Collective Geographical Embedding for Geolocating Social Network Users

Fengjiao Wang[1], Chun-Ta Lu[1], Yongzhi Qu[2], and Philip S. Yu[1]

[1] Univeristy of Illinois at Chicago, Chicago, USA
{fwang27,clu29,psyu}@uic.edu
[2] Wuhan University of Technology, China
quwong@whut.edu.cn

Abstract. Inferring the physical locations of social network users is one of the core tasks in many online services, such as targeted advertisement, recommending local events, and urban computing. In this paper, we introduce the Collective Geographical Embedding (CGE) algorithm to embed multiple information sources into a low dimensional space, such that the distance in the embedding space reflects the physical distance in the real world. To achieve this, we introduced an embedding method with a location affinity matrix as a constraint for heterogeneous user network. The experiments demonstrate that the proposed algorithm not only outperforms traditional user geolocation prediction algorithms by collectively extracting relations hidden in the heterogeneous user network, but also outperforms state-of-the-art embedding algorithms by appropriately casting geographical information of check-in.

Keywords: Geolocation, Geometrical embedding, Geometric regularization

1 Introduction

Urban computing has attracted many research attentions [22]. Cross-domain data can be fused together to aid this task [19, 21]. 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 [9, 14]. 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 [7, 8] or the content of the online footprints [1, 4]. 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 their location information. Research in Twitter suggest that only 16% of users registered city level locations in their profiles [12], and the percentage of tweets with geographical coordinates was

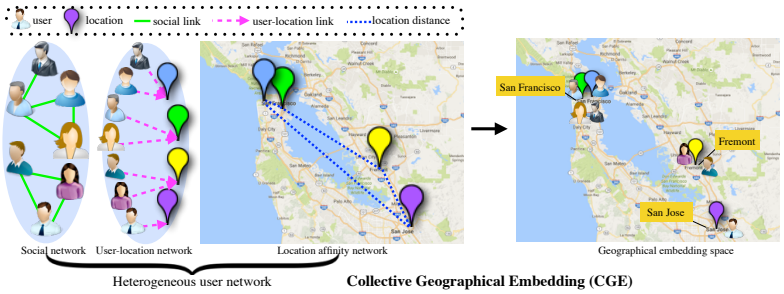


Fig. 1: Example of learning the geographical embedding space from heterogeneous networks.

merely 1% [18]. (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 [7, 8] and probability estimation based methods [2].(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 [1] require a huge amount of computational resources and may not be applicable to large-scale datasets.

Recently, network embedding techniques [5, 16, 17] 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 [5, 17] 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 (1) 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.

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}*.*

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.

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 $\boldsymbol{v}_i \in \mathbb{R}^d$, where d is the dimension of the embedding vector. Inspired by [17], 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 $p_j \in \mathcal{P}$:

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

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:

$$\mathcal{J}_{up} = -\sum_{e_{ij} \in \mathcal{E}_{up}} w_{ij} \log P(p_j|u_i) \tag{2}$$

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:

$$\mathcal{J}_{uu} = -\sum_{e_{ij} \in \mathcal{E}_{uu}} w_{ij} \log P(u_j|u_i) \tag{3}$$

By jointly learning $\{\boldsymbol{u}_i\}_{i=1,\ldots,N}$ and $\{\boldsymbol{p}_j\}_{j=1,\ldots,M}$ that minimize the objectives Eq. (2) and Eq. (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.

3.2 Collective Geometrical Embedding

According to the local invariance assumption [3], if two samples p_i, p_j are close in the intrinsic geometric with regard to the data distribution, then their embeddings $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ 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:

$$\mathcal{R}(\mathbf{P}) = \sum_{i,j=1}^{M} w_{ij}(\boldsymbol{p}_i - \boldsymbol{p}_j)^2 \tag{4}$$

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. (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:

$$\mathcal{R}(\mathbf{P}) = \frac{1}{2}\sum_{i,j=1}^{M} w_{ij}(\boldsymbol{p}_i - \boldsymbol{p}_j)^2 = \frac{1}{2}Tr(\mathbf{P}^T(\mathbf{D} - \mathbf{S})\mathbf{P}) = \frac{1}{2}Tr(\mathbf{P}^T\mathbf{L}\mathbf{P}) \tag{5}$$

