Semantic Annotation for Places in LBSN through Graph Embedding

Yan Wang
School of Information, Central
University of Finance and Economics
dayanking@gmail.com

Zongxu Qin
School of Information, Central
University of Finance and Economics
qinzongxu@163.com

Jun Pang
FSTC& SnT
University of Luxembourg
jun.pang@uni.lu

Yang Zhang
CISPA, Saarland University
Saarland Informatics Campus
yang.zhang@cispa.saarland

Jin Xin
School of Information, Central
University of Finance and Economics
jinxin@cufe.edu.cn

ABSTRACT

With the prevalence of location-based social networks (LBSNs), automated semantic annotation for places plays a critical role in many LBSN-related applications. Although a line of research continues to enhance labeling accuracy, there is still a lot of room for improvement. The crucial problem is to find a high-quality representation for each place. In previous works, the representation is usually derived directly from observed patterns of places or indirectly from calculated proximity amongst places or their combination. In this paper, we also exploit the combination to represent places but present a novel semi-supervised learning framework based on graph embedding, called *Predictive Place Embedding* (PPE). For place proximity, PPE first learns user embeddings from a user-tag bipartite graph by minimizing supervised loss in order to preserve the similarity of users visiting analogous places. User similarity is then transformed into place proximity by optimizing each place embedding as the centroid of the vectors of its check-in users. Our underlying idea is that a place can be considered as a representative of all its visitors. For observed patterns, a place-temporal bipartite graph is used to further adjust place embeddings by reducing unsupervised loss. Extensive experiments on real large LBSNs show that PPE outperforms state-of-the-art methods significantly.

CCS CONCEPTS

• **Human-centered computing → Social networking sites**;

KEYWORDS

Semantic Tag; Deep Learning; Graph Representation

1 INTRODUCTION

With the rapid development of location acquisition and wireless communication technologies, a number of location-based social networks (LBSNs) have released on the Web recently, including Foursquare, Instagram, Facebook Places and Whrrl, where users can check in at places, e.g., stores, restaurants, bars, and share their related experiences in the real world [8].

LBSN has gained much attention from researchers [11] since they seamlessly integrate the virtual and physical environments. One challenge with LBSNs is to automatically annotate semantic tags for places without any labels. It is the basis of many LBSN-related applications, e.g., effective retrieval and recommendation of POIs. Previous works [11] show that approximate 30% places in Whrrl and Foursquare datasets lack any meaningful textual descriptions.

An intuitive way to predict semantic tags for places includes two steps. The first step is to extract (or select) features from basic attributes of users and places, such as demographic information and check-in time; the second step is to classify places based on extracted features with a classifier, e.g., SVM and logistical regression. However, feature extraction plays a more important role than classifier selection in this scenario since it is extremely difficult to find high-quality representation for differentiating all tags.

In previous analogous works, the representation is usually learned either directly from observed patterns of places and users [1, 2, 12] or from calculated relatedness amongst places (or their combination) [11].Although the learned representations work well to some extent, they are usually only good at distinguishing a part of tags, meanwhile the learning process could be very inefficient. For instance, the feature learning method in [1] needs to train a forest of boosted decision trees containing 100 decision trees to select effective features, and the representation of [11] merely works on recognizing few tags in the Instagram dataset.

Currently, with the introduction of graph embedding with deep learning [5–7, 10], novel representation has now become possible. With graph embedding, each vertex in a graph can be represented by a vector and the number of its dimension is specified by a user. In general, the vector of each vertex is learned from its local structure, e.g., who its neighbors are and how strong their connections are. Similar to earlier embedding techniques, graph embedding with deep learning also evolved from unsupervised learning model [5, 6] to semi-supervised learning model [7, 10].

Inspired by [7, 10], we propose a novel semi-supervised learning framework for place semantic annotations, called *Predictive Place Embedding* (PPE). PPE attempts to make use of *both labeled and unlabeled data* to learn a low-dimension place vector that reflects

CIKM'17, November 6–10, 2017, Singapore, Singapore.
© 2017 ACM. 978-1-4503-4918-5/17/11...$15.00
DOI: 10.1145/3132847.3133075

