust** also achieves better results on **NMI**, **RI**, and **Accuracy** overall compared with baseline algorithms.

3.3.2.2 Social Network Dataset

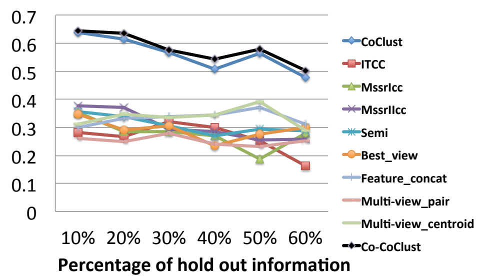
This dataset serves as a case study for a practical application in social network. Unlike **Reuters** and **webKB** datasets which have ground truth, social network dataset did not provide any ground truth for user clusters or clusters of social media objects. Moreover, it is hard to manually label users or social media objects with high quality. Therefore, in this paper, we employ a different evaluation strategy to show the performance of the proposed **Co-CoClust** on social network dataset. The idea is to justify the efficacy and robustness of the proposed approach in utilizing partially observed information for the task of clustering place. With an



(a) NMI



(b) RI



(c) Accuracy

Figure 13. Foursquare Dataset

increasing percentage of random information loss, we want to evaluate how the proposed algorithm performs to deal with the information loss by means of compensating the loss via multi-sources learning. Figure 13 shows comparison results when the percentage of information loss ranges from 10% to 60%. The "ground truth" is produced by k-means algorithm on full information in terms of place. Figure 13(a) depicts **NMI** of the proposed approach and baselines. In general, all of the other algorithms suffer more degradation when more information is hidden from experiment. However, **Co-CoClust** consistently performs the best in the perspectives of **NMI** and **Accuracy** as shown in Figure 13(a) and Figure 13(c), which demonstrates the robustness of **Co-CoClust**. **Co-CoClust** obtains the second best results when evaluated by **RI** as shown in Figure 13(b). Overall, Co-CoCluster outperforms other algorithms in combining multi-source information for co-clustering problems.

To sum up, the proposed **Co-CoClust** performs consistently better than single view clustering, co-clustering, and multi-view clustering algorithms on both social networks and document-word datasets. It proves that the proposed model steadily outperforms most of the state of the art algorithms in combining multiple source information for co-clustering problems.

3.4 Related Work

Co-clustering algorithm earns a lot of attentions from research communities in data mining and machine learning due to its capability to cluster two types of objects or object and feature simultaneously (5; 14; 8; 30). Co-clustering algorithms also prove to be a powerful data mining technique on practical applications such as text mining, social recommendation, mining networks. Spectral co-clustering algorithm proposed by Dhillon et al. (5) utilizes graph partitioning technique for the co-clustering of the bipartite graph of documents and words. It attracts a lot of attention since the objective function is well formulated and could be solved as an eigenvector problem. Other co-clustering algorithms are also proposed to embrace different techniques to simultaneously clustering two types of objects (14). Recently, researchers have developed many new co-clustering algorithms to add constraints (8; 31) or side information (32; 33; 34). (8) integrates additional information as constraints into a semi-supervised co-clustering algorithm. (31) proposes information theoretic co-clustering framework for text mining. In (32), the authors claim that using metadata as constraint in co-clustering achieves better performance than metadata-driven and metadata-injected methods. (35) works on co-clustering multiple domains of objects, achieving clusterings of multiple types of objects by linear combination. However, researches in developing a reasonable yet flexible method to handle additional information other than using them as constraints is limited.

The rise of multi-representation data creates an opportunity for multi-view learning. Many multi-view based clustering algorithms (36; 37; 38; 39; 40) have also been proposed. Non-negative matrix factorization technique is also exploited in multi-view setting (38; 40). (40) assumes that not all examples are presented in all views and proposes a non-negative matrix factorization based model for clustering under partially observed data. Multi-view clustering also sees its application in web 2.0 items (39). Similar to this paper which utilize social media objects in multi-view clustering, we did co-clustering on social media objects.

Recently, several algorithms in multi-view setting are proposed for spectral clustering (41; 29; 42; 43; 44; 45). (45) generalizes single view normalized cut approach to multiple views to obtain a graph cut by random walk based formulation. In (42), the authors focus on the two-view case of multi-view clustering by creating a bipartite graph. Spectral clustering is applied on the constructed bipartite graph. Instead of working on original features, (41) takes different clusterings coming from different sources as input and reconcile them to find a final clustering. It is suggested that they could achieve better performance by directly working on clusterings instead of original features of multiple source information. Another paper (29) also works on clusterings instead of original features. They employ the co-regularization technique to force clusterings learned from different views of the data to agree with each other. Working on clusterings instead of original features shows good performance on clustering one type of objects. Inspired by the success in clustering, in this paper, we are working on co-clustering two types of objects on multiple source information.

Other multi-view clustering algorithms (46; 47) also utilize co-regularization technique. (47) implements multi-view regularization of unlabeled examples to perform semi-supervised learning. (48) works on clustering multiple types of objects with their relationship matrices. Relationship matrices are utilized to compute co-similarity matrices. Then, different co-similarity matrices with respect to the same type of objects are combined for generating clustering result. There are two major differences between (48) and our work. First of all, they transfer co-clustering problem of multiple types of objects into clustering problems via co-similarity matrices. However, in this paper we proposed a direct co-clustering framework to simultaneously cluster multiple types of objects in the original space. Secondly, paper (48) implements the idea of combining multiple source information by combining multiple similarity matrices. It is not clear how this combining strategy discriminatively considers information from different sources. However, in this paper, different co-regularization terms are utilized to discriminatively handle multiple source information.

CHAPTER 4

COLLECTIVE GEOGRAPHICAL EMBEDDING FOR GEOLOCATING SOCIAL NETWORK USERS

(This chapter was previously published as "Collective Geographical Embedding for Geolocating Social Network Users", in *The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD '17)* (3). DOI: `https://doi.org/10.1007/978-3-319-57454-7_47`.)

4.1 Introduction

Urban computing has attracted many research attentions (49). Cross-domain data can be fused together to aid this task (50; 51). One of the core tasks towards these services is to infer the physical location of participants, as it not only advances the recognition of individual behavioural patterns but also facilitates the analysis of the crowd mobility and communication.

Intuitively, friendships between users provide a valuable hint since people tend to live close to their friends. As a partial observation of users' social relations, online social networks (OSNs) shed a light on the problem of geolocating individuals (52; 53). Another useful information is the online footprints shared in OSNs, which can be observed through the geotagged contents generated by users. Unfortunately, most of existing approaches only focus on one single data source, either the social network of the online friendships (54; 55) or the content of the online footprints (56; 57). There are several crucial challenges that hinder the performance of the existing methods: (1) **Data Sparsity**: Due to privacy concern, not many users choose to reveal

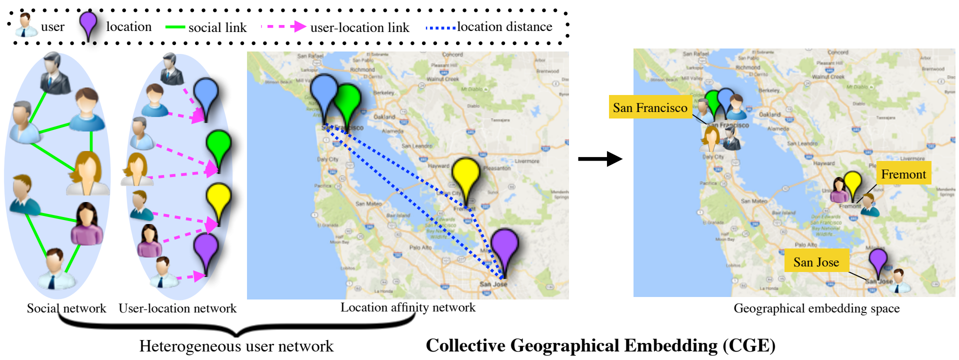


Figure 14. Example of learning the geographical embedding space from heterogeneous networks.

their location information. Research in Twitter suggest that only 16% of users registered city level locations in their profiles (58), and the percentage of tweets with geographical coordinates was merely 1% (59). (2) **Noisy Signals:** The signals retrieved from OSNs may not conform the assumption that the friends and footprints of a user will be close to the user's physical location. Reasons lead to noisy signals include global online friendships, frequent relocation, and posting geotagged contents during travel, etc. Such sparse and noisy data constitute a major challenge for label propagation based methods (54; 55) and probability estimation based methods (60).(3) **Scalability:** Since OSNs often contain millions nodes and links, how to handle such a large scale data poses another challenge. In particular, most methods that involve sophisticated NLP techniques (56) require a huge amount of computational resources and may not be applicable to large-scale datasets.

Recently, network embedding techniques (61; 62; 63) are introduced to embed network data into a low dimensional space while preserving the neighborhood closeness of the network data. Through embedding all objects into a common low dimensional space, it is possible to calculate the similarity between each pair of objects to mitigate the sparsity problem in network data. Although several studies (61; 63) have been proposed to model multiple networks concurrently, these methods do not differentiate each type of the objects involved. Furthermore, the embeddings learned by the existing methods do not have any physical meanings.

Since each tagged location is associated with a geographic coordinate (e.g., latitude and longitude), the distance between the embeddings of any pair of locations should be able to reflect the geographical distance. In this paper, we propose a Collective Geometrical Embedding (CGE) algorithm that can effectively infer the geolocation of social network users, by jointly learning the embeddings of users and check-ins with respect to the real-world geometrical space. In other words, the real geometrical distance between any pair of objects (i.e., users or locations) is resembled by euclidean distance of two vectors in the low dimensional space. Figure (Figure 14) illustrates the main concept of the geometrical embedding learning, where the left figure shows an example of a heterogeneous user network, the right figure depicts a snapshot of the geographical embedding space learned through the proposed algorithm. The heterogeneous user network shown includes a user network, a user-location network, and a location affinity network. By collectively embedding the heterogeneous network into a common subspace while preserving the geometrical distances between users and locations, the goal of inferring users' geolocations can be achieved without difficulty.

The main contributions of this paper can be summarized as follows:

1. We directly leverage multiple information sources by embedding a heterogeneous network, which alleviates the problem of sparse and noisy data.

2. We propose a collective geometrical embedding (CGE) method that integrates the geometrical regularization into the process of network embedding, which makes the learned embeddings preserving not only the neighborhood closeness of network data but also the geometrical closeness of locations. To the best of our knowledge, this work is the first to learn an embedding space that can reflect the real-world geolocation characteristics.

3. Through the extensive empirical studies on real-world datasets, we demonstrate that the proposed CGE method significantly outperforms other state-of-the-art algorithms in addressing the problem of geolocating individuals.

4.2 Preliminaries

In this section, we first introduce the definition of each source for the heterogeneous network and present the problem statement of this study.

Definition 1 *Social Network A social network can be represented by* $G_{uu} = (\mathcal{U}, \mathcal{E}_{uu})$*, where* $\mathcal{U} = \{u_1, u_2, ...u_N\}$ *denotes the set of users, and* $\mathcal{E}_{uu}$ *denotes the set of edges. Each* $e_{ij} \in \mathcal{E}_{uu}$ *is a social link between user* i *and user* j*.*

Next, we present the definition of user-location network, in which the frequency of visit was used to set the weight of edges between users and locations.

Definition 2 *User-Location Network A user-location network is represented by* $G_{up} = (\mathcal{U} \cup \mathcal{P}, \mathcal{E}_{up})$*, where* $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$ *denotes the set of users,* $\mathcal{P} = \{p_1, p_2, \ldots, p_M\}$ *denotes the* *set of locations, and the weight w_{ik} on the edge $e_{ik} \in \mathcal{E}_{up}$ is the number of times that the user u_i visited the location p_k.*

Definition 3 *Location Affinity Network A location affinity network can be represented by $G_{pp} = (\mathcal{P}, \mathcal{E}_{pp})$, where $\mathcal{P} = \{p_1, p_2, \ldots, p_M\}$ denotes the set of locations, and the weight w_{ij} on the edge $e_{ij} \in \mathcal{E}_{pp}$ indicates the location closeness between the locations p_i and p_j.*

Definition 4 *Heterogeneous User Network A heterogeneous user network can be represented by $G_u = G_{uu} \cup G_{up} \cup G_{pp}$, which consists of the social network G_{uu}, the user-location network G_{up} and the location affinity network G_{pp}. The same sets of users and locations are shared in G_u.*

Definition 5 *Geolocating Social Network Users Given a heterogeneous user network G_u, estimate a location $\hat{p}_{u_i}$ for each user u_i in $\mathcal{U}$ such that the estimated location $\hat{p}_{u_i}$ close to u_i's physical location p_{u_i}.*

4.3 Methodology

In this section, we introduce the proposed method that learns the geographical embeddings of users and locations through the heterogeneous user network w.r.t. the real-world geometrical space. Since the heterogeneous user network consists of multiple bipartite networks, we first present how to learn the network embedding from a single bipartite network.

4.3.1 Bipartite Network Embedding

Given a bipartite network $G = (\mathcal{V}_A \cup \mathcal{V}_B, \mathcal{E})$, the goal of network embedding is to embed each vertex $v_i \in \mathcal{V}_A \cup \mathcal{V}_B$ into a low dimensional vector $\vec{v}_i \in \mathbb{R}^d$, where d is the dimension of the embedding vector. Inspired by (63), we consider to learn the embeddings by preserving the second-order proximity, which means two nodes are similar to each other if they have similar neighbors. In the following, we take the user-location network $G_{up} = (\mathcal{U} \cup \mathcal{P}, \mathcal{E}_{up})$ as an example to illustrate the learning process of embeddings. To begin with, we use a softmax function to define the conditional probability of a user $u_i \in \mathcal{U}$ visits a location $p_j \in \mathcal{P}$:

$$P(p_j|u_i) = \frac{e^{\vec{p}_j^T \vec{u}_i}}{\sum_{k=1}^{M} e^{\vec{p}_k^T \vec{u}_i}} \tag{4.1}$$

To preserve the weight w_{ui} on edge e_{ui}, we make the conditional distribution $P(\cdot|u_i)$ close to its empirical distribution $\hat{P}(\cdot|u_i)$, which can be defined as $\hat{P}(p_j|u_i) = \frac{w_{ij}}{o_i}$, where $o_i = \sum_{p_k \in N(u_i)} w_{ik}$ is the out-degree of u_i, and $N(u_i)$ is the set of the u_i's neighbors, i.e., the locations that u_i have visited.

By minimizing the Kullback-Keibler (KL) divergence between two distributions $P(\cdot|u_i)$ and $\hat{P}(\cdot|u_i)$ and omitting some constants, we can obtain the objective function for embedding the bipartite graph G_{up} as follows:

$$\mathcal{J}_{up} = -\sum_{e_{ij} \in \mathcal{E}_{up}} w_{ij} \log P(p_j|u_i) \tag{4.2}$$

Since a homogeneous network can be easily converted to a bipartite network, we can derive similar objective for embedding social network G_{uu} as follows:

$$\mathcal{J}_{uu} = -\sum_{e_{ij} \in \mathcal{E}_{uu}} w_{ij} \log P(u_j|u_i) \tag{4.3}$$

By jointly learning $\{\vec{u}_i\}_{i=1,\ldots,N}$ and $\{\vec{p}_j\}_{j=1,\ldots,M}$ that minimize the objectives Eq. (Equation 4.2) and Eq. (Equation 4.3), we are able to represent social network users and locations in low dimensional vectors. By far, the embeddings are learned only from the network structure. Next, we introduce the collective geometrical embedding algorithm to preserve the geometric structure w.r.t. the physical closeness in between different objects.