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

$$\min_{\mathbf{U},\mathbf{P}} \mathcal{J} = \mathcal{J}_{uu} + \mathcal{J}_{up} + \lambda\mathcal{R}(\mathbf{P}) \tag{6}$$

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. (6). The same strategy has also been applied in literature [17], 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 [13] 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}:

$$\log\sigma(\boldsymbol{p}_j^T \cdot \boldsymbol{u}_i) + \sum_{u=1}^{k} E_{p_n\sim P_n(p)}[\log\sigma(-\boldsymbol{p}_n^T \cdot \boldsymbol{u}_i)] \tag{7}$$

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,

Algorithm 1: 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: user embedding $\boldsymbol{u}$, location embedding $\boldsymbol{p}$;
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 $\boldsymbol{p}_i$ using the partial derivative in Eq. 8.
end

then force two vectors close to each other. The second term shows after sampling negative links from whole sets of vertices, force two vectors $\boldsymbol{u}_i$ and $\boldsymbol{p}_n$ 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 [13], where o_i is the out-degree of vertex u_i. For the detailed optimization process, readers can refer to [16]. 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 $\boldsymbol{p}_i$ by gradient descent. The gradient of $\mathcal{R}(\mathbf{P})$ w.r.t. $\boldsymbol{p}_i$ can be derived as follows:

$$\frac{\partial \mathcal{R}(\mathbf{P})}{\partial \mathbf{p}_i} = \sum_j w_{ij}(\mathbf{p}_i - \mathbf{p}_j) = (\sum_j w_{ij} - w_{ii})\mathbf{p}_i - \sum_{j \neq i} w_{ij}\mathbf{p}_j = [(\mathbf{D}-\mathbf{W})\mathbf{P}]_{i*} = [\mathbf{L}\mathbf{P}]_{i*}, \tag{8}$$

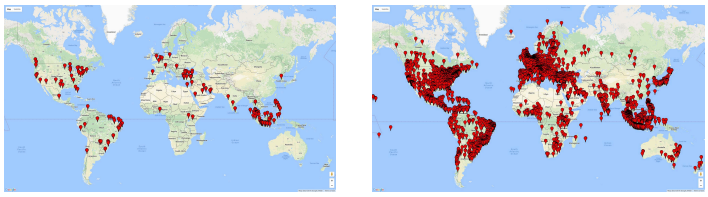
where $[\cdot]_{i*}$ means the i-th row of the given matrix.

The detailed process of the proposed algorithm is summarized in Algorithm 1. After obtaining the geometric embeddings of users and locations, we can train any classifier (e.g., SVM or 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.

4 Experiments

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 1. 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



(a) Foursquare (b) Twitter

Fig. 2: Distribution of users' locations in Foursquare and Twitter networks.

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 Fig. 2. Instead of only focusing on users lived in the US, we are tackling users globally, which creates more challenge for the learning task.

Table 1: Datasets

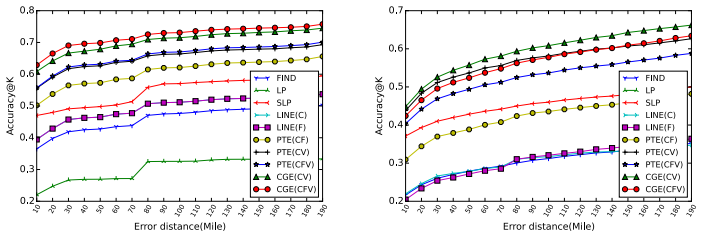
Dataset	users	locations	social links	user-location links
Foursquare	15,799	141,444	38,197	212,588
Twitter	25,355	403,770	156,060	564,298

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

1. **FIND** [2] selects the location that maximizes the probability of friendships given the distance between the location candidates and the friends' home locations.
2. **LP** [7] 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** [8] 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** [16] embeds a homogeneous network into a low dimensional space.
5. **PTE** [17] 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 [2]. 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 [8]. For all the embedding algorithms (LINE, PTE, and CGE), the embedding dimensionality is set to 100. We tried dimensionalities in the range

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



(a) Accuracy@k on Foursquare (b) Accuracy@k on Twitter

Fig. 3: Performance comparison on Foursquare and Twitter datasets

$[50, 200]$ and found that 100 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 [17]. 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 [9]. Higher AUC scores indicate better performance.

