DeepCity: A Feature Learning Framework for Mining Location Check-Ins

Jun Pang
SnT& FSTC
University of Luxembourg

Yang Zhang
CISPA, Saarland University
Saarland Informatics Campus

Abstract

Online social networks being extended to geographical space has resulted in large amount of user check-in data. Understanding check-ins can help to build appealing applications, such as location recommendation. In this paper, we propose DeepCity, a feature learning framework based on deep learning, to profile users and locations, with respect to user demographics and location category prediction. Both of the predictions are essential for social network companies to increase user engagement. The key contribution of DeepCity is the proposal of task-specific random walk which uses the location and user properties to guide the feature learning to be specific to each prediction task. Experiments conducted on 42M check-ins in three cities collected from Instagram have shown that DeepCity achieves a superior performance and outperforms state-of-the-art models significantly.

Introduction

The advancement of positioning technologies has extended online social networks (OSNs) to geographical space. Nowadays, OSN users frequently share their photos or statues together with geographical locations, namely *check-ins*. The large quantity of user check-in data has provided us with an unprecedented chance to study users' mobility behaviors. One important aim of understanding user check-ins is to profile users with the assumption that whereabouts of a user reflects who he is. Profiling users is essential for OSNs since it can help to increase user engagement. On the other side of the coin, check-in data can also help us to gain more understandings about locations, i.e., location profiling. One important problem in this direction is using users' check-in behavior at a certain location to infer the location's category.

Mining user check-ins has attracted academia a considerable amount of attention. Researchers have tackled various problems with the help of machine learning techniques. However, these tasks, most of which involve prediction, require hand-engineering features for learning algorithms. Except for the tedious efforts spent on feature engineering, these features, in many cases, are not complete.

The recent development of deep learning (Mikolov et al. 2013a; 2013b) has provided an alternative way to learn features for prediction tasks based on network structures (Perozzi, Al-Rfou, and Skiena 2014). In this setting, features are learned by optimizing a general objective function, and thus can be applied in any prediction tasks. These methods are also referred as network embedding, and the learned features of each node is named as the node's embedded vector. However, these network embedding approaches often neglect the specific knowledge w.r.t. different prediction tasks, which eventually affects the prediction performance.

In the current paper, we propose DeepCity, a general feature learning framework for mining user check-ins shared in OSNs. We concentrate on two data mining tasks: user profiling, represented by demographic inference, and location profiling, represented by location category inference. DeepCity adopts the state-of-the-art network embedding method, namely Skip-gram, to learn features for machine learning algorithms. DeepCity's key contribution is the proposal of task-specific random walk which establishes the connection between feature learning and prediction tasks in order to achieve a strong prediction performance. Concretely, for each prediction, task-specific random walk utilizes location or user properties to guide the algorithm to define each user's (location's) neighbors to be more specific to the prediction. In this way, the learned features capture the useful information for the prediction. Our experiments are conducted on a large-scale dataset collected from Instagram containing more than 42M check-ins in New York, Los Angeles and London. Extensive experiments have shown that DeepCity outperforms state-of-the-art models significantly.