4.3.2 Collective Geometrical Embedding

According to the local invariance assumption (64), if two samples p_i, p_j are close in the intrinsic geometric with regard to the data distribution, then their embeddings $\vec{p}_i$ and $\vec{p}_j$ should also be close. In this work, we consider to preserve the geometric structure of locations by incorporating the following geometric regularization in the learning process:

$$\mathcal{R}(\mathbf{P}) = \sum_{i,j=1}^{M} w_{ij}(\vec{p}_i - \vec{p}_j)^2 \tag{4.4}$$

where the w_{ij} represents the geometric closeness between locations p_i and p_j, which can be obtained with the RBF kernel.

To ease the subsequent derivation, we rewrite Eq. (Equation 4.4) in trace form. Let matrix $\mathbf{U}$ and matrix $\mathbf{P}$ denote the user embedding matrix and the location embedding matrix, respectively, where each row within $\mathbf{U}$ and $\mathbf{P}$ is the embedding vector of a user and a location. Using the weight matrix $\mathbf{W}$ whose element w_{ij} is the weight between two locations and the diagonal matrix $\mathbf{D}$ whose elements $d_{ii} = \sum_{j=1}^{M} w_{ij}$, the Laplacian matrix $\mathbf{L}$ is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Then $\mathcal{R}(\mathbf{P})$ can be reduced into the trace form:

$$\mathcal{R}(\mathbf{P}) = \frac{1}{2}\sum_{i,j=1}^{M} w_{ij}(\vec{p}_i - \vec{p}_j)^2 = \frac{1}{2}Tr(\mathbf{P}^T(\mathbf{D}-\mathbf{S})\mathbf{P}) = \frac{1}{2}Tr(\mathbf{P}^T\mathbf{L}\mathbf{P}) \tag{4.5}$$

To learn the geometrical embeddings from the heterogeneous user network, we minimize overall objective function as follows:

$$\min_{\mathbf{U},\mathbf{P}} \mathcal{J} = \mathcal{J}_{uu} + \mathcal{J}_{up} + \lambda\mathcal{R}(\mathbf{P}) \tag{4.6}$$

where λ is the regularization parameter that controls the importance of the geometric regularization.

Since the edges in different networks have different meanings and the weights are not comparable to each other, we alternatively minimize the objective of each network independently to optimize Eq. (Equation 4.6). The same strategy has also been applied in literature (63), while the geometrical regularization is not considered in previous works. For the objective term of each network, taking $\mathcal{J}_{up}$ as an example, it is time-consuming to directly evaluate as it requires to sum over the entire set of edges when calculating the conditional probability $P(\cdot|u_i)$. We adopt the techniques of negative sampling (65) to approximate the evaluation, where multiple negative edges are sampled from some noisy distribution. More specifically, it specifies the following objective function for each edge e_{ij}:

$$\log \sigma(\vec{p}_j^T \cdot \vec{u}_i) + \sum_{u=1}^{k} E_{p_n \sim P_n(p)}[\log \sigma(-\vec{p}_n^T \cdot \vec{u}_i)] \tag{4.7}$$

where $\sigma(x) = \frac{1}{1+exp(-x)}$ is the sigmoid function, and k is the number of negative edges. The first term shows that if there is a link between vertices u_i and p_j, then force two vectors close to each other. The second term shows after sampling negative links from whole sets of vertices, force two vectors $\vec{u}_i$ and $\vec{p}_n$ far away from each other if there is no link between u_i and p_n. We set the sampling distribution $P_n(p) \propto o_i^{3/4}$ as proposed in (65), where o_i is the out-degree of vertex u_i. For the detailed optimization process, readers can refer to (62). We can minimize the objective term of the social network, $\mathcal{J}_{uu}$, in a similar way.

As for minimizing the geometrical regularization, $\mathcal{R}(\mathbf{P})$, it is to enforce the embedding of each location to be as similar to the locations close to it as possible. Thus, we can sample a location $p_i \in \mathcal{P}$ at each iteration and update its embedding $\vec{p}_i$ by gradient descent. The gradient of $\mathcal{R}(\mathbf{P})$ w.r.t. $\vec{p}_i$ can be derived as follows:

$$\frac{\partial \mathcal{R}(\mathbf{P})}{\partial \mathbf{p}_i} = \sum_j w_{ij}(\mathbf{p}_i - \mathbf{p}_j) = (\sum_j w_{ij} - w_{ii})\mathbf{p}_i - \sum_{j \neq i} w_{ij}\mathbf{p}_j = [(\mathbf{D} - \mathbf{W})\mathbf{P}]_{i*} = [\mathbf{LP}]_{i*}, \tag{4.8}$$

where $[\cdot]_{i*}$ means the i-th row of the given matrix.

The detailed process of the proposed algorithm is summarized in Algorithm 3. After obtaining the geometric embeddings of users and locations, we can train any classifier (e.g., SVM or

Algorithm 3 Collective Geographical Embedding Algorithm

Input: Heterogeneous user network $G_u = G_{uu} \cup G_{up} \cup G_{pp}$, parameter λ, the embedding dimension d, the maximum number of iterations *iter*

Output: Geographical embedding matrix $\mathbf{U}$ and $\mathbf{P}$

//Initialization step

Initialize user embedding $\vec{u}$, location embedding $\vec{p}$

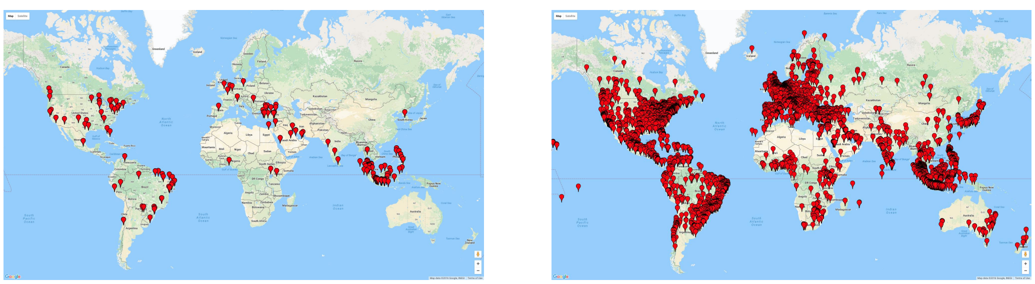
while $j \leq iter$ **do**

Sample an edge from $\mathcal{E}_{uu}$, draw k negative edges and update user embeddings

Sample an edge from $\mathcal{E}_{up}$, draw k negative edges and update user embeddings and location embeddings

Sample a location p_i from $\mathcal{P}$, update the location embedding $\vec{p_i}$ using the partial derivative in Equation 4.8

end while



(a) Foursquare (b) Twitter

Figure 15. Distribution of users' locations in Foursquare and Twitter networks.

logistic regression) by feeding the embeddings as feature vectors and the associated geographic regions at the desired scale (such as city-scale or state-scale) as the labels.

TABLE I

Datasets

Dataset	users	locations	social links	user-location links
Foursquare	15,799	141,444	38,197	212,588
Twitter	25,355	403,770	156,060	564,298

4.4 Experiments

4.4.1 Experiment Setup

To evaluate the performance of the proposed CGE algorithm, we conduct extensive experiments on the following two datasets. The statistics of each dataset is summarized in Table Table I. For both datasets, the social network is constructed from bi-directional friendships between social network users, user-location network is constructed by the users' check-in logs, and users' physical locations reported in their profiles are used as ground truth. We aim to predict users' home location to the city level, since many users only report city-level addresses. City-level location information in text format is converted into city-level coordinates according to geolocators[1]. Note that such coordinates are being canonicalized with each city district corresponding to exactly the same coordinate. Distribution of users' home locations in two datasets is shown in Figure 15. Instead of only focusing on users lived in the US, we are tackling users globally, which creates more challenge for the learning task.

We compared the proposed approach with three state-of-the-art user geolocation prediction algorithms and two network embedding algorithms.

---

[1] https://github.com/networkdynamics/geoinference

1. **FIND** (60) selects the location that maximizes the probability of friendships given the distance between the location candidates and the friends' home locations.

2. **LP** (54) selects the most popular location among the given user's friends' home locations by a simple majority voting algorithm, while the user's friends network were rebuilt via the depth-first search algorithm.

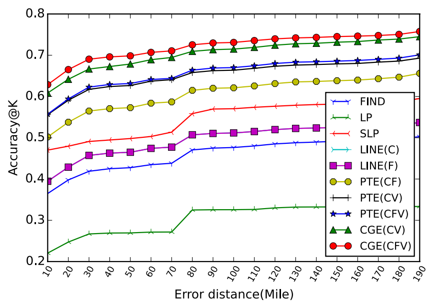
3. **SLP** (55) refers to Spatial Label Propagation. It spatially propagates location labels through the social network, using a small number of initial locations, which is an extension of the idea of label propagation.

4. **LINE** (62) embeds a homogeneous network into a low dimensional space.

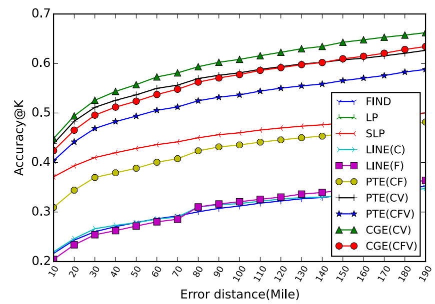
5. **PTE** (63) learns the embeddings of a heterogeneous network by joint learning the embeddings of each sub-network.

6. **CGE** is the proposed method in this paper.

To evaluate the performance of the different approaches, we randomly sample 50% of user instances as the training set and use the other 50% of user instances as the testing set. This random sampling experiment is repeated 10 times. For the FIND algorithm, three coefficients are set the same as in paper (60). For the LP algorithm, the minimal number of friends is set to 1, the maximum number of friends is set to 10000, and the minimal location votes is set to 2. For the SLP algorithm, the number of iterations is set to 5 and the other parameters' settings follow paper (55). For all the embedding algorithms (LINE, PTE, and CGE), the embedding dimensionality is set to 100. We tried dimensionalities in the range $[50, 200]$ and found that 100



(a) Accuracy@k on Foursquare



(b) Accuracy@k on Twitter

Figure 16. Performance comparison on Foursquare and Twitter datasets

generally gives the best results. To simplify the comparison, we simply set the regularization parameter λ in CGE to 1. For the other parameters in the network embedding algorithms, we follow the setting in the paper (63). The learned embeddings are used as feature vectors to train an SVM classifier with the RBF kernel.

To study the contribution of different sources, different combinations of sub-networks in the heterogeneous user network are fed into the algorithms as denoted in the following manner. For CGE taking three networks as inputs, we denote this setting as CGE(CFV), where **C** (check-in) stands for user-location network, **F** (friend) denotes friendship network, and **V** (venue) represents location affinity network. If only one or two networks were taken as inputs, we denote them as (C) or (CV), etc.

Three metrics are used to evaluate the performance of the compared methods. The first metric is **Accuracy**@k, which measures the percentage of predictions that are within k miles of the true location. We report multiple values of k to compare different approaches in a comprehensive manner. The second metric is **A**verage **E**rror **D**istance (**AED**), where a smaller value of which indicates better performance. The third metric is **A**rea **U**nder **C**urve (**AUC**) under a cumulative distribution function $F(x) = P(distance \leq x)$, where $F(x)$ shows the percentage of inferences having an error distance less than x miles away from the true location (52). Higher AUC scores indicate better performance.

4.4.2 Quantitative Results

Figure 16 shows the performance of user geolocation algorithms on two datasets. From the comparison results with regard to Accuracy@k, we make three observations as follows. Firstly, embedding-based algorithms consistently outperform non-embedding based benchmarks. For instance, if we consider Accuracy@30, in Figure 16(a), CGE(CFV) correctly predicts 66.5% of users, while the best performance of non-embedding based algorithms SLP only predicts 49.1% of users. Because embedding-based algorithms can fully explore the network structure of the given information, which alleviates the issues of sparse and noisy signals, embedding-based methods (LINE, PTE and CGE) outperform non-embedding based methods. Secondly, among embedding-based algorithms, algorithms such as PTE and CGE which are capable of handling heterogeneous networks perform better than LINE which is only applicable to homogeneous networks. Thirdly, we can observe that CGE consistently achieves the best performance in both datasets, as shown in Figure 16(a) and Figure 16(b). With exactly the same amount of information, the proposed CGE always outperforms PTE for a variety of error distance k. For example, in Figure 16(a), with user-location network and location affinity network, CGE(CV) correctly predicts 61% of users' home locations within 10 miles, while PTE(CV) correctly

TABLE II

The classification performance "mean ± standard deviation" on user geolocation prediction task. "↑" indicates the larger the value the better the performance. "↓" indicates the smaller the value the better the performance.

	Foursquare		Twitter	
	AED ↓	AUC ↑	AED ↓	AUC ↑
LP	2526.21 ± 34.05	45.52% ± 0.37%	4924.64 ± 18.24	19.30% ± 0.12%
SLP	1673.31 ± 0.73	61.21% ± 0.03%	2172.99 ± 2.40	53.21% ± 0.04%
FIND	1805.88 ± 28.25	57.41% ± 0.39%	2647.07 ± 16.84	42.53% ± 0.20%
LINE(C)	2018.94 ± 30.15	58.60% ± 0.28%	2759.46 ± 20.62	41.92% ± 0.03%
LINE(F)	1308.49 ± 19.04	63.83% ± 0.47%	2474.04 ± 15.23	44.36% ± 0.19%
PTE(CF)	1006.31 ± 21.41	68.80% ± 0.32%	1634.34 ± 16.48	54.30% ± 0.29%
PTE(CV)	1065.06 ± 24.30	71.56% ± 0.20%	1192.38 ± 133.4	63.80% ± 1.22%
PTE(CFV)	935.17 ± 11.50	72.35% ± 0.19%	1247.78 ± 4.79	61.26% ± 0.11%
CGE(CV)	779.94 ± 29.15	75.93% ± 0.35%	**991.22 ± 17.77**	**65.27% ± 0.26%**
CGE(CFV)	**773.31 ± 20.55**	**77.13% ± 0.17%**	1000.47 ± 8.97	64.24% ± 0.07%

predicts 56% of users' home locations within the same distance. These results indicate the robustness of the proposed CGE algorithm.

Table Table II shows the AED and AUC scores of various algorithms on two datasets. Similar observations can be made as above. CGE(CFV) algorithm achieves the smallest error distance and the highest AUC scores for the Foursquare dataset, while CGE(CV) achieves the best performance for the Twitter dataset. This is primarily due to the fact that Twitter relationships mixes friendship relationships with other kinds of unbalanced, asymmetrical relationships (54). More importantly, when using the same data sources, CGE always performs better than PTE. This shows that the proposed graph regularization is more suitable for modeling geographical information in user geolocation problem.