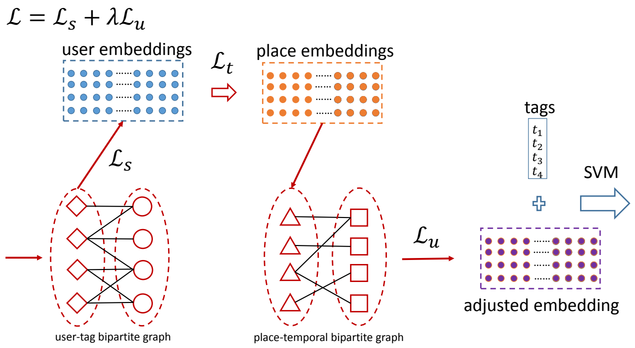

Figure 1: Overview of our PPE method. $\mathcal{L} = \mathcal{L}_s + \lambda \cdot \mathcal{L}_u$ is the loss function of our method. $\mathcal{L}_s/\mathcal{L}_u$ is the supervised/unsupervised loss function for generating/adjusting user/place embeddings, $\mathcal{L}_t$ are the loss functions for transforming user embeddings to place embeddings.

the proximity and the observed patterns of places simultaneously. Therefore, we propose to sequentially optimize two training objective functions: one is a supervised loss over a labeled *user-tag* bipartite graph for keeping the similarity of users, and the other is an unsupervised loss over an unlabeled *place-temporal* bipartite graph for conforming to a temporal pattern of places (i.e., the distributions of check-in time of places in 24-hour scale). Note that the first optimization is to preserve the similarity of users and hence we need to transform it into the proximity of places before the unsupervised learning. Intuitively, a place can be viewed as a typical “user" who represent all its check-in users. For instance, a research institute usually represents a group of scientists or researchers. Thus, it is reasonable to use the learned vectors of check-in users to generate the corresponding place vector. There are different ways to learn place vectors based on user vectors, here we simply optimize each place vector by minimizing the total Euclidean distance from the place vector to the rest of all visited user vectors.

Figure 1 describes the framework of PPE. A user-tag bipartite graph is generated from all check-in records first, then it is used to learn a low-dimension vector for each user by minimizing the supervised loss. With user vectors, the initial embedding of each place (with/without tag) is represented by the weighted average of the vectors of visited users, and it minimizes the total Euclidean distance among the place vector and its user vectors. With the initial place vectors, a place-temporal bipartite graph is exploited to further adjust place vectors by reducing the unsupervised loss. Finally, based on adjusted place vectors with tags, an SVM classifier for all tags is trained to predict the tags of unlabeled places.

In summary, we make the following contributions in this paper.

- We propose a novel semi-supervised learning framework for automated semantic annotation in LBSNs, which is a crucial prerequisite for many related applications.
- Empirical study shows that our PPE outperforms two state-of-the-art methods on large-scale datasets.
- We discover for the first time that the data distribution has an effect on the performance of representation methods.

2 RELATED WORK

There exists a line of research [1–3, 11, 12] addressing semantic annotation for places in LBSN or analogous contexts. The challenge is to find discriminative features for predicting the tags of places.

In general, existing representation approaches fall into three categories: manual feature selection, feature selection and feature extraction with machine learning techniques. For manual selection, most features are taken from various observed patterns of places, such as the duration of a visit and the neighbors of each place. In [1], the authors manually select 69 features to classify the categories of places based on two diary datasets, and their follow-up research work [2] selects additional features from the sequence of each individual’s visits. For feature selection, researchers take advantage of machine learning algorithms to learn a subset of promising features from a lot of candidate features. In [12], the Relief feature selection method and L1-regularized logistic regression are used to select 50 to 2000 most relevant features from 6k to 30k candidate features. For feature extraction, new features are learned from observed patterns. In [11], the authors provide a state-of-the-art representation method (called SAP) that utilizes *collective classification* techniques to generate a set of new features showing the probabilities of each place classified to a certain category.

Our PPE is mostly related to approaches on learning features based on graph embedding with deep learning techniques.For unsupervised learning model [5, 6], any given graph used for embedding does not contain any label information. The purpose of this model is to capture the context of each vertex. For supervised learning model [7, 10], there are more than one graphs used for embedding and at least one of the graphs contains label information.

3 MODEL FRAMEWORK

The loss function of our framework can be expressed as $\mathcal{L} = \mathcal{L}_s + \lambda \cdot \mathcal{L}_u$, where $\mathcal{L}_s$ ($\mathcal{L}_u$) is the supervised (unsupervised) loss of predicting the context of each user (place) in a user-tag (place-temporal) bipartite graph, and λ is a constant weighting factor that is used to control the impact of $\mathcal{L}_u$ on the total loss $\mathcal{L}$. In the rest of this section, we will introduce notations, $\mathcal{L}_s$ and $\mathcal{L}_u$ sequentially.

