CIFEF: Combining Implicit and Explicit Features for Friendship Inference in Location-Based Social Networks

Cheng He[1], Chao Peng[1(✉)], Na Li[2], Xiang Chen[3], Zhengfeng Yang[1], and Zhenhao Hu[1]

[1] Shanghai Key Laboratory of Trustworthy Computing, Software Engineering Institute, East China Normal University, Shanghai, China
chenghe28@qq.com,
{cpeng,zfyang}@sei.ecnu.edu.cn,
hzh@stu.ecnu.edu.cn

[2] School of Data Science and Engineering, East China Normal University, Shanghai, China
nali0606@foxmail.com

[3] School of Information Science and Technology, Nantong University, Nantong, China
xchencs@ntu.edu.cn

Abstract. With the increasing popularity of location-based social networks (LBSNs), users can share their check-in location information more easily. One of the most active problems in LBSNs is friendship inference based on their rich check-in data. Previous studies are mainly based on co-occurrences of two users, however, a large number of user pairs have no co-occurrence, which weakens the performance of previous proposed methods. In this paper, we propose a method CIFEF that ***C***ombines the ***I***mplicit ***F***eatures and a ***E***xplicit ***F***eature for friendship inference. Specifically, based on whether a user has different trajectory patterns on weekdays and weekends, we take the embedding technique to learn implicit weekdays' trajectory features and weekends' trajectory features from their check-in trajectory sequences, respectively, which can work effectively even for user pairs with no co-occurrence. Moreover, we propose a new explicit feature to capture the explicit information of user pairs who have common locations. Extensive experiments on two real-world LBSNs datasets show that our proposed method CIFEF can outperform six state-of-the-art methods.

Keywords: Location-based social networks · Implicit features · Explicit features · Friendship inference

---

This research is supported by the Scientific and Technological Innovation 2030 Major Projects under Grant 2018AAA0100902.

G. Li et al. (Eds.): KSEM 2020, LNAI 12275, pp. 168–180, 2020.
https://doi.org/10.1007/978-3-030-55393-7_16

1 Introduction

In the past decade, with the rising popularity of smart phones, applications on location-based social networks (LBSNs) [5] have attracted tens of millions of users. In LBSNs, users can share their location information (i.e., check-in) when they find a new place or take part in social activities. Huge volume check-in data of users are collected from location-based social networking services, which provides an opportunity for researchers to study various social behaviors.

One interesting question is whether we can infer the social relationship between two users based on their check-in data in LBSNs. According to social homophily principle [12], friends tend to visit a certain number of same locations to participate in some social activities, such as attending the wedding of their common friends or having dinner together at a restaurant. Inferring the relationship between two users have been largely adopted in friend recommendation [7], social influence analysis [18], and targeted marketing [11]. Therefore, given the check-in data of two users, inferring whether they are friends or strangers attracts a lot of researches [4,10,15,16,19–22].

However, previous studies have three major shortcomings. Firstly, they mined some co-occurrence (refer to Definition 2) features of user pairs to address the friendship prediction problem. In this situation, if friend pairs share co-occurrences rarely, it will affect the performance of these methods. In Table 1, we show the ratio of the friend pairs who have at least one co-occurrence on Gowalla dataset and Brightkite dataset [13], respectively. We can observe that even if we set the time threshold to 120 min and distance threshold to 200 m of co-occurrence, only 30.25% and 47.17% of friend pairs have at least one co-occurrence. Therefore, the performance of the friendship inference method based on co-occurrence is not satisfactory.

Secondly, in previous studies, He et al. [10] considered the distances between the two locations where users checked in most frequently on weekdays and weekends. However, as shown in Fig. 1, the users generate significantly different trajectory patterns on weekdays and on weekends. Therefore, modelling the users trajectories on weekday and weekend separately is more reasonable.

Thirdly, previous work used location entropy to measure the popularity of common place between two users. However, the time interval between two users' visit should be considered. If two users visit a same place, but there is a long time interval between their visits, they may not have any relationship.