To evaluate the contribution of different sub-networks, we compare the results using partial information with the results using complete information. The comparisons are performed using

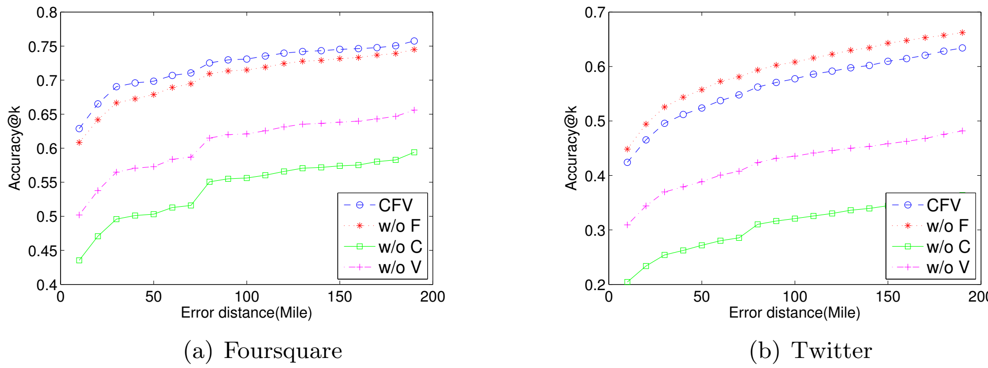
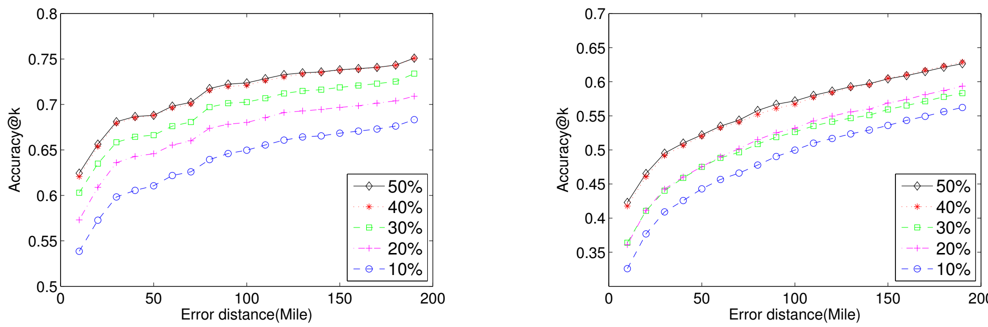


Figure 17. Performance contribution of sub-networks. "w/o" means without certain sub-network.

CGE algorithm on both datasets. As can be seen in Figure 17(a), without user-location network (green line), the performance deteriorates the most (around 19%). Without location affinity network (purple line), performance drops around 13%. Without friend network information, the algorithm drops the least compared with other cases (around 3%). Note that, without friend network information, CGE achieves slightly higher accuracy on Twitter dataset, as shown in Figure 17(b), because Twitter relationships contain heavy noise. It can be concluded that: (1) Compared with friend information and location affinity network, user-location network plays the most important role in user geolocation prediction. (2) Considering the geometrical information in location affinity network can significantly improve the prediction performance. (3) Friend network can also be a valuable complementary source.

The robustness of the proposed algorithm is also tested by varying the size of the training users. Note that, when decreasing the size of the training users, we use locations' embedding



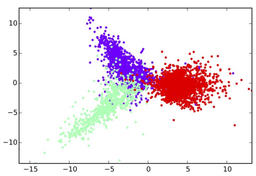
(a) Foursquare

(b) Twitter

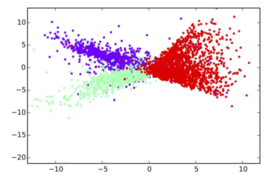
Figure 18. Performance comparison with varied training size.

vectors as additional training data to balance training samples across different settings. As can be seen in Figure 18, when the size of the training users decreases from 50% to 20%, accuracy@k only drops around 5%. The evaluation results on the size of training set indicate that CGE(CFV) is capable of producing high-quality embedding vectors of users and locations.

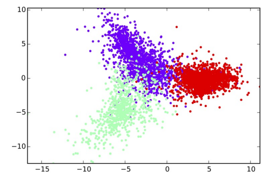
Visualization of users' embedding vectors learned by different algorithms are shown in Figure 19. Due to limited space, only the results of Foursquare dataset are shown. We pick users who reside in three different countries as three different classes. Users' embedding vectors (in 100-dimensional space) are further mapped to two-dimensional space with Isomap. Compared with other algorithms, CGE(CFV) generates the most meaningful layout, as shown in Figure 19(e), in the sense that it naturally forms three clusters and pulls the centers of the different clusters far away from each other. This indicates that the proposed CGE algorithm leveraged different source information effectively. Running time of various algorithms are shown



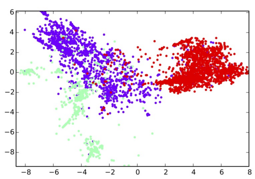
(a) PTE(CF)



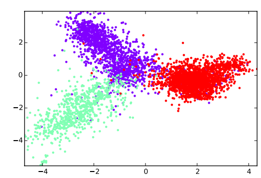
(b) PTE(CV)



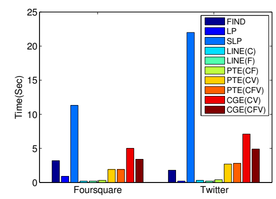
(c) PTE(CFV)



(d) CGE(CV)



(e) CGE(CFV)



(f) Running time

Figure 19. Visualization of users reside in three different countries (Blue: US, Green: Brazil, Red: Malaysia) in Foursquare. Running time comparison (f).

in Figure 19(f). The run time of CGE algorithms are modestly longer compared with other embedding methods, but provides the best prediction performance.

4.5 Related Work

4.5.1 Location Prediction

Works on identifying users' home locations (66) can be roughly divided into two categories based on the information used. One category of related works focus on extracting text information (56; 57) from tweets. The general idea is to extract location-related text information (words, phrase, topic) through language model or probabilistic model. Another category of works focus on social graphs (60; 54; 55), where they rely on the assumption that tie strength is a strong indicator of users' home locations. (60) aims to predict the location of an individual by leveraging geographic and social relationships in the Facebook network. (52) reviews most recent network-based approaches, and proposes two new metrics on comparison of different approaches. (53) studies the problem of using publicly available attributes (mayorship, tips, and likes) and geographic information of locatable friends to infer home location in three networks respectively, Twitter, Foursquare, and Google+. Other works (67; 68; 69) consider text and network information simultaneously. (68) propose an algorithm derived from a generative model. (67; 69) provide two ways of combining the results from network-based approaches and text-based algorithms. However, most of the above-mentioned algorithms were either inefficient or based on simple combination of different source information.

4.5.2 Network Embedding

Recently, network embedding technique (61; 70; 62; 63) drew lots of attention due to the merit of distributed representation learning. Embedding objects into a mutually related common space can mitigate the sparsity problem to a large extent. Moreover, by jointly modeling multiple networks, it is able to capture complex interaction among heterogeneous objects in the connected networks. Different from existing network embedding algorithms, this paper treats the guidance information (locations' geographical information) discriminately as a geometric regularization term to smoothly encode the local geometrical structure into the embedding space.

CHAPTER 5

DEEP AND BROAD LEARNING ON CONTENT-AWARE POI RECOMMENDATION

(This chapter was previously published as "Deep and Broad Learning on Content-aware POI Recommendation", in *The 3rd International Conference on Collaboration and Internet Computing (IEEE CIC 17)* (4).)

5.1 Introduction

As location-based applications rapidly gain popularity, a large volume of online contents with geo-tagged information (check-ins) is created daily. Check-ins, as a direct channel connecting the online and offline worlds, aid the development of many personalized and locational information services, such as personalized advertisement (71), local event promote (72; 73) and city management improvement (74). One of the core tasks towards these services is *Point Of Interest* (POI) recommendation, since it not only helps users enriching their urban experiences but also facilitates the analysis of the crowd mobility and communication.

Most of the prominent approaches to POI recommendation can be divided into three categories: 1) collaborative filtering, 2) sequential pattern modeling and 3) context-aware recommendation. Basically, they are derived to learn three types of information - user preference, check-in sequences, and text information, respectively. Recently, some state-of-the-art models try to learn two types of information simultaneously, such as PRME (70) and FPMC (75), which model user preference and sequential patterns together. However, most of the extended variants of the prominent approaches still relied on the original architecture and integrate other information as side information. There are several drawbacks of these algorithms. First of all, existing POI recommendation algorithms mainly focus on information of users, such as user preference, users' check-in sequence, while ignoring the characteristics of POIs. Second, current algorithms typically model different sources of information with the same metric, such as distances in PRME and transition probabilities in FPMC. However, these symbolized features may not be suitable to handle different form of dependencies. Third, they always model consecutive dependencies but ignores long term dependencies in check-in sequences. Moreover, the above-mentioned models are all shallow models, which cannot capture the highly non-linearity of sequential patterns.

Recently, researchers take the content information of POIs into consideration. Content information can be helpful in various ways. For instance, a user may search a POI's reviews or tips beforehand to decide whether she/he is interested in visiting the place. Therefore, in reality, POIs' reviews or tips can actually be part of the inputs that affect a user's check-in decision. Besides, context information can help identify semantically similar POIs, e.g., 'burgers' often appear in the reviews and descriptions of fast food shops. As shown in recent works (76; 77; 78), integrating context information can be beneficial to alleviate the sparsity problem in POI recommendation. However, most of these works are based on traditional topic models that simply use bag-of-word features and ignore the word orders. Sentences with similar N-grams but total different semantic meanings are hard to differentiate for bag-of-words based technique (79). Therefore, previous methods may not fully uncover semantic information of POIs. Moreover, topic models can be easily affected by the scalability problem and also cannot handle new users and new POIs.

Due to the success of the deep neural networks, researchers have also applied deep models on POI recommendation tasks. Among which, Recurrent Neural Networks (RNN) is especially suitable for sequential prediction. Recently, (80) shows RNN's superior performance on sequential click prediction. By concurrently model spatial and temporal patterns in LBSNs through transition matrix of RNN, (81) achieves promising performance improvement over matrix factorization based and Markov chain-based algorithms.

In order to broadly fuse different sources of information (user preference, check-in sequences, and text information), in this paper we propose a new deep and broad learning model named as Deep Content-aware POI Recommendation model (DCPR) to learn effective representations of POIs and users to facilitate POI recommendation task. In particular, in the proposed model we design a multi-layer deep architecture which consists of multiple deep neural networks. The composition of multiple layers of deep neural networks can first map the data (POI associated with text information) into a highly non-linear latent space (POI space), and then the user representations can be learned through user preference and check-in sequence modeling with deep neural networks in the produced highly non-linear latent space.

Specifically, to enable the content-aware features as well as to address the sparsity problem and long term sequential pattern mining, the proposed model utilizes convolutional neural networks (CNN) model to capture semantic information and common opinion of POIs while preserving the word-orders for the original documents. Then, long short term memory networks (LSTM) is employed to store user preference through modeling check-in sequences to collectively learn user preference from similar users. The LSTM network and CNN network are connected in a structural manner as LSTM learns user preference and sequential patterns with prior knowledge of POIs' semantic information by taking the representational vectors as input from CNN layer. Finally, the personalized ranking layer on top jointly optimize latent representations produced in the first two layers (convolutional layers CNN, recurrent layers LSTM), as it refines the learned latent features in the first two layers towards generating more accurate patterns and better recommendations. The proposed architecture makes DCPR model an end-to-end trainable deep model.

Contributions of this paper is summarized as follows.

1. We propose a deep and broad learning approach based on a deep content-aware model (DCPR) in which content-based POI features and user specific sequential patterns are learned synergistically. The hierarchical model can jointly learn a multi-source heterogeneous network and is robust to sparsity.

2. We propose a structural pair deep learning model, in which the first deep learning algorithm effectively learns an embedding space with latent representations of POIs, and the second deep learning model learns global structure of the constructed embedding space with physical meanings to mine users' mobility patterns. Both the deep representation learning and deep mobility mining are optimized by an unified ranking based objective function.

3. The proposed model is extensively evaluated on three real LBSN datasets. The results demonstrate that it outperforms state-of-the-art sequential modeling methods and deep recommendation models in POI recommendation tasks.

4. The proposed deep learning framework can be employed to solve a generic class of problems involving heterogeneous network learning.

The rest of the paper is organized as follows: Section 2 gives the details of the problem definition. Section 3 illustrates the proposed architecture and mathematical formulation. Section 4 shows the experimental results as well as the discussion. Section 5 presents a review for the state-of-the-art research status. Section 5 concludes the paper.

5.2 Problem Formulation

In this section, we will introduce the problem formulation. Given a set of users $\mathcal{U}$ where $\mathcal{U} = \{u_1, u_2, ...u_N\}$ and a set of POIs $\mathcal{P}$ where $\mathcal{P} = \{p_1, p_2, \ldots, p_M\}$ in a location-based service. N is the total number of users and M denotes the total number of POIs. Each user in $\mathcal{U}$ has a check-in list in chronological order. For instance, user u_i's check-in list is denoted as $\mathcal{C}_{u_i}$, where $\mathcal{C}_{u_i} = \{c^1_{u_i}, c^2_{u_i}, ...c^n_{u_i}\}$. The k-th check-in $c^k_{u_i}$ in the list $\mathcal{C}_{u_i}$ is defined as $c^k_{u_i} = (u_i, p_l)$, which means that user u_i checked in at POI p_l at the k-th time stamp. Each POI in $\mathcal{P}$ is associated with a list of reviews or tips. For example, for POI p_l, its list of reviews/tips is denoted as $\mathcal{R}_{p_l}$, where $\mathcal{R}_{p_l} = \{r^1_{p_l}, r^2_{p_l}, ..., r^m_{p_l}\}$ with $r^j_{p_l}$ indicates the j-th review/tip in POI p_l's review/tip list. Given $G = (\mathcal{U}, \mathcal{P}, \mathcal{C}, \mathcal{R})$, which consists of a list of users $\mathcal{U}$, a list of POIs $\mathcal{P}$, all users' check-ins $\mathcal{C}$, and all POIs' reviews/tips $\mathcal{R}$, the task is to recommend a certain number of POIs for each user based on previous check-ins.

In this paper, we utilize ranking-based loss (82) to train the deep neural networks. For the ranking-based loss, each training sample usually contains a positive item and a negative item. For the proposed problem, each training sample is a sequence of check-in POIs performed by a user, the positive item is the POI checked in after the sequence of check-ins, while the negative item is the POI uniformly sampled from the list of POIs that are not in the user's training sequence. Then, a user, a positive POI, and a negative POI form one training sample. Training data D_s is defined as

$$D_s := \{(u_i, p_j, p_{j'}) | u_i \in \mathcal{U} \wedge p_j \in \mathcal{P}_i^+ \wedge p_{j'} \in \mathcal{P}_i^-\} \tag{5.1}$$

where u_i, p_j, and $p_{j'}$ are uniformly sampled from $\mathcal{U}, \mathcal{P}_i^+, \mathcal{P}_i^-$, respectively. $\mathcal{P}_i^+$ denotes the list of positive POIs for user u_i, while $\mathcal{P}_i^-$ represents the list of negative POIs for user u_i.