3.1 Notations

Given two sets U and P containing all the users and places, a user of U and a place of P are denoted by u and p, respectively. We assume that each user at least visits (checks in) one place and the total number of unique places visited by users is $|P|$. On the other hand, each place is at least visited by one user and the total number of unique users visiting places in P is $|U|$. Meanwhile, we use T to represent the set of all semantic tags and t to denote a tag in T. We assume that each place has at most one tag. In view of tags, the set P is divided into two subsets P' and $\bar{P}'$, i.e., the places in P' are labeled with one semantic tag while the places in $\bar{P'}$ are not. With the above notations, we proceed with the following two definitions:

Definition 1 (User-Tag Graph). *A user-tag graph, denoted as $G_{ut} = (U' \cup T, E_{ut})$, is a bipartite graph where U' is a subset of all users ($U' \subseteq U$). E_{ut} is the set of edges between the users and the tags. The weight w_{ik} between a user u_i ($1 \leq i \leq |U'|$) and a tag t_k ($1 \leq k \leq |T|$) is defined as $w_{ik} = \sum_{(p:t=t_k)} n_{pi}$ where n_{pi} is the total number of user u_i visiting place p annotated with the tag t_k.*

Definition 2 (Place-Temporal Graph). *A place-temporal graph, denoted as $G_{ph} = (P \cup H, E_{ph})$, is a bipartite graph where $H = \{1, 2, ..., 24\}$ represents the 24 hours of a day. The weight w_{jh} between a place p_j ($1 \leq i \leq |P|$) and an hour $h \in H$ is simply defined as the total number of times that users visit place p_j at hour h.*

Note that 1) each user/tag vertex should have at least one edge in a user-tag graph; and 2) if there exist users only visiting the places without tags, then they will not appear in any user-tag graph as user vertex and it is the reason why we have $U' \subseteq U$.

3.2 User embedding

For a user-tag graph, the conditional probability of a user vertex $u_i \in U'$ given by a tag vertex $t_k \in T$ is defined as:

$$p(u_i|t_k) = \frac{\exp(\vec{u_i}^T \cdot \vec{u_k})}{\sum_{i'=1}^{|U'|} \exp(\vec{u_{i'}}^T \cdot \vec{u_k})} \tag{1}$$

where $\vec{u_i}$ and $\vec{u_k}$ represent the vectors for vertex u_i and vertex t_k, respectively. The denominator is the sum of the product of the vector of each vertex in U' and the vector of t_k. Actually, Eq 1 defines a conditional distribution $p(\cdot|t_k)$ over all vertices in U'; meanwhile, its empirical distribution $\hat{p}(\cdot|t_k)$ can be obtained from the user-tag graph, i.e. $\hat{p}(u_i|t_k) = \frac{w_{ik}}{\sum_{i'=1}^{|U'|} w_{i'k}}$. In order to keep the two distributions of all tag vertices as close as possible, we attempt to minimize the supervised loss function shown below:

$$\mathcal{L}_s = \sum_{t_k \in T} \lambda_k \cdot KL(\hat{p}(\cdot|t_k)||p(\cdot|t_k)) \tag{2}$$

where $KL(\cdot||\cdot)$ is the KL-divergence between two distributions and $KL(\hat{p}(\cdot|t_k)||p(\cdot|t_k)) = \sum_{i=1}^{|U'|} \hat{p}(u_i|t_k) \log \frac{\hat{p}(u_i|t_k)}{p(u_i|t_k)}$, which is used to measure the distance of two distributions. The factor λ_k reflects the importance of vertex t_k in the bipartite graph and here we set $\lambda_k = \sum_{i=1}^{|U'|} w_{ik}$.

After putting the definition of KL divergence into Eq 2 and removing all constants, we have

$$\mathcal{L}_s = - \sum_{(i,k) \in E_{ut}} w_{ik} \cdot \log p(u_i|t_k). \tag{3}$$

The optimization of Eq 3 can be achieved by systematically learning all $\vec{u_i}$ and $\vec{u_k}$, and the learning method used is Stochastic Gradient Descent (SGD) with negative sampling and edge sampling. Both sampling techniques successfully overcome the deficiency of SGD and significantly improve the convergence rate of all vectors [6].

Notice that each learned user embedding $\vec{u_i}$ preserves the information of the probability distribution $p(u_i|\cdot)$ and user vertices with similar distributions over T are similar to each other. $p(u_i|\cdot)$ shows the probabilities of a user u_i visiting places with different tags and reflects the *visiting pattern* of a user. For example, Tom totally visited two kinds of places (i.e., *Office* and *Bar*) and for each kind the occurrence probabilities are 80% and 20%, respectively. Here we use the distribution $p(u_i|\cdot)$ to discriminate users and it means that $\vec{u_i}$ is the extracted feature of each user u_i.