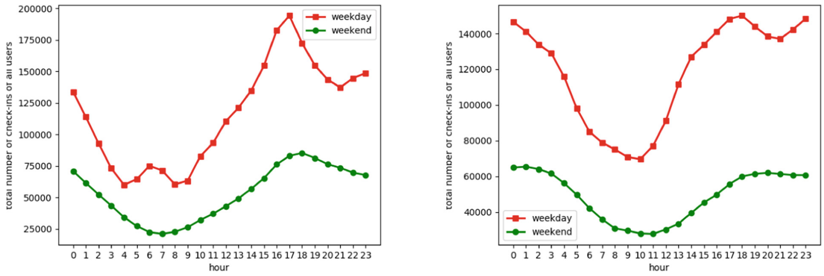
To address the aforementioned issues, we propose a method that combining implicit and explicit features (CIFEF) for inferring friendship in LBSNs. Specifically, we first exploit the embedding learning technique to capture each user's contextual trajectory information of weekdays and weekends, respectively. In this way, we can get the implicit vector representation of each user, which does not rely on the co-occurrences. Besides, we further propose a new explicit feature by introducing check-in time factor into location entropy, which can mine the explicit information of user pairs who have visited a common place. In summary, to our best knowledge, the major contributions of this paper are as follows:

1. We exploit the embedding technique to learn latent vector representation of user's trajectory, which can work effectively even for user pairs with no co-occurrence. Moreover, because a user has different trajectory patterns on weekdays and weekends, we learn implicit representation of users weekday's trajectory and weekend's trajectory, respectively.
2. We further propose a new feature named *twcle* to measure the importance of each common place of user pairs by introducing the time interval of check-ins into location entropy.
3. We propose an effective method CIFEF to infer friendship in LBSNs, which combines implicit features and explicit feature for inferring friendship.
4. We conduct extensive experiments on two real-world datasets to evaluate the performance of our proposed method and the experiment results show that our method is superior to six state-of-art methods.

Table 1. The ratio of friend pairs who have at least one co-occurrence under different time threshold and distance threshold

(Time threshold, Distance threshold)	Gowalla	Brightkite
(10 min, 0 m)	14.21%	19.94%
(10 min, 100 m)	16.21%	22.62%
(10 min, 200 m)	17.60%	24.11%
(30 min, 0 m)	19.04%	27.72%
(30 min, 100 m)	21.35%	31.44%
(30 min, 200 m)	23.03%	33.60%
(60 min, 0 m)	22.57%	33.13%
(60 min, 100 m)	24.92%	37.52%
(60 min, 200 m)	26.85%	40.20%
(120 min, 0 m)	25.73%	38.88%
(120 min, 100 m)	28.13%	44.05%
(120 min, 200 m)	30.25%	47.17%

The rest of the paper is organized as follows. In Sect. 2, we give a brief review on related work. In Sect. 3, we introduce the preliminaries and the detail of our method. In Sect. 4, we report our experimental study. In Sect. 5, we discuss the contribution of the implicit features and explicit feature to our method. We will also analysis the sensitivity of our method to the parameter *embedding size*. In Sect. 6, we conclude our work and discuss the future work.