5.3 The Proposed Architecture

In this section, we introduce the proposed model DCPR that effectively learns embeddings of users and POIs for POI recommendation through a deep network architecture. DCPR collectively models broad information on check-in sequences and text information with the deep neural network in a hierarchical manner, and it is coupled with probabilistic matrix factorization (83) to provide top-N recommendations for users. The advantages of the proposed DCPR model are two-fold. Firstly, DCPR is an end-to-end deep model which can learns more representative embeddings of users and POIs. Secondly, the proposed model explains how check-in behaviors are formed by modeling text information and check-in sequences in a hierarchical order.

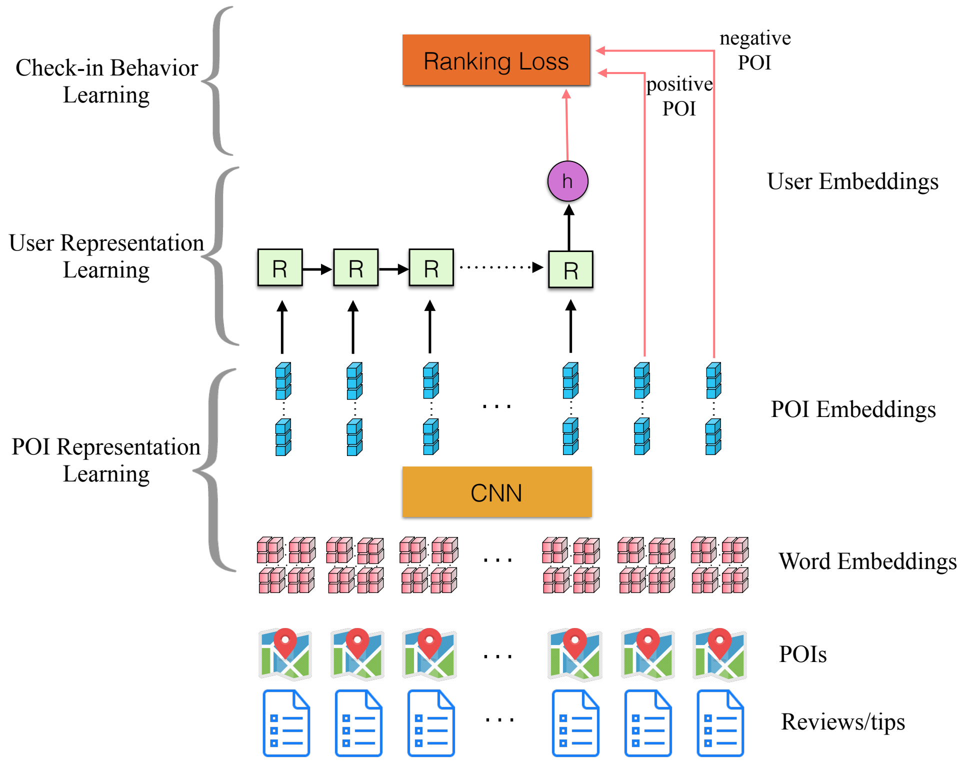


Figure 20. Network Architecture. The architecture contains three components: 1) POI representation learning; 2) user representation learning; 3) check-in behavior learning.

5.3.1 Architecture

The architecture of the proposed framework is illustrated in Figure 20. It consists of three components: POI context extraction, user preference and check-in sequence modeling, and personalized ranking from bottom to top.

At the bottom of DCPR, the POI context extraction component of the algorithm learns semantic information of POIs to generate latent representations from reviews/tips by employing CNN. Above the POI context extraction component is the check-in sequence modeling com-

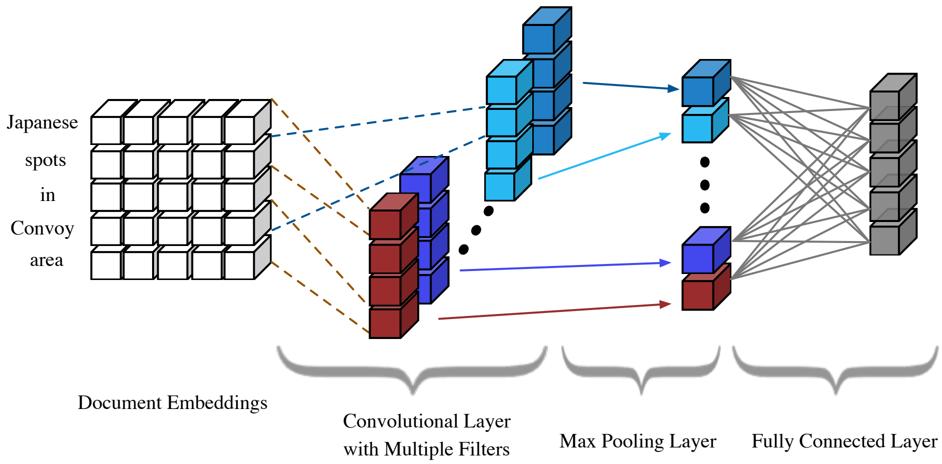


Figure 21. The structure of the POI representation learning component.

ponent, which is responsible for modeling check-in sequences to learn latent representations of users by utilizing LSTM. In the check-in sequence modeling component, rectangle R stands for recurrent cell and h denotes hidden state in LSTM. The POI embeddings learned by CNN from reviews/tips represent POIs' properties and can help explain users checked-ins. Compared to previous models ignoring the textual content, such as (75; 70), it can facilitate the check-in sequence modeling component to learn more effective latent representations of users. Futhermore, the above-mentioned two components are directly connected and organized in a hierarchical order. At the top of DCPR is the personalized ranking component which optimizes the latent representations of users and POIs following the fashion of probabilistic matrix factorization (83).

5.3.2 POI Representation Learning

Given all POIs' reviews/tips, we aim to learn latent representations of POIs to facilitate POI recommendations. Intuitively, when a user searches a POI online, he/she is more likely to browse some of the reviews/tips to sum up the property and general opinions of this POI. To mimic this online behavior and accurately model POIs from their textual content, we propose to learn model POIs from their reviews/tips.

To sum up all reviews/tips belonging to one POI, we firstly concatenate all reviews/tipcs of the POI into one document. Formally, for the q-th POI p_q, its list of reviews/tips can be concatenated into one document d^q. The d^q contains semantic information and common opinions of the q-th POI. This helps to construct a meaningful solution space and facilitate the prediction of users' future check-ins. Also, it helps learn the users' historical behavior more effectively and boost the performance of prediction.

Given a document d^q of POI p_q, before feeding to the POI context extraction component, we first apply a word embedding function, denoted as Φ, on each word of d^q. Φ maps each word into a n-dimentional vector.

Assume there are N words in document d^q, then an embedding matrix Pi of document d^q is represented as:

$$\mathbf{\Pi}^q = \Phi(w_1) \oplus \Phi(w_2) \oplus ... \oplus \Phi(w_n) \oplus .. \oplus \Phi(w_N) \tag{5.2}$$

where Φ is a word embedding function mapping each word to a n-dimentional vector, $\mathbf{\Pi}^q$ denotes the embedding matrix of document d^q, and $\oplus$ is the concatenation operator. Note that n-th column of Φ corresponds to embedding of n-th word in document d^q.

Following the embedding function, three inner layers inside CNN, including a convolution layer, a max-pooling layer and a fully connected layer, are built to learn feature vectors of POIs. The structure of the POI representation learn component is illustrated in Figure Figure 21. Next, we will explain these three layers in details.

Convolutional layers apply convolution operator on document embeddings to generate new features. A convolution operation corresponds to a neuron in neural networks. It employs a filter $K_j \in \mathbb{R}^{h \times t}$ to a window of h words to generate a new feature. For example, applying convolution operation on document d^q produces feature z_j is defined as follows.

$$z_j^q = f(\mathbf{\Pi}^q * K_j + b_j) \tag{5.3}$$

where z_j^q is the new convolution feature produced by filter K_j, $\mathbf{\Pi}^q$ is the q-th document that convolution operation works on, symbol $*$ is the convolution operator, b_j is the bias term, and f is the activation function. Rectified Linear Units (ReLUs) (84) are used as activation units. It has been shown that using ReLUs as activation units in CNN effectively shortening the training time of neural networks (85). The equation of ReLUs is

$$f(x) = max\{0, x\} \tag{5.4}$$

Following the convolution layer, a max pooling operation is applied on the newly produced features as Equation 5.5

$$l_j = max\{z_j^1, z_j^2, ..., z_j^{n-h+1}\} \tag{5.5}$$

Here, l_j denotes the feature corresponding to filter K_j. For all of the filters, the produced features after max pooling layer is

$$L = \{l_1, l_2, l_3, ..., l_{n_1}\} \tag{5.6}$$

where n_1 denotes the number of filters (neurons). The output of max pooling layer is feed into a fully connected layer as:

$$x_q = f(W \times L + g), \tag{5.7}$$

where W is the weight matrix in the fully connected layer, $x_q \in \mathbb{R}^{n_2 \times 1}$ is latent features of the q-th POI. The fully connected layer is designed to learn non-linear combination of extracted features from convolution and max pooling operations.

5.3.3 User Representation Learning

In this section, we aim to model a user's interests from the user's past POI sequences. Traditional approaches represents each POI with one-hot encoding and lose the rich semantic information existing in the textual information. In order to utilize the semantic information, in this paper, we build a check-in sequence modeling component to utilize POI representations from the POI representation learning component. Given a list of POIs for a user i and their cor-

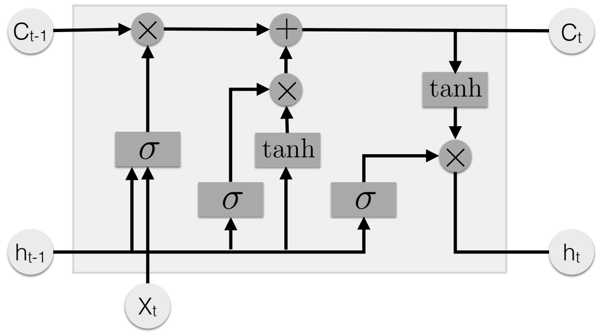
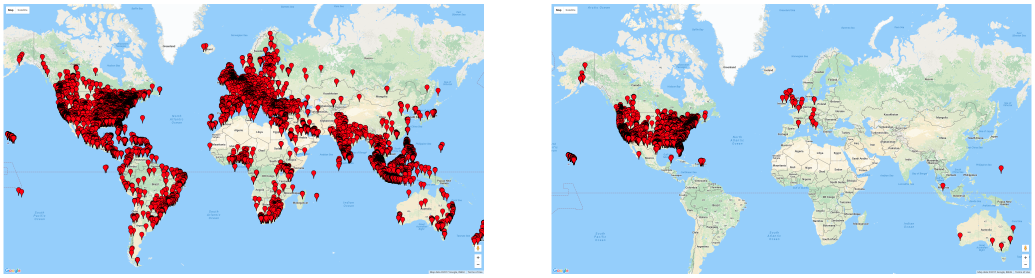


Figure 22. Basic structure of LSTM.

responing embeedings, the sequence modeling component generates a vector as the embeeding of user i.

Check-in sequence modeling component employs long short term memory networks (LSTM) (86) to model check-in sequences with long term dependencies. The architecture of the LSTM is illustrated in Figure Figure 22. A special mechanism is involved in the basic structure of LSTM which includes memory cell, input and output gate, and forget gate. Different parts work collaboratively to store and access information in memory cell which is a unique part introduced in LSTM to handle long term dependency problem, particularly. The equations introduced in this special mechanism is listed as follows.



(a) Data distribution@4sq (b) Data distribution@yelp

Figure 23. Global distribution of POIs' location in Foursquare and Yelp datasets.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5.8}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5.9}$$

$$\widetilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_C) \tag{5.10}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{5.11}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5.12}$$

$$h_t = o_t * \tanh(C_t) \tag{5.13}$$

where, W_f, W_i, W_c, and W_o are weight matrices, and b_f, b_i, b_C, and b_o are bias terms. Equation (Equation 5.8) works in forget gate layer, which calculates how much information should be discarded for memory cell. In equation (Equation 5.8), h_{t-1} denotes hidden state in last time stamp, x_t stands for the input at the time stamp t, b_f is the bias term in forget gate, f_t determines how much information should be kept in memory cell, and σ is the sigmoid function. In our scenario, input x_t is the embedding vector of POI checked in at this time stamp. Equation (Equation 5.8) decides what information to forget for the memory cell in last time stamp, while equations (Equation 5.9) and (Equation 5.10) determine what new information should be stored in the new memory cell. Equation (Equation 5.9) works in input gate layer, which decides which values will be updated according to last hidden state and input, and equation (Equation 5.10) deploys a tanh layer to create a vector of new candidate values $\widetilde{C}_t$. Equation (Equation 5.11) updates memory cell values in this time stamp. Equations (Equation 5.12) and (Equation 5.13) calculate the values in the new hidden state based on values in the new memory cell and hidden state in last time stamp as well as input of this time stamp.

5.3.4 Check-in Behavior Learning

Ranking based loss function attracts lots of attentions lately (75; 70) since it directly optimizes the ranking order of POIs. Essentially, to recommend POIs is to provide a ranking on the list of POIs with top POIs with high probability to be visited by user. Inspired by Bayesian Personalized Ranking (87), we model the conditional probability over POI j's latent features with Gaussian distribution as

$$p(\mathbf{v}_j|x_j, \lambda) = \mathcal{N}(\mathbf{v}_j|x_j, \lambda_v \mathbf{I}) \tag{5.14}$$

where $\mathbf{I}$ is a $K \times K$ identity matrix. Similarly, conditional probability over user i's latent representation with Gaussian distribution is defined as

$$p(\mathbf{u}_i|h_i, \lambda) = \mathcal{N}(\mathbf{u}_i|h_i, \lambda_u \mathbf{I}) \tag{5.15}$$

where $\mathbf{I}$ is also a $K \times K$ identity matrix. The goal is to maximize the difference between positive POI and negative POI. The difference probability given user $i \in U$, positive POI $j \in P_i^+$, and negative POI $j' \in P_i^-$is defined as

$$p(r_{i,j,j'}|\mathbf{u}_i, \mathbf{v}_j, \mathbf{v}_{j'}) = \sigma(\mathbf{u}_i^T \mathbf{v}_j - \mathbf{u}_i^T \mathbf{v}_{j'}) \tag{5.16}$$

where $\mathbf{u}_i$, $\mathbf{v}_j$, and $\mathbf{v}_j'$ are latent features of user u, POI j, and POI j'. Furthermore, σ is sigmoid function.

For optimization, we utilize the technique of MAP. Maximizing the posterior probability of $\mathbf{u}$, $\mathbf{v}$, and parameters in deep neural networks is to minimize the negative of log-likelihood.

$$\begin{aligned} \mathcal{L} = - \sum_{(i,j,j') \in D_S} \{ &\log\sigma(\mathbf{u}_i^T \mathbf{v}_j - \mathbf{u}_i^T \mathbf{v}_{j'} \\ &+ \frac{\lambda_u}{2}(\mathbf{u}_i - h_i)^T(\mathbf{u}_i - h_i) + \frac{\lambda_v}{2}(\mathbf{v}_j - x_j)^T(\mathbf{v}_j - x_j))\} \end{aligned} \tag{5.17}$$

The first term of equation (Equation 5.17) enforces user preference in the way of maximizing the difference between product of user factors with positive embeddings and product of user factors with negative embeddings. The second and third terms forces u_i and v_j to be close to user i's latent factors and POI j's latent features respectively. Stochastic Gradient Descent algorithm (88) is utilized to minimize the loss function.