3.3 Place embedding

In the literature [1, 2, 12], researchers illustrate that the information of users, such as demographic information and user-trace records, can be extracted as part of features to distinguish semantic tags of places. Here we further develop this idea and exploit user embeddings to directly generate place embeddings. The underlying idea is that two places can be distinguished by the behaviors of their visitors, following the intuition that the visiting patterns of users who often visit, for example, bars and users who frequently check in libraries could be very different.

Given a check-in set of a place p_j as $s = \{u^1, u^2, ..., u^n\}$ with $u^i \in U'$, $p_j \in P$, our objective is to minimize the following loss function to find an optimal vector for p_j,

$$\mathcal{L}_t = \sum_{i=1}^{n} l(\vec{u^i}, \vec{p_j}) = \sum_{i=1}^{n} \sum_{e=1}^{d} \frac{1}{2}(x_e^i - y_e^j)^2 \tag{4}$$

where the loss function $l(\cdot, \cdot)$ between a user embedding $\vec{u_i} = (x_1^i, \ldots, x_d^i)$ and a place embedding $\vec{p_j} = (y_1^j, \ldots, y_d^j)$ is specified as least square optimization, and d is the number of the dimension of the vector of each user/place vector.

In order to find $\vec{p_j}$ that minimizes Eq 4, for $\mathcal{L}_t$, we get its derivative with respect to $\vec{p_j}$ and simultaneously let it be 0, then we obtain a closed form solution as follows:

$$y_j^e = \frac{\sum_{i=1}^{n} x_i^e}{n}, (1 \leq e \leq d) \tag{5}$$

i.e., $\vec{p}$ is the weighted average of the vectors of its users.

3.4 Embedding adjustment

Now, the initial place embeddings are obtained from user embeddings based on their similarities. Actually, except user features, some observed patterns of places are usually used to describe places in previous works, e.g., the total number of check-ins. Here we follow this idea and exploit the distribution of check-in time in 24-hour scale to adjust all initial place embeddings. The information of the distributions of all places is preserved in a place-temporal graph as defined in Section 3.1.

Our unsupervised loss function based on a place-temporal graph is defined as follows:

$$\mathcal{L}_u = - \sum_{(i,h) \in E_{ih}} w_{ih} \cdot \log p(p_j|h). \tag{6}$$

Here we omit the derivation of Eq 6 since it has a similar derivation with Eq 3. From the point of view of Eq 6, if two places p_i and p_j have similar temporal distributions, the optimization process will pull their embeddings close to each other.

4 EXPERIMENTS

In this section, we describe our datasets and present the experimental results to demonstrate PPE's performance.

4.1 Dataset description

We have carried out a set of experiments on three large real LBSN datasets, i.e., two original datasets from Instagram [4] and one from Foursquare [9]. The two datasets of Instagram contain the check-in information of New York and London, respectively. The Foursquare dataset records the check-in situation of New York as well. Each place in the three datasets has at most one tag. The statistics of the datasets are shown in Table 1.

Table 1: The statistics of the three datasets.

Properties	New York (Instagram)	London (Instagram)	New York (Foursqaure)
#Check-ins	855,493	2,788,527	227,428
#Users	49,738	39,994	1,083
#Places	27,940	22,817	38,333
#Tags	381	381	400

In the datasets, many places are not "active" enough, in the sense that they only have at most 10 check-in records. Thus, we need to remove all such "inactive" places from each dataset as a preprocessing step. Moreover, for comparison purpose, 1) we group all 381 tags of Instagram datasets into 10 superclasses; 2) we build three new sub-datasets from original Instagram and Foursquare datasets, which only contain places with top-20 tags ranked by the number of check-ins.

4.2 Experimental results and analysis

In this paper, we compare our PPE framework with the SAP [11] and LINE [6] models. Here we adopt LINE model to train a place-user bipartite graph and directly obtain place embeddings. For PPE and LINE, the number of the dimension for user and place vectors is set as 200 and the number of iteration for optimization is 100 million. The prediction is performed with the SVM classifier and based on 10-fold cross-validation. For measurement, the standard micro-averaged and macro-averaged F_1 are used to evaluate all experimental results.

Table 2 shows the prediction results on three sub-datasets of Instagram and Foursquare with 10 superclasses and top-20 classes respectively. From the table, we draw the following observations:

1. Our PPE outperforms the other two models significantly;
2. The LINE model is stably better than the SAP model;
3. The results of our PPE on the Foursquare datasets is worse than its results on the Instagram datasets.