(a) Gowalla (b) Brightkite

Fig. 1. Trajectory patterns at different hours on weekday and weekend

2 Related Work

Inferring social relationships from location-based check-in data has been a hot research topic in the past few years. The existing studies can be roughly classified into two categories based on the features they consider: co-occurrence based approaches and others. The co-occurrence based methods rely on the explicit co-occurrence [4,10,16,19,21], these methods mainly mine some co-occurrence based features of user pairs. However, these approaches need to set time thresholds and distance thresholds for co-occurrence. What's worse, a large number of user pairs do not have co-occurrences in real life as shown in Table 1. Some another approaches measure the similarity of user pairs by well-designed location features, such as the distance of home[20], the Jaccard similarity of check-in sequences and the number of common locations, etc. Pham et al. [19] proposed an entropy-based model, which designed two features: *location diversity* and *weight frequency*. The *location diversity* measures how diverse the co-occurrences of a user pair are. The *weight frequency* uses location entropy to measure how important the user pair's co-occurrences. Wang et al. [21] argued that not all co-occurrences are equally important in predicting user pair's friendship. They considered three features, including *personal factor*, *global factor* and *temporal factor*. The *personal factor* aims to determine the significance of a co-occurrence between two users, they think that if two users meet at a place where they frequently visit, then this meeting event is likely to happen by chance. The *global factor* is to reward the co-occurrence at private locations and penalize those at public locations. The *temporal factor* aims to penalize a co-occurrence if it is temporally close to other events. Njoo et al. [16] proposed two features *stability* and *duration* in the temporal domain of the co-occurrences, which can reflect the consistency and the total duration of the co-occurrences between two users. Cheng et al. [4] proposed a feature called *weighted number of co-occurrences*, which were aimed to strengthen the co-occurrence happened at private locations and while weaken the co-occurrence happened at popular locations. He et al. [10] designed 12 spatiotemporal features from four aspects to infer friendship.

In the above mentioned works, the co-occurrence based methods cannot deal with the situation that the user pairs have no co-occurrence. Besides, They all ignored the time interval between two users who have visited a same place. Therefore, we propose a method that combines implicit embedding features and one explicit feature to address the above issues.

3 The Proposed Method

In this section, we introduce the preliminaries and the details of our method.

3.1 Preliminaries

Definition 1 (Check-in Triplet). *When user u checks in location l at time t, the information can be called a check-in triplet $c_u =< u, l, t >$. Given user u, all his check-ins form a trajectory sequence $S_u = \{< u, l_1, t_1 >, \cdots, < u, l_n, t_n >\}$.*

Definition 2 (Co-occurrence). *A user u_1 and a user u_2 have a co-occurrence if their check-in distance is less than a distance threshold and the time interval is less than a time threshold.*

3.2 Implicit Features

As shown in Table 1, most friend pairs have no co-occurrences. Besides, hand-designed features cannot capture some implicit information of user's check-in. Therefore, it prompts us to learn the latent information from user's check-in data. *Word2vec* [14] is a very effective method to learn embedding representations in word sequences, which achieves a great success in recommendation systems [2]. Therefore, we also adopt *word2vec* to learn the trajectory embedding of each user. Concretely, we view each check-in location as a "word" and each "sentence" represents a user trajectory, then using skip-gram [14] to learn a location latent vector. In skip-gram model, given a user's check-in trajectory sequence S_u and the window size k, we need to maximize the following objective function:

$$J(S_u) = \frac{1}{|S_u|} \sum_{l_i \in S_u} \sum_{-k \leq j \leq k} (\log(P(l_{i+j}|l_i)) \tag{1}$$

where l_i represents the target location and l_{i+j} denotes the context location. Then we adopt a softmax function to formulate the probability $P(l_{i+j}|l_i)$:

$$P(l_{i+j}|l_i) = \frac{exp(I'_{i+j} \cdot I_i)}{\sum_{l_i \in L} exp(I'_{l_i} \cdot I_i)} \tag{2}$$

where I' and I denote output and input vector of location, respectively. $|L|$ is the number of all locations in the dataset. From Eq. 2, we can know its time complexity is very high, we apply the negative sampling method [14] to speed up learning process and the objective function can be redefined as follows:

$$J(S_u) = \frac{1}{|S_u|} \sum_{l_i \in S_u} \sum_{-k \leq j \leq k} (\log(\sigma(I'_{i+j} \cdot I_i)) + \sum_{h=1}^{H} \log(\sigma(-I'_h \cdot I_i))) \tag{3}$$

where $\{I'_h|h = 1...H\}$ are sampled from L, and $\sigma(\cdot)$ is the sigmoid function.