5.4 Experiments

To test whether the proposed architecture can effectively modeling users' check-in sequences and extracting semantic information from text, we evaluate the performance of the proposed framework and state-of-the-art baselines in this section with various metrics and case studies.

5.4.1 Experiment Setting

We conduct extensive experiments to evaluate the proposed DCPR algorithm on the following three datasets. The statistics of each dataset is summarized in Table Table III. For Foursquare and Yelp datasets, check-ins are from tweets in Twitter network. Each tweet's source indicates whether it is a Foursquare check-in or Yelp check-in or something else. Foursquare tips are from Foursquare network. Yelp reviews are from Yelp website. To remove users or POIs with too few check-ins, we filter out users with less than 5 check-ins and POIs with less than 3 visits for both Foursquare and Yelp datasets. TIST dataset is a public dataset. It is originally utilized for monitoring and visualizing global check-in behaviors (89). Note that, this dataset only contain user check-ins and it does not contain text information. For TIST dataset, we remove users with less than 20 check-ins and POIs with less than 20 visits. Distributions of POIs' locations in Foursquare and Yelp datasets are shown in Figure 23. Yelp dataset contains more check-ins in the United States, while Foursquare dataset includes check-ins spreading worldwide which creates more challenges for the learning task. We omit the description for TIST dataset, please refer to the paper (89) for more details.

TABLE III

Datasets

Dataset	Foursquare	Yelp	TIST
Users	74,140	30,367	266,909
POIs	104,844	25,728	3,680,126
Check-ins	418,081	146,456	33,263,631

We compared the proposed approach with two state-of-the-art POI recommendation algorithms (FPMC, PRME), one traditional recommendation algorithm (FM), and two deep models (RNN, CDL).

1. **FPMC** (75) refers to factorized personalized Markov chains model, which constructs a transition tensor to model the probability of users' next behavior based on previous behaviors. A factorization model is proposed to decompose the tensor to estimate the probability. The factorization model is able to learn information among similar users and similar items.

2. **PRME** (70) stands for personalized ranking metric embedding model. It learns two embeddings in two separate spaces. One embedding is based on sequential transition probability, while the other embedding is based on user preferences. Each user's top-N recommendation is based on linear combination of the learned embeddings.

3. **FM** (90) refers to **F**actorization **M**achine. It models pairwise interactions between all features. Note that, for the proposed problem, there are three types of features constructed

for FM, including one hot encoding of users, combinations of one hot encoding of POIs in check-in sequences, and one hot encoding of POI checked in after the sequences.

4. **RNN** (80) is the state-of-the-art deep model for sequential prediction by adopting recurrent neural networks.

5. **CDL** (91) jointly models text information with deep representation learning and user feedback with collaborative filtering.

6. **DCPR** is the proposed method in this paper.

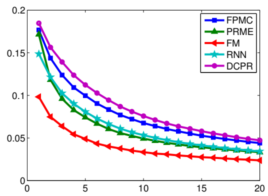
To evaluate the performance of the different approaches, for each user, we pick the first 80% of check-ins as training data, and the remained 20% of check-ins are considered as testing data. For the FPMC algorithm, the training data is further divided into 80% and 20%, for training and validation, respectively. Learning rate is set to 0.005, the parameter for the regularization term is set to 0.03, and the factorization dimension is set to 20. For the PRME algorithm, parameter α and latent dimension are set to 0.02 and 60 respectively, which follows the setting in the original paper. For the RNN algorithm, the dimension of POIs' embeddings is set to 50, the number of neurons in the recurrent layer is set to 64, cross entropy is employed as the loss function. For the proposed DCPR algorithm, embedding dimension of POIs is set to 50. For the convolution layer, the number of filters is set to 100, filter length is set to 3. The number of neurons in the fully connected layer and the recurrent layer is set to 50. Note that, we use different latent dimensions for different comparison algorithms to optimize the performance for each case.

Three metrics are used to evaluate the performance of the compared methods. The output of the compared methods is a ranked list of all POIs which indicate the likelihood of the POI being checked in at the testing period from high to low. The first metric is **Precision**@N, which measures the percentage of correct predictions in the top-N ranked list. The second metric **Recall**@N measures the percentage of correct predictions in the top-N ground truth set. Note that, top-N ground truth set is constructed based on the time difference between training check-in sequence and testing check-ins. The closer the time difference is, the higher position the POI's takes in the top-N ground truth list. The third metric **F1-score**@N is the harmonic mean of above-mentioned two metrics, which shows a comprehensive evaluation of the compared methods.

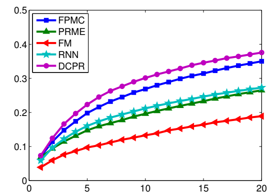
5.4.2 Performance Comparison

Figure 24 shows the performance of POI recommendation on Foursquare and Yelp datasets with metrics Precision@N, Recall@N, and F1-score@N. N varies from 1 to 20. Four observations are made as follows.

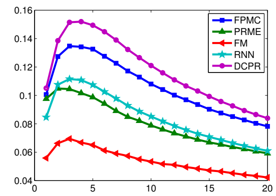
- DCPR consistently outperform other compared methods in two datasets, as shown in Figure 24. Although in yelp dataset, PRME achieves slightly better results when $N = 1$, the proposed DCPR algorithm performs the best in most of general cases. The reason that PRME shows slightly higher results is because that PRME utilizes metric embedding technique to model sequential transition probability. The metric embedding technique is designed to learn transition probability between consecutive check-ins, but it can not model long term sequential influences. In contrast, the proposed DCPR algorithm employ



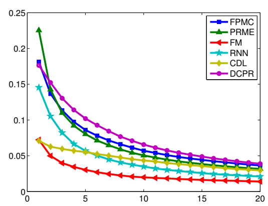
(a) Precision@TopN (4sq)



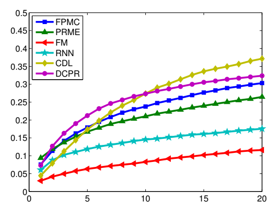
(b) Recall@TopN (4sq)



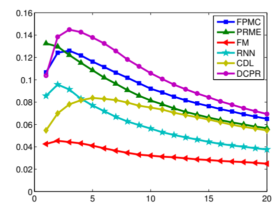
(c) F1-score@TopN (4sq)



(d) Precision@TopN (Yelp)



(e) Recall@TopN (Yelp)



(f) F1-score@TopN (Yelp)

Figure 24. Performance on Foursquare and Yelp datasets.

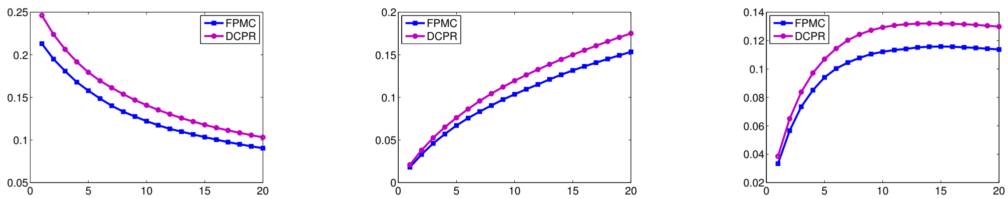
special recurrent structure to particularly modeling long-term dependencies, therefore, DCPR wins in almost all of the varied N.

- FM usually performs well in rate prediction tasks, while it achieves inferior results compared to other methods in POI recommendation task. Although FM captures all pairwise interactions between all features, the model is incapable of differentiate the importance of different feature interactions. Therefore, it is not able to focus on important feature interactions and ignore insignificant ones. In comparison, the proposed DCPR have different parts to specialize on modeling specific type of information and jointly learn the importance of each part in one loss function. Therefore, it achieves superior results compared to FM.

- The proposed DCPR outperforms other two deep neural network based models. It can be seen from Figure 24 that DCPR achieves much higher accuracy compared with typical RNN algorithms and CDL as well. Even though RNN tries to model check-in sequences, long term dependencies may not be captured by deep recurrent neural networks. Also, RNN ignores text information and thus loses another source of information to tackle the problem. Although CDL algorithm learns deep representation for content information, it is not capable of modeling sequential influence. The proposed DCPR algorithm models text information and check-in sequences simultaneously, so it outperforms RNN and CDL with big margin.

- Comparing the performance of the comparison methods on three different metrics, we observe that Precision@N and Recall@N always monotonically decrease or increase in all three datasets, while Recall@N shows non-monotonic trending. It increases first and then decrease. It is worth noticing that DCPR almost always achieves the biggest improvement when the comparison algorithms are at their highest F1-score. For example, in Foursquare dataset, evaluated in F1-score, when $N = 4$, DCPR achieves 13% improvement over FPMC, and DCPR obtains 128% improvement over FM. Interestingly, for Foursquare and Yelp datasets, almost all algorithms perform best when $N = 4$. It is probably because Foursquare and Yelp datasets contain more users with short sequences.

From Figure 24, we can conclude that FPMC is the best performing baseline method. Therefore, for the large-scale TIST dataset, we only compare the performance of the proposed DCPR algorithm with FPMC in Figure 25. Since the TIST dataset does not contain text

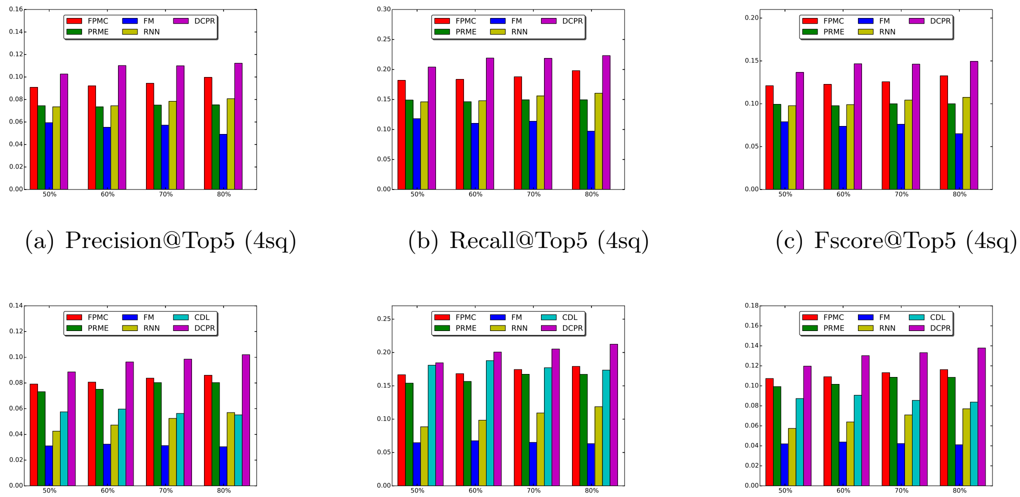


(a) Precision@TopN (TIST) (b) Recall@TopN (TIST) (c) F1-score@TopN (TIST)

Figure 25. Performance on TIST dataset.

information, we accommodate the proposed DCPR algorithm to only generate embeddings for POIs by omitting the convolution feature generating process. We can see that, for all three different metrics, DCPR always outperforms FPMC with a big margin. For instance, for F1-score@N metric, when N equals to 12, DCPR achieves 15.2% improvement over FPMC. For the F1-score@N metric, both algorithms perform the best when $N = 15$. It is probably because TIST dataset include users with longer sequences compared to that of Foursquare and Yelp datasets. It shows that the proposed DCPR is robust in terms of varying sequence length. Also, compared to the performance on foursquare and yelp datasets, the proposed DCPR algorithm achieves the largest improvement in TIST dataset. It is probably due to the reason that the proposed DCPR is especially good at modeling long term dependencies and average sequence length of TIST dataset is much longer than that of other two datasets.

The robustness of the proposed algorithm is also tested by varying the size of the training check-ins in Foursquare and Yelp datasets. Also, we pick $N = 5$ for illustration purpose. As can be seen in Figure 26, the proposed DCPR always outperform other compared algorithms. For



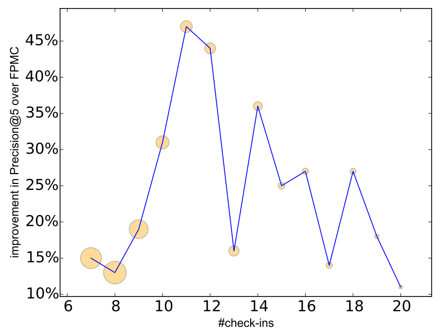
(a) Precision@Top5 (4sq) (b) Recall@Top5 (4sq) (c) Fscore@Top5 (4sq)

(d) Precision@Top5 (4sq) (e) Recall@Top5 (4sq) (f) Fscore@Top5 (4sq)

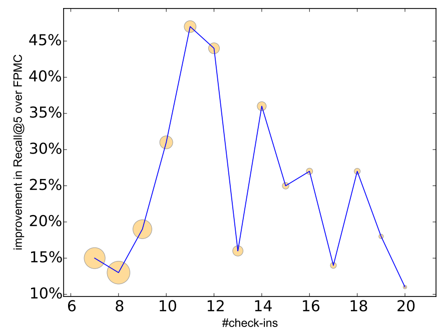
Figure 26. Performance comparison with varied training size.

instance, in Yelp dataset, when the size of the training data increase from 50% to 80%, FPMC's Recall@Top5 increases 7.54%, while DCPR's Recall@Top5 increases 14.92%. The evaluation results on the size of training set indicate that DCPR is capable of producing high-quality embedding vectors of users and POIs.