4.2 Quantitative results

Fig. 3 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 Fig. 3a, 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

Table 2: 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.

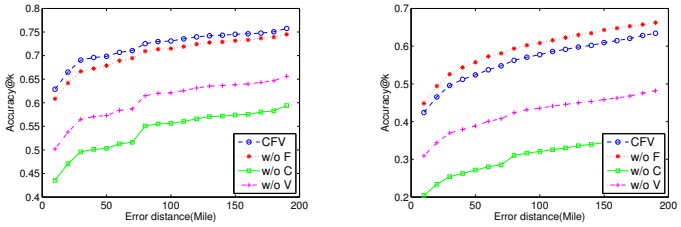
	Foursquare		Twitter	
	AED ↓	AUC ↑	AED ↓	AUC ↑
LP	2526.21 ± 34.05	45.52% ± 0.37%	4924.64 ± 18.24	19.30% ± 0.12%
SLP	1673.31 ± 0.73	61.21% ± 0.03%	2172.99 ± 2.40	53.21% ± 0.04%
FIND	1805.88 ± 28.25	57.41% ± 0.39%	2647.07 ± 16.84	42.53% ± 0.20%
LINE(C)	2018.94 ± 30.15	58.60% ± 0.28%	2759.46 ± 20.62	41.92% ± 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%

datasets, as shown in Figs. 3a and 3b. With exactly the same amount of information, the proposed CGE always outperforms PTE for a variety of error distance k. For example, in Fig. 3a, with user-location network and location affinity network, CGE(CV) correctly predicts 61% of users' home locations within 10 miles, while PTE(CV) correctly predicts 56% of users' home locations within the same distance. These results indicate the robustness of the proposed CGE algorithm.

Table 2 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 [7]. 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.

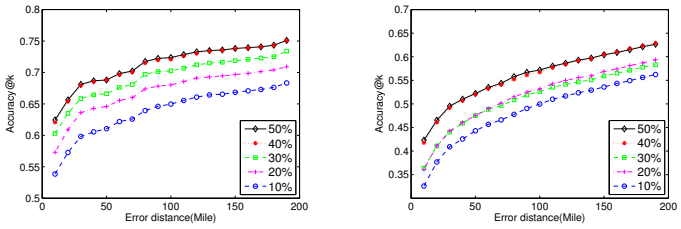
To evaluate the contribution of different sub-networks, we compare the results using partial information with the results using complete information. The comparisons are performed using CGE algorithm on both datasets. As can be seen in Fig. 4a, without user-location network (green line), the performance 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 Fig. 4b, 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 vectors as additional training data to balance training samples across different settings. As can be seen in Fig. 5, when the size of the training users decreases from 50% to 20%, accuracy@k only drops around 5%.



(a) Foursquare (b) Twitter

Fig. 4: Performance contribution of sub-networks. "w/o" means without certain sub-network.



(a) Foursquare (b) Twitter

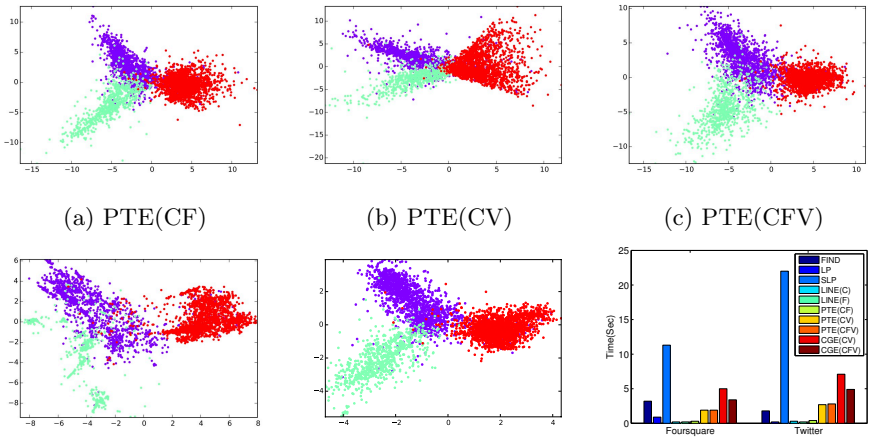
Fig. 5: Performance comparison with varied training size.