For Observations 1 and 2, our PPE is around 30% better than the LINE model since the label (tag) information is learned into place embeddings by a user-tag bipartite graph. Meanwhile, the LINE model outperforms the SAP model due to the use of deep learning. From Table 1, we can see that, in the Foursquare dataset, 1,083 users can cover 38,333 places, however in the Instagram dataset, 49,738 users only cover 27,940 places. It means that the distribution $p(u_i|\cdot)$ in Foursquare and Instagram datasets are very different. The distribution from the Foursquare dataset should be much more even than the distribution from Instagram datasets. Hence, any two distributions $p(u_i|\cdot)$ and $p(u_j|\cdot)$ in Foursquare are much closer to each other than the same context in the Instagram datasets, This gives rise to the fact that user and place embeddings become less discriminative, explaining Observation 3.

5 CONCLUSION

We have presented a semi-supervised learning framework to generate discriminative low-dimension embeddings for places in LBSNs. For semantic annotation, our framework outperforms the two state-of-the-art representation learning models with SVM. The success of our framework is due not only to the foundation of deep learning and the introduction of label information into place embeddings, but also the intriguing idea of representing places by user behaviors.

Table 2: The prediction results. (The upper part is for the two Instagram sub-datasets with 10 superclasses, the lower is for all sub-datasets with 20 top classes; ut, pt and pu are user-tag, place-temporal, place-user graphs; for PPE(ut+pt), $\lambda = 0.02$.)

Model	New York Instagram		London Instagram		New York Foursqaure	
	Micro	Macro	Micro	Macro	Micro	Macro
PPE(ut)	54.6	39.3	52.3	40.0	N/A	N/A
PPE(ut+pt)	57.0	41.6	53.8	40.5	N/A	N/A
LINE(pu)	37.1	20.7	49.3	33.5	N/A	N/A
SAP	29.2	12.0	29.3	4.5	N/A	N/A
PPE(ut)	48.7	41.0	55.7	38.9	38.3	32.3
PPE(ut+pt)	49.1	38.0	57.6	39.8	28.5	14.9
LINE(pu)	28.2	12.5	35.9	14.4	22.5	13.0
SAP	17.7	6.2	27.0	2.1	15.2	6.2

ACKNOWLEDGMENTS

This work has been partially supported by National Key Research Program of China (2016YFB1001101) and the Joint Funds of the National Nature Science Foundation of China (U1509214).

REFERENCES

[1] John Krumm and Dany Rouhana. 2013. Placer: Semantic place labels from diary data. In *Proc. 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing.* ACM, 163–172.

[2] John Krumm, Dany Rouhana, and Ming-Wei Chang. 2015. Placer++: Semantic place labels beyond the visit. In *Proc. 13th IEEE International Conference on Pervasive Computing and Communications.* IEEE CS, 11–19.

[3] Lin Liao, Dieter Fox, and Henry Kautz. 2007. Extracting places and activities from gps traces using hierarchical conditional random fields. *The International Journal of Robotics Research* 26, 1 (2007), 119–134.

[4] Jun Pang and Yang Zhang. 2017. DeepCity: A feature learning framework for mining location check-ins. In *Proc. 11th International AAAI Conference on Web and Social Media.* AAAI, 652–655.

[5] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online learning of social representations. In *Proc. 20th ACM International Conference on Knowledge Discovery and Data Mining.* ACM, 701–710.

[6] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. LINE: Large-scale information network embedding. In *Proceedings of 24th International World Wide Web Conference.* ACM, 1067–1077.

[7] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. PTE: Predictive text embedding through large-scale heterogeneous text networks. In *Proc. 21st ACM International Conference on Knowledge Discovery and Data Mining.* ACM, 1165–1174.

[8] Min Xie, Hongzhi Yin, Fanjiang Xu, Hao Wang, and Xiaofang Zhou. 2016. Graph-based metric embedding for next POI recommendation. In *Proc. 25th International Conference on Web Information Systems Engineering (LNCS)*, Vol. 10042. Sringer, 207–222.

[9] Dingqi Yang, Daqing Zhang, Vincent W. Zheng, and Zhiyong Yu. 2015. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2015), 129–142.

[10] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2016. Revisiting semi-supervised learning with graph embeddings. In *Proc. 33th International Conference on Machine Learning.* IEEE CS, 40–48.

[11] Mao Ye, Dong Shou, Wang-Chien Lee, Peifeng Yin, and Krzysztof Janowicz. 2011. On the semantic annotation of places in location-based social networks. In *Proc. 17th ACM International Conference on Knowledge Discovery and Data Mining.* ACM, 520–528.

[12] Yin Zhu, Erheng Zhong, Zhongqi Lu, and Qiang Yang. 2012. Feature engineering for place category classification. In *Proceedings of Workshop on the Nokia Mobile Data Challenge.*