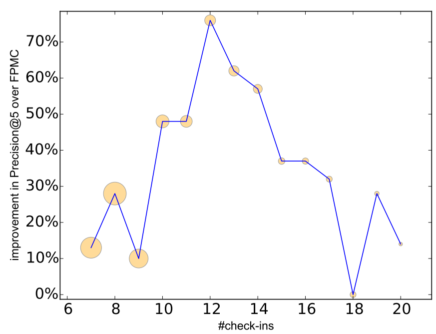
Besides evaluating the proposed approach on the whole dataset with different metrics in macro level, we also show a comprehensive study on the performance of the compared approaches in micro level. Specifically, we study the performance of the proposed algorithm on different users groups where users are clustered according to the length of their check-in sequences. As an illustration example, Figure 27 shows the gain of DCPR over FPMC in Precision@5 and Recall@5. We pick users with modestly long sequences in the overall population



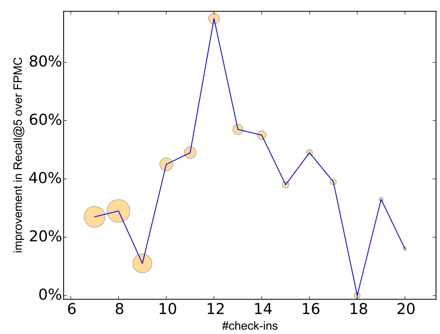
(a) Precision@Top5-4sq



(b) Recall@Top5-4sq



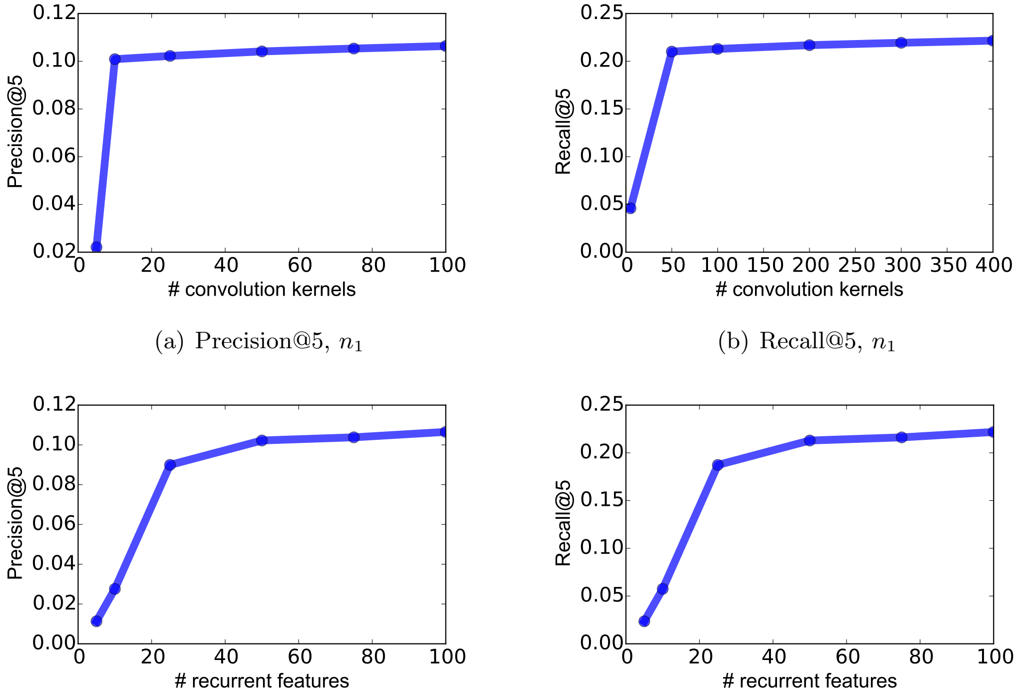
(c) Precision@Top5-Yelp



(d) Recall@Top5-Yelp

Figure 27. Gain over FPMC on Foursquare and Yelp datasets.

for Foursquare and Yelp datasets. At the same time, population density of each group is shown to provide in depth understanding of the performance of different algorithms. The population density of each groups is indicated by the size of orange marker in Figure 27. First of all, in all of the different group of users, DCPR achieves larger than 10% improvements. Interestingly, for both datasets, highest improvement always achieved when users having 11 or 12 check-ins. For instance, when sequence length equals to 11, the proposed algorithm achieves nearly 50%



(a) Precision@5, n_1

(b) Recall@5, n_1

(c) Precision@5, n_2

(d) Recall@5, n_2

Figure 28. Sensitivity Analysis.

improvement over FPMC in Foursquare dataset, while it also improves FPMC nearly 70% in Yelp dataset. The possible reason for this observation is that when feeding too long a sequence from the past may contain more noise, while too short a sequence does not capture enough behavior information.

5.4.3 Sensitivity analysis

We perform the sensitivity analysis in Figure 28 on two parameters: one is the number of convolution kernels n_1, while the other is the number of latent recurrent features n_2. These results are all based on Yelp dataset due to space limitation. Upper two figures show results of n_1, while bottom two figures display results of n_2. First column's figures display analysis on the Precision@5, while the second column's figures indicate analysis on Recall@5. As can be seen, for parameter n_1, when it increases from 5 to 50, values increase, however, when it increases beyond 50, values almost stay same. For the parameter n_2, when it increases from 5, the performance increases drastically, when it reaches 50, the performance stays evenly. Therefore, for the proposed DCPR algorithm, we pick the number of convolution kernels equals to 100 and the number of recurrent features as 50.

5.5 Related Work

5.5.1 POI Recommendation

Similar like the traditional recommender systems, matrix factorization technique is introduced in POI recommendation (92; 93). Different from item recommender systems which employ explicit user feedback such as ratings, POI recommendation utilize implicit user behavior (check-ins) as user feedback. Other implicit information is introduced such as location of check-in POIs, temporal information of check-ins, and social networks. Some recent works focus on leveraging geographical (92; 93), social influences (93) and temporal effects. (93) combines users' preference, social influence, and geographical influence based on matrix factorization framework. (92) proposes a GeoMF model which jointly models geographical information and user preference. (94) introduces ranking based loss into the GeoMF model.

Sequential pattern mining gains lots of attentions lately in personalized recommendation (75; 95). Rendle at al. (75) proposes a FPMC model which constructs a personalized probability transition tensor based on Markov chains. Then, a factorization model is proposed to estimate the transition tensor. FPMC model is extended by incorporating geographical constraints (96). Embedding technique (70; 97) attracts lots of research attentions lately since it is capable of learning better representations for various tasks. Personalized Ranking Metric Embedding (PRME) (70) model learns embeddings in two separate spaces which models sequential transition probability and user preference. Bayesian personalized ranking loss is introduced to combine learned embeddings to predict future check-ins. Instead of learning POI representations only from previous check-ins, (97) proposes to learn representations from surrounding check-ins inspired by skip-gram. (98) incorporates skip-gram model with bayesian personalized ranking loss. Even though PRME also models sequential pattern and user preference, simply linear combination of embeddings cannot explain the complex relationship interacted between these two factors.

5.5.2 Context-Aware Recommendation

Although spatial, temporal, and social information have been investigated in POI recommendation, text information is relatively less explored in POI recommendation(99; 77; 78). Text information includes reviews, tags, tips, and categories, etc. (77) proposes a topic and location aware probabilistic matrix factorization model using POI-associated tags. Firstly, users' interest with respect to semantic topics is learned from text information of POIs through Latent Dirichlet Allocation (LDA) model. Then, learned users' topic interests is compared with POIs' topic distribution to find potential POIs utilizing probabilistic matrix factorization. Meanwhile, word-of-month opinions are considered in the above-mentioned factor-based model. Yang et al. (76) employs sentiment analysis techniques to extract users' preference from text information (tips). And then, preference inferred from contents is considered simultaneously with preference learned from users' check-in behavior. Factor analysis framework is also extended to model geographical influence (93). Similar to LDA model, (78) proposes a spatial topic model by simultaneously modeling spatial and content information in Twitter networks. (100) investigates personal and local preferences from POIs' contents. (99) exploits contents associated with POIs' and comments written by users with weighted matrix factorization. (101) models personal preferences and sequential influence with a latent probabilistic generative model.

Above-mentioned models learn text similarity only based on lexical similarity. Two reviews can be semantically similar when they have low lexical overlaps, as English vocabulary is very diverse. These works ignores semantic meaning which plays an important role in understanding POIs. In addition, topic modeling-based approaches can easily be affected by sparsity problem and also cannot cope with new users and POIs.

5.5.3 Deep Learning for Recommendation

Lately, neural network based methods attract lots of attentions not only because it generates useful representations for various learning tasks but also delivers state-of-the-art performance on natural language processing and other sequential modeling tasks (80; 81). Among which, Recurrent Neural Networks (RNN) is especially good at modeling sequence (102; 103). For example, (80) shows RNN's superior performance on sequential click prediction. By concurrently model spatial and temporal patterns in LBSNs through transition matrix of RNN, (81) achieves promising improvement over matrix factorization-based and markov chain-based algorithms.

Researchers start to focus on employing neural network based models for traditional recommender systems (104; 91). (104) proposes an item recommendation algorithm which jointly models users and items from reviews utilizing deep neural networks.

As discussed above, while there are studies try to model sequential pattern in check-in sequences and review text in item recommender system, they did not address both challenges simultaneously. Instead of learning sequence from markov chain-based models, the proposed DCPR model learns personalized sequential behaviors with the aid of advanced deep model. Instead of only relying on topic modeling based models to handle review text, the proposed DCPR learns the semantic meaning and sentimental attitudes of reviews with deep CNN model.

CHAPTER 6

CONCLUSION

(Part of the chapter was previously published in (1; 2; 3; 4).)

In this thesis, we have explored modeling and knowledge discovery in location-based social networks. Towards this direction, we thoroughly studied four different research problems: goal-oriented co-clustering, collaborative co-clustering across multiple social media, geolocating social media users by geographical network embedding, deep POI recommendation. The effectiveness of the proposed models and algorithms are evaluated by extensive experiments on various real-world datasets. The contributions of our work are summarized as below:

- First, we studied the problem of goal-oriented co-clustering in social networks. We proposed a novel framework that integrated user provided information to produce co-clusterings according to multiple goals. The SGCC model was proposed to take selected feature clusters as inputs and directly create co-clusterings. Further, the FGCC model was proposed to incorporate subspace learning technique to make full use of all features to create co-clusterings according to multiple goals. Evaluation on two real world datasets from Foursquare and Yelp shows that both SGCC and FGCC models can meet user's expectation and produce high quality co-clusterings. Particularly, the FGCC model achieves better results since it makes full use of all features. Case studies on place and user clusters show that FGCC model did live up to the goal-oriented expectation. Another case study shows the possibility that clustering results can be used in social recommendation. Since recommendation is a highly important task in location-based social networks and review networks, we expect to apply the proposed models to real recommendation tasks.

- Second, we formulate the problem of co-clustering on multiple source information in a multi-view fashion. The relationship matrix is used to construct *relationship view*, while features of each individual objects from different sources are used to construct *feature views*. We propose a collaborative co-regularization co-clustering model (**Co-CoClust**), which learns the co-cluster from *relationship view* and *feature views*. Co-regularization term is proposed to regularize "co-cluster" from *relationship view* and "cluster" from *feature view*. Alternating minimization is utilized to learn co-cluster iteratively. The proposed **Co-CoClust** is compared with 9 baselines on two benchmark datasets. We also provide a case study on social network dataset (**Foursquare+Twitter**), which demonstrates efficacy and robustness of the proposed approach. For future work, instead of considering two types of objects in a multi-view setting, we would like to see an extension to multiple types of objects that are related with each other in multiple sources. Another interesting extension of our work could be focusing on feature selection in multi-view setting since different source provide rich information with regards to each type of objects.

- Third, we proposed a collective geometrical embedding (CGE) algorithm to tackle the problem of geolocating users. Multiple heterogeneous networks are embedded into a low dimensional space through two strategies: the first is to embed the social network and the user-location network by preserving local structures; while the other is to incorporate the geographical information as the guidance through graph regularization. Evaluation on two different real-world datasets demonstrated the effectiveness of the proposed approach. For future work, multiple types of social links and multiple types of user-location relations can be included in the proposed framework. Besides, the proposed embedding method can be further extend for location recommendation.

- Lastly, we proposed a deep content-aware POI recommendation (DCPR) algorithm to tackle the problem of POI recommendation. Broad learning from multiple sources of information is utilized to solve this challenging problem. Specifically, text information associated with POIs and users' check-in sequences are simultaneously modeled in this paper. Furthermore, two different types of deep neural networks are combined in an architectural framework with each one learns one information source, and finally a ranking-based loss is introduced to learn the users' overall check-in behaviors. The proposed DCPR model learns different source information discriminatively. Therefore, it can synergistically learns multi-source heterogeneous networks. To this end, it is a deep and broad learning model. Evaluation on three different real-world datasets demonstrated the effectiveness of the proposed approach. For future work, other side information such as temporal information and geographical information can be included in the proposed framework. Besides, the proposed deep framework can be further extended for event recommendation.

APPENDICES

RightsLink®

Home | Create Account | Help

IEEE Requesting permission to reuse content from an IEEE publication

Title:	Concurrent goal-oriented co-clustering generation in social networks
Conference Proceedings:	Semantic Computing (ICSC), 2015 IEEE International Conference on
Author:	Fengjiao Wang
Publisher:	IEEE
Date:	Feb. 2015

LOGIN

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK | CLOSE WINDOW

Copyright Clearance Center RightsLink®

Home | Create Account | Help

IEEE
Requesting permission to reuse content from an IEEE publication

Title:	Collaborative Co-clustering across Multiple Social Media
Conference Proceedings:	Mobile Data Management (MDM), 2016 17th IEEE International Conference on
Author:	Fengjiao Wang
Publisher:	IEEE
Date:	June 2016

LOGIN

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK | CLOSE WINDOW

SPRINGER LICENSE
TERMS AND CONDITIONS

Nov 09, 2017

This Agreement between Fengjiao Wang ("You") and Springer ("Springer") consists of your license details and the terms and conditions provided by Springer and Copyright Clearance Center.

License Number	4224930704423
License date	Nov 09, 2017
Licensed Content Publisher	Springer
Licensed Content Publication	Springer eBook
Licensed Content Title	Collective Geographical Embedding for Geolocating Social Network Users
Licensed Content Author	Fengjiao Wang, Chun-Ta Lu, Yongzhi Qu et al
Licensed Content Date	Jan 1, 2017
Type of Use	Thesis/Dissertation
Portion	Full text
Number of copies	1
Author of this Springer article	Yes and you are a contributor of the new work
Order reference number	
Title of your thesis / dissertation	Data Analysis on Location-based Social Networks
Expected completion date	Nov 2017
Estimated size(pages)	133
Requestor Location	Fengjiao Wang 1111 S Laflin St Apt 1310 CHICAGO, IL 60607 United States Attn: Fengjiao Wang
Billing Type	Invoice
Billing Address	Fengjiao Wang 1111 S Laflin St Apt 1310 CHICAGO, IL 60607 United States Attn: Fengjiao Wang
Total	0.00 USD

Terms and Conditions

Introduction
The publisher for this copyrighted material is Springer. By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at http://myaccount.copyright.com).
Limited License
With reference to your request to reuse material on which Springer controls the copyright, permission is granted for the use indicated in your enquiry under the following conditions:
- Licenses are for one-time use only with a maximum distribution equal to the number stated in your request.
- Springer material represents original material which does not carry references to other sources. If the material in question appears with a credit to another source, this permission is not valid and authorization has to be obtained from the original copyright holder.
- This permission
• is non-exclusive
• is only valid if no personal rights, trademarks, or competitive products are infringed.
• explicitly excludes the right for derivatives.
- Springer does not supply original artwork or content.
- According to the format which you have selected, the following conditions apply accordingly:
• **Print and Electronic:** This License include use in electronic form provided it is password protected, on intranet, or CD-Rom/DVD or E-book/E-journal. It may not be republished in electronic open access.
• **Print:** This License excludes use in electronic form.
• **Electronic:** This License only pertains to use in electronic form provided it is password protected, on intranet, or CD-Rom/DVD or E-book/E-journal. It may not be republished in electronic open access.
For any electronic use not mentioned, please contact Springer at permissions.springer@spi-global.com.
- Although Springer controls the copyright to the material and is entitled to negotiate on rights, this license is only valid subject to courtesy information to the author (address is given in the article/chapter).
- If you are an STM Signatory or your work will be published by an STM Signatory and you are requesting to reuse figures/tables/illustrations or single text extracts, permission is granted according to STM Permissions Guidelines: http://www.stm-assoc.org/permissions-guidelines/
For any electronic use not mentioned in the Guidelines, please contact Springer at permissions.springer@spi-global.com. If you request to reuse more content than stipulated in the STM Permissions Guidelines, you will be charged a permission fee for the excess content.
Permission is valid upon payment of the fee as indicated in the licensing process. If permission is granted free of charge on this occasion, that does not prejudice any rights we might have to charge for reproduction of our copyrighted material in the future.
-If your request is for reuse in a Thesis, permission is granted free of charge under the

IEEE COPYRIGHT FORM

To ensure uniformity of treatment among all contributors, other forms may not be substituted for this form, nor may any wording of the form be changed. This form is intended for original material submitted to the IEEE and must accompany any such material in order to be published by the IEEE. Please read the form carefully and keep a copy for your files.