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 Fig. 6. 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 Fig. 6e, 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 in Fig. 6f. The run time of CGE algorithms are modestly longer compared with other embedding methods, but provides the best prediction performance.

5 Related Work

Location Prediction: Works on identifying users' home locations [20] can be roughly divided into two categories based on the information used. One category of related works focus on extracting text information [1, 4] 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 [2, 7, 8], where they rely on the assumption that tie strength is a strong indicator of users' home locations. [2] aims to predict the location of an individual by leveraging geographic and social relationships in the Facebook network. [9] reviews most recent network-based approaches, and proposes two new metrics on comparison of different approaches. [14] studies the problem of



(a) PTE(CF) (b) PTE(CV) (c) PTE(CFV)

(d) CGE(CV) (e) CGE(CFV) (f) Running time

Fig. 6: Visualization of users reside in three different countries (Blue: US, Green: Brazil, Red: Malaysia) in Foursquare. Running time comparison (f).

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 [10, 11, 15] consider text and network information simultaneously. [11] propose an algorithm derived from a generative model. [10, 15] 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.

Network Embedding: Recently, network embedding technique ([5, 6, 16, 17]) 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.

6 Conclusion

This paper 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.

Acknowledgments. This work is supported in part by NSF through grants IIS-1526499, and CNS-1626432, and NSFC 61672313. Yongzhi Qu would like to acknowledge national natural science foundation of China (NSFC 51505353).

References

1. Ahmed, A., Hong, L., Smola, A.J.: Hierarchical geographical modeling of user locations from social media posts. WWW '13
2. Backstrom, L., Sun, E., Marlow, C.: Find me if you can: Improving geographical prediction with social and spatial proximity. WWW '10
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: NIPS '01
4. Cha, M., Gwon, Y., Kung, H.T.: Twitter geolocation and regional classification via sparse coding. ICWSM '15
5. Chang, S., Han, W., Tang, J., Qi, G.J., Aggarwal, C.C., Huang, T.S.: Heterogeneous network embedding via deep architectures. KDD '15
6. Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y.M., Yuan, Q.: Personalized ranking metric embedding for next new poi recommendation. IJCAI'15
7. Jr., C.A.D., Pappa, G.L., de Oliveira, D.R.R., de Lima Arcanjo, F.: Inferring the location of twitter messages based on user relationships. T. GIS pp. 735–751 (2011)
8. Jurgens, D.: That's what friends are for: Inferring location in online social media platforms based on social relationships. ICWSM '13
9. Jurgens, D., Finethy, T., McCorriston, J., Xu, Y.T., Ruths, D.: Geolocation prediction in twitter using social networks: A critical analysis and review of current practice. ICWSM '15
10. Kotzias, D., Lappas, T., Gunopulos, D.: Addressing the sparsity of location information on twitter. EDBT/ICDT '14 Workshops
11. Li, R., Wang, S., Chang, K.C.: Multiple location profiling for users and relationships from social network and content. PVLDB (2012)
12. Li, R., Wang, S., Deng, H., Wang, R., Chang, K.C.C.: Towards social user profiling: Unified and discriminative influence model for inferring home locations. KDD '12
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. NIPS'13
14. Pontes, T., Magno, G., Vasconcelos, M., Gupta, A., Almeida, J., Kumaraguru, P., Almeida, V.: Beware of what you share: Inferring home location in social networks. ICDMW '12
15. Rahimi, A., Vu, D., Cohn, T., Baldwin, T.: Exploiting text and network context for geolocation of social media users. HLT '15
16. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. WWW '15
17. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Pte: Predictive text embedding through large-scale heterogeneous text networks. KDD '15
18. Valkanas, G., Gunopulos, D.: Location extraction from social networks with commodity software and online data. ICDMW '12
19. Wang, F., Lin, S., Yu, P.S.: Collaborative co-clustering across multiple social media. MDM '16
20. Zheng, Y.: Location-based social networks: Users. In: Computing with Spatial Trajectories (2011)
21. Zheng, Y.: Methodologies for cross-domain data fusion: An overview. IEEE Trans. Big Data (2015)
22. Zheng, Y., Capra, L., Wolfson, O., Yang, H.: Urban computing: Concepts, methodologies, and applications. ACM Trans. Intell. Syst. Technol. (2014)