We train the above model using stochastic gradient descent and finally obtain an M-dimensional embedding vector for each location. Therefore, for the trajectory sequence S_u, we can get a $N \times M$ matrix $W_u = (w_{l_1}, w_{l_2}, \cdots, w_{l_N})^T$ and take the maximum value for each column of the W_u. Finally, we can get the vector representation of user u:

$$V_u = [v_u^1, v_u^2, \cdots, v_u^n]^T \tag{4}$$

3.3 Explicit Feature

Location entropy [6] was used to measure the popularity of a location, which can be defined as follows:

$$loc\_entropy(l) = -\sum_{u \in \Phi_l} \frac{C_u^l}{C^l} log \frac{C_u^l}{C^l} \tag{5}$$

where Φ_l represents all users who have checked in location l, C_u^l represents the check-in frequency of the user u at l and C^l is the total number of check-ins that all users have at place l. If two users meet at a popular location, they may be strangers because their meeting is more likely to happen by chance.

However, the location entropy ignored the check-in time interval of two users. For example, there are three users $< u_1, u_2, u_3 >$, u_1 checked in at location l_1 at 3 p.m. on December 5, 2019, u_2 checked in at location l_1 at 10 a.m. on December 1, 2018 and u_3 checked in at location l_1 at 3:30 p.m. on December 5, 2019. Although $< u_1, u_2 >$, $< u_1, u_3 >$ both have one common place, respectively, it is intuitive that $< u_1, u_3 >$ is more likely to be friends than the $< u_1, u_2 >$ because the check in time of $< u_1, u_3 >$ is closer.

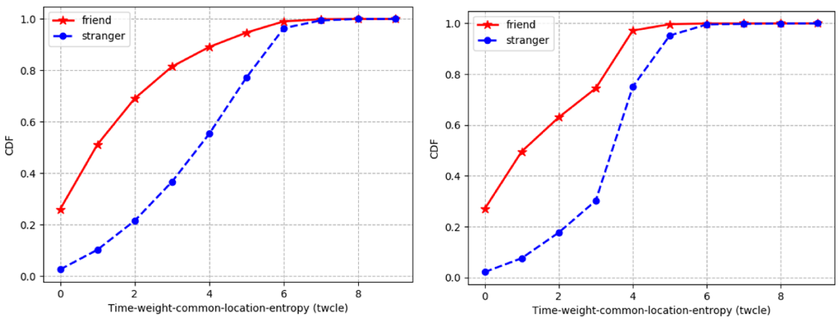
Based on the above consideration, by introducing the time interval of check-ins into location entropy, we propose a new feature named *Time-weight-common-location-entropy (twcle)*. it can be defined as follows:

$$twcle(u, v, l) = loc\_entropy(l) \times e^{\rho \times |t_u^l - t_v^l|} \tag{6}$$

where t_u^l is the check-in time of the user u at location l. The parameter ρ is a constant coefficient to prevent the result of exponential function from overflowing. In our experiment, we set it to 0.001. Note that the time unit is a day. From the above formula, we can know that if two users visit a same private location at a closer time, their *twcle* value is lower. For two users u and v, the set of their common places is $L_{u,v}$, their *twcle* value can be computed as follows:

$$twcle(u, v, L_{u,v}) = \min_{l \in L_{u,v}} (loc\_entropy(l) \times e^{\rho \times |t_u^l - t_v^l|}) \tag{7}$$

To verify the effectiveness of *twcle* in differentiate the friend pairs and stranger pairs, we plot the cumulative distribution function (CDF) of *twcle* between friends and strangers. As shown in Fig. 2, the friend pairs and stranger pairs are easily separable on two dataset, respectively. Specifically, on Gowalla dataset, more than 80% of friend pairs' *twcle* values are less than 3, while only less than 40% of stranger pairs' *twcle* values are below 3. On Brightkite dataset, the maximum gap is 45%. These indicate the *twcle* feature is an effective measure to differentiate the friend pairs and stranger pairs.