TITLE OF PAPER/ARTICLE/REPORT/PRESENTATION/SPEECH (hereinafter, "the Work"):

Deep and Broad Learning on Content-aware POI Recommendation

COMPLETE LIST OF AUTHORS:

Fengjiao Wang, Yongzhi Qu, Lei Zheng, Chun-Ta Lu, Philip S. Yu

IEEE PUBLICATION TITLE (Journal, Magazine, Conference, Book):

The 3rd IEEE International Conference on Collaboration and Internet Computing (IEEE CIC 2017)

Copyright Transfer

The undersigned hereby assigns to the Institute of Electrical and Electronics Engineers, Incorporated (the "IEEE") all rights under copyright that may exist in and to the above Work, and any revised or expanded derivative works submitted to the IEEE by the undersigned based on the Work. The undersigned hereby warrants that the Work is original and that he/she is the author of the Work; to the extent the Work incorporates text passages, figures, data or other material from the works of others, the undersigned has obtained any necessary permissions. **See reverse side for Retained Rights and other Terms and Conditions.**

Author Responsibilities

The IEEE distributes its technical publications throughout the world and wants to ensure that the material submitted to its publications is properly available to the readership of those publications. Authors must ensure that their Work meets the requirements of IEEE Policy 6.3, including provisions covering originality, authorship, author responsibilities and author misconduct. The full policy may be viewed at **http://www.ieee.org/about/whatis/policies/p6-3.html**. Authors are advised especially of IEEE Policy 6.3.1A(l): "It is the responsibility of the authors, not the IEEE, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it."

General Terms

- The undersigned represents that he/she has the power and authority to make and execute this assignment.
- The undersigned agrees to indemnify and hold harmless the IEEE from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.
- In the event the above work is not accepted and published by the IEEE or is withdrawn by the author(s) before acceptance by the IEEE, the foregoing copyright transfer shall become null and void and all materials embodying the Work submitted to the IEEE will be destroyed.
- For jointly authored Works, all joint authors should sign, or one of the authors should sign as authorized agent for the others.

(1) Wang Jiao ______________________ **11/09/2017** ______________

Author/Authorized Agent for Joint Authors **Date**

U.S. Government Employee Certification (where applicable)

This will certify that all authors of the Work are U.S. government employees and prepared the Work on a subject within the scope of their official duties. As such, the Work is not subject to U.S. copyright protection.

(2)______________________ ______________________

Authorized Signature **Date**

(Authors who are U.S. government employees should also sign signature line (1) above to enable the IEEE to claim and protect its copyright in international jurisdictions.)

Crown Copyright Certification (where applicable)

This will certify that all authors of the Work are employees of the British or British Commonwealth Government and prepared the Work in connection with their official duties. As such, the Work is subject to Crown Copyright and is not assigned to the IEEE as set forth in the first sentence of the Copyright Transfer Section above. The undersigned acknowledges, however, that the IEEE has the right to publish, distribute and reprint the Work in all forms and media.

(3)______________________ ______________________

Authorized Signature **Date**

(Authors who are British or British Commonwealth Government employees should also sign line (1) above to indicate their acceptance of all terms other than the copyright transfer.)

IEEE COPYRIGHT FORM *(continued)*

RETAINED RIGHTS/TERMS AND CONDITIONS

1. Authors/employers retain all proprietary rights in any process, procedure, or article of manufacture described in the Work.

2. Authors/employers may reproduce or authorize others to reproduce the Work, material extracted verbatim from the Work, or derivative works for the author's personal use or for company use, provided that the source and the IEEE copyright notice are indicated, the copies are not used in any way that implies IEEE endorsement of a product or service of any employer, and the copies themselves are not offered for sale.

3. Authors/employers may make limited distribution of all or portions of the Work prior to publication if they inform the IEEE in advance of the nature and extent of such limited distribution.

4. In the case of a Work performed under a U.S. Government contract or grant, the IEEE recognizes that the U.S. Government has royalty-free permission to reproduce all or portions of the Work, and to authorize others to do so, for official U.S. Government purposes only, if the contract/grant so requires.

5. For all uses not covered by items 2, 3, and 4, authors/employers must request permission from the IEEE Intellectual Property Rights office to reproduce or authorize the reproduction of the Work or material extracted verbatim from the Work, including figures and tables.

6. Although authors are permitted to re-use all or portions of the Work in other works, this does not include granting third-party requests for reprinting, republishing, or other types of re-use. The IEEE Intellectual Property Rights office must handle all such third-party requests.

INFORMATION FOR AUTHORS

IEEE Copyright Ownership

It is the formal policy of the IEEE to own the copyrights to all copyrightable material in its technical publications and to the individual contributions contained therein, in order to protect the interests of the IEEE, its authors and their employers, and, at the same time, to facilitate the appropriate re-use of this material by others. The IEEE distributes its technical publications throughout the world and does so by various means such as hard copy, microfiche, microfilm, and electronic media. It also abstracts and may translate its publications, and articles contained therein, for inclusion in various compendiums, collective works, databases and similar publications.

Author/Employer Rights

If you are employed and prepared the Work on a subject within the scope of your employment, the copyright in the Work belongs to your employer as a work-for-hire. In that case, the IEEE assumes that when you sign this Form, you are authorized to do so by your employer and that your employer has consented to the transfer of copyright, to the representation and warranty of publication rights, and to all other terms and conditions of this Form. If such authorization and consent has not been given to you, an authorized representative of your employer should sign this Form as the Author.

Reprint/Republication Policy

The IEEE requires that the consent of the first-named author and employer be sought as a condition to granting reprint or republication rights to others or for permitting use of a Work for promotion or marketing purposes.

PLEASE DIRECT ALL QUESTIONS ABOUT THIS FORM TO:
Manager, IEEE Intellectual Property Rights Office, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1131.
Telephone +1 (732) 562-3966

CITED LITERATURE

1. Wang, F., Wang, G., Lin, S., and Yu, P. S.: Concurrent goal-oriented co-clustering generation in social networks. In Proceedings of the 9th IEEE International Conference on Semantic Computing, ICSC 2015, Anaheim, CA, USA, February 7-9, 2015.

2. Wang, F., Lin, S., and Yu, P. S.: Collaborative co-clustering across multiple social media. In IEEE 17th International Conference on Mobile Data Management, MDM 2016, Porto, Portugal, June 13-16, 2016.

3. Wang, F., Lu, C., Qu, Y., and Yu, P. S.: Collective geographical embedding for geolocating social network users. In Advances in Knowledge Discovery and Data Mining - 21st Pacific-Asia Conference, PAKDD 2017, Jeju, South Korea, May 23-26, 2017, Proceedings, Part I.

4. Wang, F., Qu, Y., Zheng, L., Lu, C.-T., and Yu, P. S.: Deep and broad learning on content-aware poi recommendation. In IEEE 3rd International Conference on Collaboration and Internet Computing (IEEE CIC '17).

5. Dhillon, I. S.: Co-clustering documents and words using bipartite spectral graph partitioning. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '01, pages 269–274, New York, NY, USA, 2001. ACM.

6. Berjani, B. and Strufe, T.: A recommendation system for spots in location-based online social networks. In SNS, page 4. ACM, 2011.

7. Agarwal, D. and Merugu, S.: Predictive discrete latent factor models for large scale dyadic data. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '07, pages 26–35, New York, NY, USA, 2007. ACM.

8. Shi, X., Fan, W., and Yu, P. S.: Efficient semi-supervised spectral co-clustering with constraints. In Proceedings of the 2010 IEEE International Conference on Data Mining, ICDM '10, pages 1043–1048, Washington, DC, USA, 2010. IEEE Computer Society.

9. Edelman, A., Arias, T. A., and Smith, S. T.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl., 20(2):303–353, April 1999.

10. Bach, F. R. and Jordan, M. I.: Kernel independent component analysis. J. Mach. Learn. Res., 3:1–48, March 2003.

11. Armijo, L.: Minimization of functions having lipschitz continuous irst partial derivatives. pages 1–3, 1966.

12. Cheng, Z., Caverlee, J., Lee, K., and Sui, D. Z.: Exploring millions of footprints in location sharing services. In ICWSM, eds. L. A. Adamic, R. A. Baeza-Yates, and S. Counts. The AAAI Press.

13. Wu, M.-L., Chang, C.-H., and Liu, R.-Z.: Co-clustering with augmented matrix. Appl. Intell., 39(1):153–164, 2013.

14. Dhillon, I. S., Mallela, S., and Modha, D. S.: Information-theoretic co-clustering. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '03, pages 89–98, New York, NY, USA, 2003. ACM.

15. Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., and Modha, D. S.: A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '04, pages 509–514, New York, NY, USA, 2004. ACM.

16. Long, B., Zhang, Z. M., and Yu, P. S.: Co-clustering by block value decomposition. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, KDD '05, pages 635–640, New York, NY, USA, 2005. ACM.

17. Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., and Modha, D. S.: A generalized maximum entropy approach to bregman co-clustering and matrix approximation. J. Mach. Learn. Res., 8:1919–1986, December 2007.

18. Shan, H. and Banerjee, A.: Bayesian co-clustering. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, ICDM '08, pages 530–539, Washington, DC, USA, 2008. IEEE Computer Society.

19. Kluger, Y., Basri, R., Chang, J. T., and Gerstein, M.: Spectral biclustering of microarray data: Coclustering genes and conditions. Genome Res., 13(4):703–716, April 2003.

20. Cho, H., Dhillon, I. S., Guan, Y., and Sra, S.: Minimum sum-squared residue co-clustering of gene expression data. In Proceedings of the Fourth SIAM International Conference on Data Mining, ICDM '04, pages 114–125, 2004.

21. Rohwer, R. and Freitag, D.: Towards full automation of lexicon construction. In HLT-NAACL 2004: Workshop on Computational Lexical Semantics, eds. D. Moldovan and R. Girju, pages 9–16, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

22. Freitag, D.: Toward unsupervised whole-corpus tagging. In Proceedings of the 20th international conference on Computational Linguistics, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

23. Gao, B., Liu, T.-Y., Zheng, X., Cheng, Q.-S., and Ma, W.-Y.: Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, KDD '05, pages 41–50, New York, NY, USA, 2005. ACM.

24. Hiroya, T. and Yuji, M.: Co-clustering for text categorization. Transactions of Information Processing Society of Japan, 44(2):443–450, 2003.

25. Niu, D., Dy, J. G., and Jordan, M. I.: Multiple non-redundant spectral clustering views. In ICML, pages 831–838, 2010.

26. Caruana, R., Elhawary, M. F., Nguyen, N., and Smith, C.: Meta clustering. In ICDM, pages 107–118, 2006.

27. Yin, X., Han, J., and Yu, P. S.: Crossclus: User-guided multi-relational clustering. Data Min. Knowl. Discov., 15(3):321–348, December 2007.

28. Cheng, Z., Caverlee, J., Lee, K., and Sui, D. Z.: Exploring millions of footprints in location sharing services. In ICWSM, eds. L. A. Adamic, R. A. Baeza-Yates, and S. Counts. The AAAI Press, 2011.

29. Abhishek, K., Rai, P., and III, H. D.: Co-regularized multi-view spectral clustering. In NIPS, eds. J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, 2011.

30. Wang, F., Wang, G., Lin, S., and Philip, S. Y.: Concurrent goal-oriented co-clustering generation in social networks. In 2015 IEEE International Conference on Semantic Computing (ICSC). IEEE, 2015.

31. Song, Y., Pan, S., Liu, S., Wei, F., Zhou, M. X., and Qian, W.: Constrained co-clustering for textual documents. In Proc. Conf. Artificial Intelligence (AAAI), 2010.

32. Schifanella, C., Sapino, M. L., and Candan, K. S.: On context-aware co-clustering with metadata support. J. Intell. Inf. Syst., 38(1), 2012.

33. Wang, P., Domeniconi, C., Rangwala, H., and Laskey, K. B.: Feature enriched nonparametric bayesian co-clustering. In PAKDD '12, 2012.

34. Wu, M.-L., Chang, C.-H., and Liu, R.-Z.: Co-clustering with augmented data matrix. In DaWaK, eds. A. Cuzzocrea and U. Dayal, volume 6862 of Lecture Notes in Computer Science. Springer, 2011.

35. Greco, G., Guzzo, A., and Pontieri, L.: Coclustering multiple heterogeneous domains: Linear combinations and agreements. IEEE Trans. Knowl. Data Eng., 22(12), 2010.

36. Bickel, S. and Scheffer, T.: Multi-view clustering. In ICDM '04, pages 19–26, 2004.

37. Chaudhuri, K., Kakade, S. M., Livescu, K., and Sridharan, K.: Multi-view clustering via canonical correlation analysis. In ICML '09, 2009.

38. Gao, J., Han, J., Liu, J., and Wang, C.: Multi-view clustering via joint nonnegative matrix factorization. In SDM. SIAM, 2013.

39. He, X., Kan, M.-Y., Xie, P., and Chen, X.: Comment-based multi-view clustering of web 2.0 items. In Proceedings of the 23rd International Conference on World Wide Web, WWW '14, 2014.

40. Li, S., Jiang, Y., and Zhou, Z.: Partial multi-view clustering. In In Proceedings of the twenty-eighth AAAI conference on artificial intelligence, 2014.

41. Abhishek, K. and III, H. D.: A co-training approach for multi-view spectral clustering. In ICML, eds. L. Getoor and T. Scheffer. Omnipress, 2011.

42. de Sa, V. R.: Spectral clustering with two views. In ICML '05, 2005.

43. Tang, W., Lu, Z., and Dhillon, I. S.: Clustering with multiple graphs. In ICDM '09, 2009.

44. Wang, X., Qian, B., Ye, J., and Davidson, I.: Multi-objective multi-view spectral clustering via pareto optimization. In Proceedings of the 13th SIAM International Conference on Data Mining, May 2-4, 2013. Austin, Texas, USA., 2013.

45. Zhou, D. and Burges, C. J. C.: Spectral clustering and transductive learning with multiple views. In ICML '07, New York, NY, USA, 2007. ACM.

46. Cheng, W., Zhang, X., Guo, Z., Wu, Y., Sullivan, P. F., and Wang, W.: Flexible and robust co-regularized multi-domain graph clustering. In KDD '13, 2013.

47. Sindhwani, V., Niyogi, P., and Belkin, M.: A co-regularized approach to semi-supervised learning with multiple views. In Proceedings of the ICML Workshop on Learning with Multiple Views, 2005.

48. Bisson, G. and Grimal, C.: Co-clustering of multi-view datasets: A parallelizable approach. In ICDM, eds. M. J. Zaki, A. Siebes, J. X. Yu, B. Goethals, G. I. Webb, and X. Wu. IEEE Computer Society, 2012.

49. Zheng, Y., Capra, L., Wolfson, O., and Yang, H.: Urban computing: Concepts, methodologies, and applications. ACM Trans. Intell. Syst. Technol., 2014.