DeepCity Framework

Notations. We denote each user by u and each location by ℓ, two sets $\mathcal{U}$ and $\mathcal{L}$ contain all the users and locations, respectively. For demographic prediction, instead of only considering a user's spatial information, we take one step further to consider both his spatial and temporal information, thus we introduce another notation $q \in \mathcal{Q}$ to denote a temporal-location, it is used to describe a user visits *a certain location at certain time*. In the current paper, the temporal information is considered in a 24-hour scale, thus $\mathcal{Q} \subseteq \mathcal{L} \times \{1, 2, \ldots, 24\}$. Similarly, for location category prediction, we define a notion namely temporal-user, denoted by y, which is used to describe a location is visited by *a certain user at certain time*, $\mathcal{Y} \subseteq \mathcal{U} \times \{1, 2, \ldots, 24\}$ contains all the temporal-users. We denote the bipartite graph between $\mathcal{U}$ and $\mathcal{Q}$ by $\mathcal{G}_{\mathcal{U},\mathcal{Q}} = (\mathcal{U}, \mathcal{Q}, \mathcal{W}_{\mathcal{U},\mathcal{Q}})$ where $\mathcal{W}_{\mathcal{U},\mathcal{Q}}$ represents the weighted edges between users and temporal-locations. Correspondingly, $\mathcal{G}_{\mathcal{L},\mathcal{Y}} = (\mathcal{L}, \mathcal{Y}, \mathcal{W}_{\mathcal{L},\mathcal{Y}})$ represents the location temporal-user graph. We define the edge weight from u to q, and from q to u in $\mathcal{G}_{\mathcal{U},\mathcal{Q}}$ as $w_{u,q} = \frac{\tau_{u,q}}{\tau_u}$ and $w_{q,u} = \frac{\tau_{u,q}}{\tau_q}$, respectively. The weights reflect how frequent u visits q and q is visited by u. Similarly, the edge weight from ℓ to y, and from y to ℓ in $\mathcal{G}_{\mathcal{L},\mathcal{Y}}$ are defined as $w_{\ell,y} = \frac{\tau_{\ell,y}}{\tau_\ell}$ and $w_{y,\ell} = \frac{\tau_{\ell,y}}{\tau_y}$. Here, $\tau_{u,q}$ ($\tau_{\ell,y}$) is u's (ℓ's) number of check-ins at q (by y), while τ_u and τ_q (τ_ℓ and τ_y) represent u's and q's (ℓ's and y's) total number of check-ins.

Skip-gram. Skip-gram proposed in (Mikolov et al. 2013a; 2013b) is first designed to embed words in documents into a continuous vector space (word2vec), with the object that a word's embedded vector can predict its nearby words. Perrozi et al. (Perozzi, Al-Rfou, and Skiena 2014) take the Skip-gram model into the network mining field, namely DeepWalk, by establishing an analogy that a node in a network is a word and the network itself is a document. DeepWalk creates "sentences" by simulating random walk traces in the network and feed them into the Skip-gram model to obtain the vector representation of each node.

We introduce the Skip-gram model into user check-in mining. For presentation purposes, we only describe the model for $\mathcal{G}_{\mathcal{U},\mathcal{Q}}$. Skip-gram is essentially a maximal likelihood optimization problem, represented as $\underset{\theta}{\operatorname{argmax}} \prod_{v \in \mathcal{U} \cup \mathcal{Q}} \prod_{n \in \mathcal{N}(v)} p(n|v;\theta)$. Here, $\mathcal{N}(v)$ represents the neighbor nodes of v in random walk traces and θ represents the parameters of the model. To model the conditional probability $p(n|v;\theta)$, a softmax function is adopted: $p(n|v;\theta) = \frac{e^{\varphi_n \cdot \varphi_v}}{\sum\limits_{m \in \mathcal{U} \cup \mathcal{Q}} e^{\varphi_m \cdot \varphi_v}}$ where φ_v and φ_n represent the vectors for node v and its neighbor n, respectively, and $\varphi_n \cdot \varphi_v$ is the two vectors' dot product. Both φ_v and φ_n belong to the parameters θ, and we set the vector length to d for all vectors. By adding every piece together and switching the multiplication to summation through log-likelihood, we obtain the following optimization object:

$$\underset{\theta}{\operatorname{argmax}} \sum_{v \in \mathcal{U} \cup \mathcal{Q}} \sum_{n \in \mathcal{N}(v)} (\varphi_n \cdot \varphi_v - \log(\sum_{m \in \mathcal{U} \cup \mathcal{Q}} e^{\varphi_m \cdot \varphi_v})). \quad (1)$$

The term $\sum_{m \in \mathcal{U} \cup \mathcal{Q}} e^{\varphi_m \cdot \varphi_v}$ is expensive to compute since it requires the summation over all nodes, thus negative sampling is proposed (Mikolov et al. 2013b) which we adopt in the current paper as well. In the end, we utilize stochastic gradient decent (SGD) to optimize the object function to obtain the vector of each node v, i.e., φ_v.

Task-specific random walk. Skip-gram aims to preserve the neighbors of each node. To define a node's neighbors, DeepWalk adopts the random walk approach, i.e., a node's neighbors are those that are close to it in random walk traces. The learned vectors under these models are not specific to any prediction problems, which makes them applicable to any tasks. However, for each individual prediction task, random walk approach does not fully take into account useful information w.r.t. the prediction. For instance, if several users frequently visit the same metro station at the same hour, then they should be close to each other in the learned vector spaces since they have similar neighbors in random walk traces, which will result in them being partitioned into the same class by classifiers. However, the neighbors introduced by the metro station do not provide much information about a user's demographics, i.e., people use metro to commute. On the other hand, if we know that these users often go to locations that are "biased" towards people of certain demographics, then we are more confident that they belong to the same demographic group. Here, "bias" means that a temporal-location's (temporal-user's) check-in distribution over a certain demographic (location category) is quite different from the general check-in distribution over the demographic (location category).