(a) Gowalla (b) Brightkite

Fig. 2. The CDF of *twcle* between friend pairs and stranger pairs

3.4 The Detail of Our Method CIFEF

In this subsection, we introduce the details of our friendship inference method CIFEF. Figure 3 shows the overview of CIFEF. Firstly, we divide each user's check-in trajectory into weekday trajectory and weekend trajectory, and utilize *word2vec* to learn the embedding vector, respectively. Then, we apply element-wise max operations to the list of weekday location embedding vectors to get weekday trajectory embedding vector, weekend location does the same operation. Thirdly, for each user pair, their weekday vector and weekend vector are sent to the interaction layer for interacting (i.e., element-wise multiplication and element-wise subtraction). Next, each user pair gets four vectors: *weekday_hm*, *weekday_sub*, *weekend_hm* and *weekend_sub*. Together with their weekday vector and weekend vector, we finally get eight feature vectors. Moreover, we extract *twcle* feature according to Eq. 7. Finally, based on the above eight feature vectors and the *twcle* feature, we train the friendship inference model.

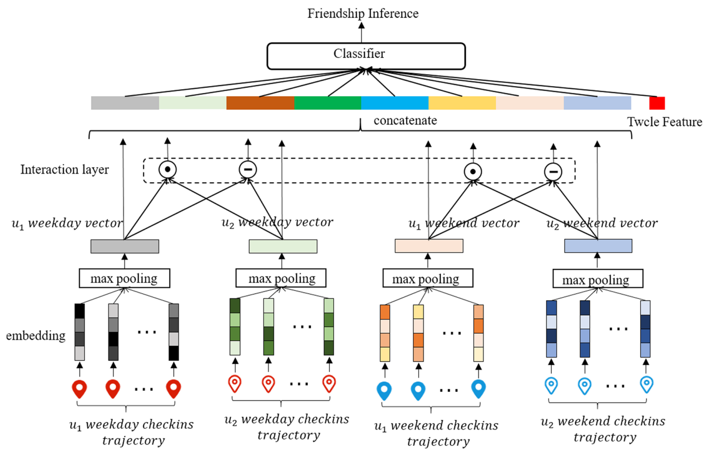


Fig. 3. The overview of our proposed method CIFEF

4 Experiment Study

In this section, we show experimental study in detail, including datasets, evaluation strategy and metric, baseline methods and experiment result.

4.1 Dataset

We conduct our experiment on two public real-world datasets: Gowalla dataset and Brightkite dataset [13], which have been widely used in previous studies. Table 2 shows the detailed statistics of the two datasets. We first select the active users who have more than 100 check-ins from two original dataset, getting 54,713 friend pairs and 54,221 friend pairs from two datasets, respectively. Then we double the dataset size by randomly sample the same number of stranger pairs in the two processed dataset, respectively. Finally, we conduct our experiment on processed Gowalla dataset with 109,426 samples and processed Brightkite dataset with 108,442 samples.

Table 2. Statistics of Datasets

Dataset	#Users	#Check-ins	#Friend pairs
Gowalla	107,092	6,442,890	950,327
Brightkite	58,228	4,491,143	214,078

4.2 Evaluation Strategy and Metric

To verify the effectiveness of CIFEF, we use AUC score as evaluation metric in our experiments. Moreover, to confidently evaluate the performance of CIFEF, we consider four classical machine learning algorithms: Logistic Regression (LR), K-Nearest Neighbor (KNN), Support Vector Machine (SVM) and Random Forest (RF) in our experiment. These algorithms are implemented by using Scikit-learn version 0.22.0 [17], with default value for hyper parameters. Lastly, all the experiments are done by using 5-fold cross validation.

4.3 Baseline Methods

To show the effectiveness of CIFEF, we consider the following baseline methods.