50. Wang, F., Lin, S., and Yu, P. S.: Collaborative co-clustering across multiple social media. MDM '16.

51. Zheng, Y.: Methodologies for cross-domain data fusion: An overview. IEEE Trans. Big Data, 2015.

52. Jurgens, D., Finethy, T., McCorriston, J., Xu, Y. T., and Ruths, D.: Geolocation prediction in twitter using social networks: A critical analysis and review of current practice. ICWSM '15.

53. Pontes, T., Magno, G., Vasconcelos, M., Gupta, A., Almeida, J., Kumaraguru, P., and Almeida, V.: Beware of what you share: Inferring home location in social networks. ICDMW '12.

54. Jr., C. A. D., Pappa, G. L., de Oliveira, D. R. R., and de Lima Arcanjo, F.: Inferring the location of twitter messages based on user relationships. T. GIS, pages 735–751, 2011.

55. Jurgens, D.: That's what friends are for: Inferring location in online social media platforms based on social relationships. ICWSM '13.

56. Ahmed, A., Hong, L., and Smola, A. J.: Hierarchical geographical modeling of user locations from social media posts. WWW '13.

57. Cha, M., Gwon, Y., and Kung, H. T.: Twitter geolocation and regional classification via sparse coding. ICWSM '15.

58. Li, R., Wang, S., Deng, H., Wang, R., and Chang, K. C.-C.: Towards social user profiling: Unified and discriminative influence model for inferring home locations. KDD '12.

59. Valkanas, G. and Gunopulos, D.: Location extraction from social networks with commodity software and online data. ICDMW '12.

60. Backstrom, L., Sun, E., and Marlow, C.: Find me if you can: Improving geographical prediction with social and spatial proximity. WWW '10.

61. Chang, S., Han, W., Tang, J., Qi, G.-J., Aggarwal, C. C., and Huang, T. S.: Heterogeneous network embedding via deep architectures. KDD '15.

62. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q.: Line: Large-scale information network embedding. WWW '15.

63. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q.: Pte: Predictive text embedding through large-scale heterogeneous text networks. KDD '15.

64. Belkin, M. and Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In NIPS '01.

65. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J.: Distributed representations of words and phrases and their compositionality. NIPS'13.

66. Zheng, Y.: Location-based social networks: Users. In Computing with Spatial Trajectories. 2011.

67. Kotzias, D., Lappas, T., and Gunopulos, D.: Addressing the sparsity of location information on twitter. EDBT/ICDT '14 Workshops.

68. Li, R., Wang, S., and Chang, K. C.: Multiple location profiling for users and relationships from social network and content. PVLDB, 2012.

69. Rahimi, A., Vu, D., Cohn, T., and Baldwin, T.: Exploiting text and network context for geolocation of social media users. HLT '15.

70. Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y. M., and Yuan, Q.: Personalized ranking metric embedding for next new poi recommendation. IJCAI'15.

71. Agarwal, A., Hosanagar, K., and Smith, M. D.: Location, location, location: An analysis of profitability of position in online advertising markets. JMR'11.

72. Lee, R. and Sumiya, K.: Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection. In SIGSPATIAL Workshop'10.

73. Sakaki, T., Okazaki, M., and Matsuo, Y.: Earthquake shakes twitter users: Real-time event detection by social sensors. In WWW'10.

74. Xia, C., Schwartz, R., Xie, K., Krebs, A., Langdon, A., Ting, J., and Naaman, M.: Citybeat: Real-time social media visualization of hyper-local city data. In WWW'14.

75. Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In WWW'10.

76. Yang, D., Zhang, D., Yu, Z., and Wang, Z.: A sentiment-enhanced personalized location recommendation system. In HT'13.

77. Liu, B. and Xiong, H.: Point-of-interest recommendation in location based social networks with topic and location awareness. In SDM'13.

78. Hu, B. and Ester, M.: Spatial topic modeling in online social media for location recommendation. In RecSys'13.

79. Wallach, H. M.: Topic modeling: Beyond bag-of-words. In ICML'06.

80. Zhang, Y., Dai, H., Xu, C., Feng, J., Wang, T., Bian, J., Wang, B., and Liu, T.-Y.: Sequential click prediction for sponsored search with recurrent neural networks. In AAAI'14.

81. Liu, Q., Wu, S., Wang, L., and Tan, T.: Predicting the next location: A recurrent model with spatial and temporal contexts. In AAAI'16.

82. Cohen, W. W., Schapire, R. E., and Singer, Y.: Learning to order things. J. Artif. Int. Res., 1999.

83. Salakhutdinov, R. and Mnih, A.: Probabilistic matrix factorization. In NIPS'08.

84. Nair, V. and Hinton, G. E.: Rectified linear units improve restricted boltzmann machines. In ICML'10.

85. Krizhevsky, A., Sutskever, I., and Hinton, G. E.: Imagenet classification with deep convolutional neural networks. In NIPS'12.

86. Hochreiter, S. and Schmidhuber, J.: Long short-term memory. Neural Comput., 1997.

87. Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In UAI'09.

88. LeCun, Y., Bottou, L., Orr, G. B., and Müller, K.-R.: Effiicient backprop. In Neural Networks: Tricks of the Trade, 1998.

89. Yang, D., Zhang, D., Chen, L., and Qu, B.: Nationtelescope: Monitoring and visualizing large-scale collective behavior in lbsns. J. Network and Computer Applications, 2015.

90. Rendle, S.: Factorization machines with libFM. ACM Trans. Intell. Syst. Technol., 2012.

91. Wang, H., Wang, N., and Yeung, D.-Y.: Collaborative deep learning for recommender systems. In KDD'15.

92. Lian, D., Zhao, C., Xie, X., Sun, G., Chen, E., and Rui, Y.: Geomf: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In KDD'14.

93. Liu, B., Fu, Y., Yao, Z., and Xiong, H.: Learning geographical preferences for point-of-interest recommendation. In KDD'13.

94. Li, X., Cong, G., Li, X.-L., Pham, T.-A. N., and Krishnaswamy, S.: Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In SIGIR'15.

95. Zhang, J.-D., Chow, C.-Y., and Li, Y.: Lore: Exploiting sequential influence for location recommendations. In SIGSPATIAL'14.

96. Cheng, C., Yang, H., Lyu, M. R., and King, I.: Where you like to go next: Successive point-of-interest recommendation. In IJCAI'13.

97. Liu, X., Liu, Y., and Li, X.: Exploring the context of locations for personalized location recommendations. In IJCAI'16.

98. Zhao, S., Zhao, T., King, I., and Lyu, M. R.: Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation. In WWW Companion'17.

99. Gao, H., Tang, J., Hu, X., and Liu, H.: Content-aware point of interest recommendation on location-based social networks. In AAAI'15.

100. Yin, H., Sun, Y., Cui, B., Hu, Z., and Chen, L.: Lcars: A location-content-aware recommender system. In KDD'13.

101. Wang, W., Yin, H., Sadiq, S. W., Chen, L., Xie, M., and Zhou, X.: SPORE: A sequential personalized spatial item recommender system. In ICDE'16.

102. Graves, A.: Generating sequences with recurrent neural networks. CoRR'13.

103. Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In A Field Guide to Dynamical Recurrent Neural Networks. 2001.

104. Zheng, L., Noroozi, V., and Yu, P. S.: Joint deep modeling of users and items using reviews for recommendation. In WSDM'17.

105. Qiu, G.: Image and feature co-clustering. In Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4 - Volume 04, ICPR '04, pages 991–994, Washington, DC, USA, 2004. IEEE Computer Society.

106. Bisson, G. and Grimal, C.: Co-clustering of multi-view datasets: A parallelizable approach. In Proceedings of the 2012 IEEE International Conference on Data Mining, ICDM '12, pages 828–833, 2012.

107. Gao, B., Liu, T.-Y., and Ma, W.-Y.: Star-structured high-order heterogeneous data co-clustering based on consistent information theory. In Proceedings of the Sixth International Conference on Data Mining, ICDM '06, pages 880–884, Washington, DC, USA, 2006. IEEE Computer Society.

108. Gretton, A., Bousquet, O., Smola, A., and Schlkopf, B.: Measuring statistical dependence with hilbert-schmidt norms. In PROCEEDINGS ALGORITHMIC LEARNING THEORY, pages 63–77. Springer-Verlag, 2005.

109. Chow, C.-Y., Bao, J., and Mokbel, M. F.: Towards location-based social networking services. In Proceedings of the 2Nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, LBSN '10, pages 31–38, New York, NY, USA, 2010. ACM.

110. Horozov, T., Narasimhan, N., and Vasudevan, V.: Using location for personalized poi recommendations in mobile environments. In Proceedings of the International Symposium on Applications on Internet, SAINT '06, pages 124–129, Washington, DC, USA, 2006. IEEE Computer Society.

111. Levandoski, J. J., Sarwat, M., Eldawy, A., and Mokbel, M. F.: Lars: A location-aware recommender system. In Proceedings of the 2012 IEEE 28th International Conference on Data Engineering, ICDE '12, pages 450–461, Washington, DC, USA, 2012. IEEE Computer Society.

112. Takeuchi, Y. and Sugimoto, M.: Cityvoyager: An outdoor recommendation system based on user location history. In Proceedings of the Third International Conference on Ubiquitous Intelligence and Computing, UIC'06, pages 625–636, Berlin, Heidelberg, 2006. Springer-Verlag.

113. Ye, M., Yin, P., and Lee, W.-C.: Location recommendation for location-based social networks. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '10, pages 458–461, New York, NY, USA, 2010. ACM.

114. Ye, M., Yin, P., Lee, W.-C., and Lee, D.-L.: Exploiting geographical influence for collaborative point-of-interest recommendation. In Proceedings of the 34th International

ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11, pages 325–334, New York, NY, USA, 2011. ACM.

115. Zheng, V. W., Zheng, Y., Xie, X., and Yang, Q.: Towards mobile intelligence: Learning from gps history data for collaborative recommendation. Artif. Intell., 184-185:17–37, 2012.

116. Bowman, M., Debray, S. K., and Peterson, L. L.: Reasoning about naming systems. ACM Trans. Program. Lang. Syst., 15(5):795–825, November 1993.

117. Clark, M.: Post congress tristesse. In TeX90 Conference Proceedings, pages 84–89. TeX Users Group, March 1991.

118. Herlihy, M.: A methodology for implementing highly concurrent data objects. ACM Trans. Program. Lang. Syst., 15(5):745–770, November 1993.

119. Salas, S. and Hille, E.: Calculus: One and Several Variable. New York, John Wiley and Sons, 1978.

120. Assam, R. and Seidl, T.: Context-based location clustering and prediction using conditional random fields. In Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia, MUM '14, 2014.

121. Pianese, F., An, X., Kawsar, F., and Ishizuka, H.: Discovering and predicting user routines by differential analysis of social network traces. WoWMoM '13.

122. Cheng, Z., Caverlee, J., and Lee, K.: You are where you tweet: A content-based approach to geo-locating twitter users. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10, pages 759–768, 2010.

123. Kurashima, T., Iwata, T., Hoshide, T., Takaya, N., and Fujimura, K.: Geo topic model: Joint modeling of user's activity area and interests for location recommendation. In WSDM'13.

124. Wang, H., Shi, X., and Yeung, D.: Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks. In NIPS'16.

125. Oord, A. v. d., Dieleman, S., and Schrauwen, B.: Deep content-based music recommendation. In NIPS'13.

126. Cho, E., Myers, S. A., and Leskovec, J.: Friendship and mobility: User movement in location-based social networks. In KDD'11.

127. Gonzalez, M. C., Hidalgo, C. A., and Barabasi, A.-L.: Understanding individual human mobility patterns. Nature'08.

128. Xie, M., Yin, H., Wang, H., Xu, F., Chen, W., and Wang, S.: Learning graph-based poi embedding for location-based recommendation. In CIKM'16.

129. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J.: Distributed representations of words and phrases and their compositionality. In NIPS'13.

130. Lindqvist, J., Cranshaw, J., Wiese, J., Hong, J., and Zimmerman, J.: I'm the mayor of my house: Examining why people use foursquare - a social-driven location sharing application. In CHI'11.

131. Johnson, R. and Zhang, T.: Effective use of word order for text categorization with convolutional neural networks. In HLT-NAACL'15.

132. Pascanu, R., Mikolov, T., and Bengio, Y.: On the difficulty of training recurrent neural networks. In ICML'13.

133. Sutskever, I., Vinyals, O., and Le, Q. V.: Sequence to sequence learning with neural networks. In NIPS'14.

134. Kim, Y.: Convolutional neural networks for sentence classification. In EMNLP'14.

135. Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S.: Recurrent neural network based language model. In INTERSPEECH'10.

VITA

Name: Fengjiao Wang

EDUCATION:

B.S. in Information and Computational Science, Beihang University, 2010.

PUBLICATIONS:

- Fengjiao Wang, Yongzhi Qu, Lei Zheng, Chun-Ta Lu, Philip S. Yu. Deep and Broad Learning on Content-aware POI Recommendation. In *The 3rd International Conference on Collaboration and Internet Computing (IEEE CIC 17)*, 2017.
- Fengjiao Wang, Chun-Ta Lu, Yongzhi Qu, Philip S. Yu. Collective Geographical Embedding for Geolocating Social Network Users. In *The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD '17)*, 2017.
- Xiao Pan, Jiawei Zhang, Fengjiao Wang, Philip S. Yu. DistSD: Distance-based Social Discovery with Personalized Posterior Screening. In *IEEE International Conference on Big Data (BigData '16)*, 2016.
- Fengjiao Wang, Shuyang Lin, Philip S. Yu. Collaborative Co-clustering across Multiple Social Media. In *IEEE International Conference on Mobile Data Management (MDM '16)*, 2016.
- Fengjiao Wang, Guan Wang, Shuyang Lin, Philip S. Yu. Concurrent Goal-oriented Co-clustering Generation in Social Networks. In *IEEE International Conference on Semantic Computing (ICSC '15)*, 2015.

- Fengjiao Wang, Guan Wang, Shuyang Lin, Philip S. Yu. Why Checkins: Exploring User Motivation on Location Based Social Networks. In *IEEE International Conference on Data Mining (ICDM '14)*, 2014 (*Workshop*).
- Shuyang Lin, Qingbo Hu, Fengjiao Wang, Philip S. Yu. Steering Information Diffusion Dynamically against User Attention Limitation. In *IEEE International Conference on Data Mining (ICDM '14)*, 2014.
- Yilin Shen, Fengjiao Wang, Hongxia Jin. Defending against User Identity Linkage Attack across Multiple Online Social Networks. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion (WWW Companion '14)*, 2014.
- Ning Yang, Xiangnan Kong, Fengjiao Wang, Philip S. Yu. When and Where: Predicting Human Movements Based on Social Spatial-Temporal Events. In *Proceedings of 2014 SIAM International Conference on Data Mining (SDM '14)*, 2014.
- Shuyang Lin, Fengjiao Wang, Qingbo Hu, Philip S. Yu. Extracting social events for learning better information diffusion models,. In *Proceedings of the Nineteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD '13)*, 2013.
- Chenhao Tan, Jie Tang, Jimeng Sun, Quan Lin, Fengjiao Wang. Social Action Tracking via Noise Tolerant Time-varying Factor Graphs,. In *Proceedings of the Sixteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD '10)*, 2010.