To take into account the bias for specific prediction tasks, we propose task-specific random walk. To proceed, we first propose a measurement to quantify to which extent a temporal-location (temporal-user) is biased towards a certain demographic group (location category). As discussed above, the bias refers to the distribution difference, to capture it, we resort to KL divergence. Taking gender as an example, we denote the number of all check-ins made by female users as τ_{female} and use $p(female) = \frac{\tau_{female}}{\tau_{female} + \tau_{male}}$ to represent the check-in proportion of female users. For a temporal-location q, we use $\tau_{q,female}$ to denote the number of check-ins made by female users at q, and define the proportion $p(q, female) = \frac{\tau_{q,female}}{\tau_{q,female} + \tau_{q,male}}$ Then, the gender biased value of q is defined as $\kappa_{q,gender} = \sum\limits_{i \in \{female, male\}} p(q,i) \log \frac{p(q,i)}{p(i)}$. Our gender biased value follows the original definition of KL divergence, higher value indicates larger difference between two distributions. The biased values of the other two demographics studied in the current paper, i.e., race and age, is defined accordingly. We also utilize KL divergence to quantify the location category biased value of each temporal-user. After obtaining the biased values of each temporal-location w.r.t. all demographics, we extend $\mathcal{G}_{\mathcal{U},\mathcal{Q}}$ into $\mathcal{G}^{\rho}_{\mathcal{U},\mathcal{Q}} = (\mathcal{U}, \mathcal{Q}, \mathcal{W}^{\rho}_{\mathcal{U},\mathcal{Q}})$ where $\rho \in \{gender, race, age\}$. The weight between u and q in $\mathcal{G}^{\rho}_{\mathcal{U},\mathcal{Q}}$ is defined as $w^{\rho}_{u,q} = \frac{\tau_{u,q}}{\tau_u} \kappa_{q,\rho}$ where $\kappa_{q,\rho}$ represents the ρ biased value of the temporal-location q. On the other hand, the edge from q to u in $\mathcal{G}_{\mathcal{U},\mathcal{Q}}$ stays unchanged, i.e., $w^{\rho}_{q,u} = w_{q,u}$. This way of edge weight definition drives our random walks to be biased towards ρ biased temporal-locations, and users frequently visiting these ρ biased locations, which in the end results in each user's neighbors being able to reflect his demographic information. Meanwhile, $\mathcal{G}_{\mathcal{L},\mathcal{Y}}$ is extended into $\mathcal{G}^{\varrho}_{\mathcal{L},\mathcal{Y}} = (\mathcal{L}, \mathcal{Y}, \mathcal{W}^{\varrho}_{\mathcal{L},\mathcal{Y}})$ based on the location category biased value of each temporal-user, and edge weights in $\mathcal{G}^{\varrho}_{\mathcal{L},\mathcal{Y}}$ is modified accordingly.

The task-specific random walk is the normal random walk (on weighted graphs) executed on the above introduced extended bipartite graphs. From each user, we simulate r number of random walk traces and each trace is s step long. The walk traces obtained are fed into Skip-gram for embedding. The learned vectors are treated as features of each node, and are directly fed into machine learning classifiers, in order to perform prediction.

Experiments

Datasets. We utilize Instagram's API to collect check-in data in New York, Los Angeles and London. The reason we choose Instagram is two-fold. First, Instagram users share many more check-ins than other OSNs such as Twitter (Manikonda, Hu, and Kambhampati 2014). Second, Instagram's API is linked with Foursquare's API, which provides us with each check-in location's category information. This enables us to perform location category prediction. Note that Foursqaure organizes location categories into a tree structure, we adopt the first level categories in the tree, as labels for location category prediction.

To obtain each users' demographic information, we resort to Face++, to analyze users' profile photos. Face++ takes a photo as input and returns gender, race (Asian, White, African) and age of the user (or users) in the photo. It is worth noticing that Face++ has been used in many works to obtain users' demographics for analysis, e.g., (Souza et al. 2015). We model age prediction as a classification problem, and adopt the methods in (Felbo et al. 2015) to discretize age into three equal groups: 15-20, 21-25 and 26-36.