1. **STIF:** As introduced in Sect. 2. STIF [10] designed 12 features from four aspects for inferring friendship. This method include some co-occurrence based features and some spatiotemporal features.
2. **PGT:** PGT [21] used 6 features by considering the combinations of three features described in Sect. 2. It relies on the co-occurrences of user pairs.
3. **SCI:** SCI [16] extracted three features based on the co-occurrences as introduced in Sect. 2. it also relies on the co-occurrences of user pairs.
4. **SCI+:** SCI+ [15] is an improved version of SCI, which generalizes the temporal model by accommodating all meeting events between users instead of just considering the last meeting event. Besides, they also use two previous features: co-occurrences frequency and location popularity.
5. **OSFLC:** Bayrak et al. [1] aimed to reduce the time cost of friendship inference by feature selection. The 19 location-based features were collected from previous friendship inference papers in LBSNs. We use 15 features of them (there are 4 features we cant use because they rely on location category, which is not available in the dataset we use.) as a baseline. This method does not rely the co-occurrences of user pairs.
6. **CIFEF_T:** This is a baseline we designed, which uses a user's complete trajectory sequence to train the embedding vector instead of dividing the trajectory into workdays trajectory and weekends trajectory. The other parts are consistent with CIFEF method.

4.4 Comparison with Baseline Methods

In this Subsection, we report the experiment results of baseline methods and our method CIEFE. Table 3 shows the AUC scores of CIEFE and the above baseline methods. We firstly analyze the performance of baseline methods that based on co-occurrence, i.e., SCI, SCI+ and PGT. From Table 3, we can see that these three methods have a relatively poor performance, mainly because there are very few user pairs who have co-occurrences. Therefore, features extracted by these methods contain lots of missed value, which weakens their performance.

Table 3. AUC for different supervised classifiers on the two datasets

	Gowalla					Brightkite				
	LR	KNN	SVM	RF	Average	LR	KNN	SVM	RF	Average
CIFEF	**0.861**	**0.889**	**0.859**	**0.939**	**0.887**	**0.834**	**0.871**	**0.832**	0.903	**0.860**
CIFEF_T	0.847	0.887	0.849	0.911	0.874	0.823	0.856	0.820	0.876	0.844
OSFLC	0.804	0.865	0.773	0.914	0.839	0.752	0.863	0.746	**0.911**	0.818
STIF	0.733	0.826	0.771	0.873	0.801	0.715	0.799	0.740	0.876	0.783
PGT	0.599	0.599	0.600	0.599	0.599	0.564	0.581	0.557	0.581	0.571
SCI	0.551	0.551	0.551	0.551	0.551	0.556	0.537	0.541	0.556	0.548
SCI+	0.597	0.596	0.597	0.596	0.597	0.576	0.575	0.576	0.575	0.576

Secondly, we discuss the baseline STIF. Table 3 shows that on Gowalla dataset, the average AUC value of STIF has a 33.7%, 45.4%, and 34.2% performance improvement over those of PGT, SCI and SCI+, respectively. On Brightkite dataset, compared to PGT, SCI and SCI+, STIF improves AUC by 37.1%, 42.9% and 35.9%, respectively. The reason why the performance of STIF can be so significantly improved is that it not only contains some co-occurrence based features, but also some other spatiotemporal features, which makes it possible to work well even without co-occurrence.

Thirdly, we discuss the baseline OSFLC, which does not depend on co-occurrence. As shown in Table 3, although the performance of OSFLC is worse than our method CIFEF, it achieves much better performance than the above four baseline methods. It is worth mentioning that on the Brightkite dataset, for the random forest classifier, OSFLC method has achieved better performance than all other methods. However, OSFLC method needs a large number of well-designed features, which requires strong expertise knowledge.

Nextly, from Table 3, we can observe that CIFEF_T achieves better performance compared to SCI, SCI+, PGT, STIF and OSFLC. However, it performs worse than CIFEF by 1.5% and 1.9% on Gowalla dataset and Brightkite dataset, respectively. This shows that dividing user's trajectory into weekdays' trajectory and weekends' trajectory is good for friendship inference in LBSNs.