In the end, we collect more than 19.6M check-ins in New York, 14.9M in Los Angeles and 8.4M in London. To resolve the data sparseness issue, we concentrate on users with at least 20 check-ins, whom we term as active users.

Experiment setup. For evaluating demographic prediction, we adopt three baseline models including STL (Zhong et al. 2015), MF (Kosinski, Stillwell, and Graepel 2013) and DeepWalk (Perozzi, Al-Rfou, and Skiena 2014). For location category prediction, both MF and DeepWalk can be directly applied. Besides, we also adopt SAP (Ye et al. 2011) as one baseline. We utilize logistic regression as the learning algorithm: user gender is predicted through a binary classifier while the other three tasks are predicted through one-vs-rest logistic regression. We randomly split the learned features with 70% for training classifiers while the left 30% for testing. The random split is repeated for 10 times and we report the average results. We adopt AUC as the metric for evaluating gender prediction and Macro-F1 for age, race and location category prediction.

There are several parameters in DeepCity, including the length of each walk (s), number of walks per node (r), dimension of learned features (d). We follow the settings of DeepWalk, i.e., $s = 80$, $r = 10$ and $d = 128$.

Results. The experimental results of four prediction tasks are presented in Tables 1, 2, 3 and 4. From the results, we observe that DeepCity for demographic and location category prediction achieves a superior performance, e.g., the AUC score for gender prediction is 0.95 for all cities, and it outperforms all the baselines across all prediction tasks: in most prediction tasks, DeepCity outperforms all baselines by at least 10%. This indicates that our DeepCity framework is promising on mining user check-ins.

	New York	Los Angeles	London
DeepCity	0.95	0.95	0.95
STL	0.85	0.83	0.73
MF	0.58	0.61	0.59
DeepWalk	0.68	0.65	0.67

Table 1: AUC on gender prediction.

	New York	Los Angeles	London
DeepCity	0.72	0.66	0.62
STL	0.38	0.36	0.32
MF	0.31	0.31	0.32
DeepWalk	0.46	0.44	0.37

Table 2: Macro-F1 on race prediction.

	New York	Los Angeles	London
DeepCity	0.53	0.52	0.54
STL	0.47	0.47	0.41
MF	0.34	0.33	0.32
DeepWalk	0.35	0.35	0.34

Table 3: Marco-F1 on age prediction.

	New York	Los Angeles	London
DeepCity	0.48	0.44	0.41
MF	0.25	0.21	0.27
SAP	0.29	0.31	0.20
DeepWalk	0.33	0.33	0.27

Table 4: Macro-F1 on location category prediction.

STL achieves a competitive performance. The reason why DeepCity outperforms STL in our experiments might be caused by the data sparseness. Also, the lack of locations' reviews and keywords may be another reason, since reviews and keywords contain resourceful information of each location which can be used to describe users' demographics. Moreover, DeepCity outperforming STL suggests that network embedding is another useful approach for mining location check-ins besides tensor decomposition. The matrix factorization approach, on the other hand, performs the worst as it does not take into account enough information. For location category prediction, DeepCity outperforms SAP, one reason might be that SAP's information retrieval feature set considers only one hop neighbors of each location, and the connections established by users in SAP can introduce many noises on grouping locations of the same category together. On the other hand, our method takes into account more than one hop neighbors of each location, i.e., the context size in Skip-gram (Mikolov et al. 2013a; 2013b), and the location category biased values of temporal-users mitigate the noises introduced on location connections.

We also observe that DeepCity outperforms DeepWalk in all prediction tasks, this indicates that task-specific random walk indeed increases the prediction performance over general network embedding methods, which validates our intuitions presented previously. This result should not only be limited to mining user check-ins, but also other data mining fields, such as profiling users through published statues in OSNs. However, even though without any adjustments, DeepWalk still achieves competitive performances, which further suggests network embedding's effectiveness.