Finally, except the case of RF classifier on Brightkite, our method CIFEF outperforms all baseline methods in all cases. Compared to OSFLC method, the average AUC score of CIFEF has a 5.7% and 5.1% improvement on Gowalla dataset and Brightkite dataset, respectively. The major reason is that our method is not only able to capture the explicit information of user pairs who share common place, but also learns latent multi-grain trajectory information, which don't rely on user pair's co-occurrence. However, on RF classifier, CIFEF performs slightly worse than OSFLC method by 0.8%. We observe that compared with Gowalla dataset, the performance of all methods on the Brightkite dataset decreased at different degrees. According to our statistics, on Gowalla dataset, 12% of users check in more than 50 times with the interval less than

5 min, while the number is 25% on Brightkite dataset. Therefore, it means that there is more noise on Brightkite dataset.

5 Discussion

In this section, we first discuss the contributions of the implicit features and explicit feature to CIFEF. Then, we analyze the sensitivity of our method to the parameter *embedding size*.

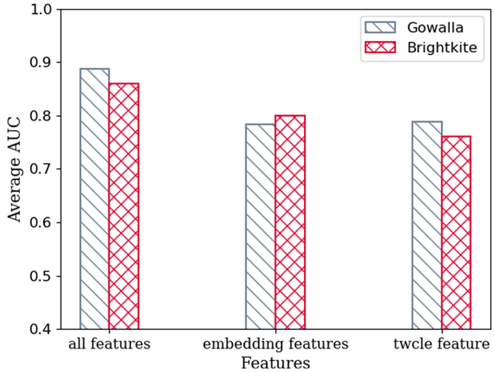


Fig. 4. The contributes of implicit features and explicit feature to CIFEF

5.1 The Contributions of Implicit Features and Explicit Feature

In this Subsection, we study the contributions of implicit features and explicit feature (i.e., the embedding features and the *twcle* feature) to CIFEF. Figure 4 shows the average AUC score of the aforementioned four classifiers. We can see that both features are useful to improve the performance of CIFEF. Concretely, on Gowalla dataset, the contribution of *twcle* feature is slightly greater than the embedding features, while the embedding features' contribution is greater than *twcle* feature on Brightkite dataset. Moreover, the average AUC score of embedding features on two datasets are 0.784 and 0.799, which are all worse than the average AUC score of OSFLC. However, the AUC score is greatly improved by adding the *twcle* feature, which is significantly better than OSFLC.

5.2 Paramter Sensitivity Analysis

We try several values of *embedding size*: {4,8,10,12,14,16,32,64,128}. From the Fig. 5, we can observe that our method achieves the best performance when embedding size is set to 8 on Gowalla dataset and 14 on Brightkite dataset. It seems to be a good balance for two datasets when setting the embedding size to 32.

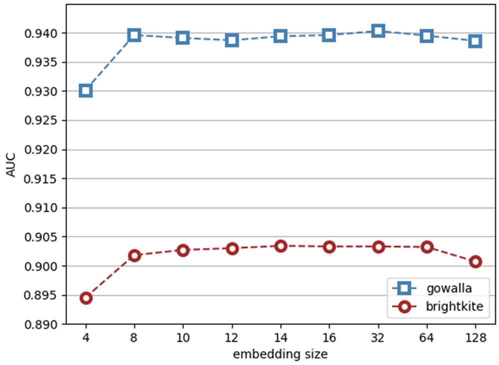


Fig. 5. The AUC of different embedding sizes on RF classifier

6 Conclusion and Future Work

In this paper, we study the problem of friendship inference based on users' check-in data in LBSNs. We adopt the embedding method to learn implicit features of user's weekday trajectory and weekend trajectory, respectively, which works effectively even if user pairs have no co-occurrence. Meanwhile, we propose a new feature *twcle*, which measures the importance of user pair's common place based on the time interval of check-in and location entropy. We have conducted extensive experiments on two public real-world datasets, the experiment results demonstrate the superiority of our method over state-of-the-art baseline methods. For the future work, we plan to design a more effective end-to-end deep learning model[3,8,9] for friendship inference in LBSNs.

References

1. Bayrak, A.E., Polat, F.: Mining individual features to enhance link prediction efficiency in location based social networks. In: ASONAM, pp. 920–925 (2018)
2. Caselles-Dupré, H., Lesaint, F., Royo-Letelier, J.: Word2vec applied to recommendation: hyperparameters matter. In: RecSys, pp. 352–356. ACM (2018)
3. Chen, S., Peng, C., Cai, L., Guo, L.: A deep neural network model for target-based sentiment analysis. In: IJCNN2018, pp. 1–7. IEEE (2018)
4. Cheng, R., Pang, J., Zhang, Y.: Inferring friendship from check-in data of location-based social networks. In: ASONAM, pp. 1284–1291. ACM (2015)
5. Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: SIGKDD, pp. 1082–1090. ACM (2011)
6. Cranshaw, J., Toch, E., Hong, J., Kittur, A., Sadeh, N.: Bridging the gap between physical location and online social networks. In: UbiComp2010, pp. 119–128 (2010)
7. Ding, D., Zhang, M., Li, S.Y., Tang, J., Chen, X., Zhou, Z.H.: Baydnn: friend recommendation with bayesian personalized ranking deep neural network. In: CIKM, pp. 1479–1488. ACM (2017)
8. Ding, J., Zhang, X., Chen, M., Xue, K., Zhang, C., Pan, M.: Differentially private robust ADMM for distributed machine learning. In: BigData, pp. 1302–1311 (2019)

9. Gu, H., et al.: A collaborative and sustainable edge-cloud architecture for object tracking with convolutional siamese networks. IEEE Trans. Sustain. Comput. (2019)
10. He, C., Peng, C., Li, N., Chen, X., Guo, L.: Exploiting spatiotemporal features to infer friendship in location-based social networks. In: Geng, X., Kang, B.-H. (eds.) PRICAI 2018. LNCS (LNAI), vol. 11013, pp. 395–403. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-97310-4_45
11. Hu, Z., Peng, C., He, C.: Io-aware factorization machine for user response prediction. In: IJCNN2020. IEEE (2020)
12. Kossinets, G., Watts, D.J.: Origins of homophily in an evolving social network. Am. J. Sociol. **115**(2), 405–450 (2009)
13. Leskovec, J., Krevl, A.: SNAP Datasets: Stanford large network dataset collection, June 2014. http://snap.stanford.edu/data
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NeurIPS, pp. 3111–3119 (2013)
15. Njoo, G.S., Hsu, K.W., Peng, W.C.: Distinguishing friends from strangers in location-based social networks using co-location. PMC **50**, 114–123 (2018)
16. Njoo, G.S., Kao, M.-C., Hsu, K.-W., Peng, W.-C.: Exploring check-in data to infer social ties in location based social networks. In: Kim, J., Shim, K., Cao, L., Lee, J.-G., Lin, X., Moon, Y.-S. (eds.) PAKDD 2017. LNCS (LNAI), vol. 10234, pp. 460–471. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57454-7_36
17. Pedregosa, F., et al.: Scikit-learn: machine learning in python. JMLR **12**, 2825–2830 (2011)
18. Peng, S., Yang, A., Cao, L., Yu, S.: Social influence modeling using information theory in mobile social networks. Inf. Sci. **379**, 146–159 (2017)
19. Pham, H., Shahabi, C., Liu, Y.: EBM: an entropy-based model to infer social strength from spatiotemporal data. In: SIGMOD, pp. 265–276. ACM (2013)
20. Scellato, S., Noulas, A., Mascolo, C.: Exploiting place features in link prediction on location-based social networks. In: SIGKDD, pp. 1046–1054. ACM (2011)
21. Wang, H., Li, Z., Lee, W.C.: PGT: Measuring mobility relationship using personal, global and temporal factors. In: ICDM, pp. 570–579. IEEE (2014)
22. Zhang, Y., Pang, J.: Distance and friendship: a distance-based model for link prediction in social networks. In: Cheng, R., Cui, B., Zhang, Z., Cai, R., Xu, J. (eds.) APWeb 2015. LNCS, vol. 9313, pp. 55–66. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25255-1_5