Parameter sensitivity. DeepCity involves a number of parameters, we concentrate on the sensitivity of two of them, including the length of each walk (s) and number of walks

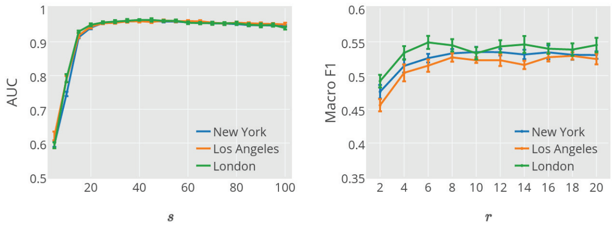

Figure 1: Parameter sensitivity study with gender (left) and age (right) prediction.

per node (r), Except for the parameter being tested, others are set to their default values. We utilize gender prediction to study the sensitivity of s. Figure 1 (left) shows that the prediction result (AUC) experiences a dramatic increase when s changing from 5 to 20, and then stay stable. Similarly, the number of walks starting from each node increases the performance of age prediction (Macro-F1) when shifted from 2 to 8 and saturate then (Figure 1 (right)). The reason that the performance gain by increasing r is not that large compared with s is that $s = 80$ in the default setting already provides Skip-gram with a large quantity data from embedding. We also study other parameters' sensitivity, such as dimension of learned features and context window size in Skip-gram (its default value is 10 following DeepWalk), the prediction results do not vary much and are omitted here.

Related Work

With the large amount of user check-in data being available, researchers have concentrated on mining these data. One direction is to use user check-ins to predict friendships, such as (Scellato, Noulas, and Mascolo 2011; Zhang and Pang 2015). Meanwhile, a few works have explored a user's friends information to predict his locations (Cho, Myers, and Leskovec 2011; Jurgens 2013; Pang and Zhang 2015). Recently, check-in data are used to profile users, e.g, the STL model (Zhong et al. 2015) adopted by us as a baseline for demographic prediction. On the other hand, many works have been conducted on reshaping our understandings of locations from the user aspects. For instance, researchers have used check-in data to measure the happiness (Quercia, Schifanella, and Aiello 2014), walkability (Quercia et al. 2015) and sociality (Pang and Zhang 2016) of locations, and find the similar neighborhoods across different cities (Falher, Gionis, and Mathioudakis 2015). All of these open up an emerging field, namely urban informatics or urban computing (Zheng et al. 2014).

Conclusion

In this paper, we propose a general framework, namely DeepCity, to learn features for user and location profiling. We propose a method, i.e., task-specific random walk, to guide the learning algorithms to embed users with similar demographics (locations with similar categories) closer in the resulted vector space. Experimental results on a large collection of Instagram data have demonstrated the effectiveness of DeepCity over other models.

References

Cho, E.; Myers, S. A.; and Leskovec, J. 2011. Friendship and mobility: User movement in location-based social networks. In *KDD*.

Falher, G. L.; Gionis, A.; and Mathioudakis, M. 2015. Where is the soho of Rome? Measures and algorithms for finding similar neighborhoods in cities. In *ICWSM*.

Felbo, B.; Sundsøy, P.; Pentland, A.; Lehmann, S.; and de Montjoye, Y.-A. 2015. Using deep learning to predict demographics from mobile phone metadata. *CoRR* abs/1511.06660.

Jurgens, D. 2013. That's what friends are for: Inferring location in online social media platforms based on social relationships. In *ICWSM*.

Kosinski, M.; Stillwell, D.; and Graepel, T. 2013. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences 110(15)*.

Manikonda, L.; Hu, Y.; and Kambhampati, S. 2014. Analyzing user activities, demographics, social network structure and user-generated content on Instagram. *CoRR* abs/1410.8099.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. In *ICLR*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionally. In *NIPS*.

Pang, J., and Zhang, Y. 2015. Location prediction: Communities speak louder than friends. In *COSN*.

Pang, J., and Zhang, Y. 2016. Quantifying location sociality. *CoRR* abs/1604.00175.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *KDD*.

Quercia, D.; Aiello, L. M.; Schifanella, R.; and Davies, A. 2015. The digital life of walkable streets. In *WWW*.

Quercia, D.; Schifanella, R.; and Aiello, L. M. 2014. The shortest path to happiness: recommending beautiful, quiet, and happy routes in the city. In *HT*.

Scellato, S.; Noulas, A.; and Mascolo, C. 2011. Exploiting place features in link prediction on location-based social networks. In *KDD*.

Souza, F.; de Las Casas, D.; Flores, V.; Youn, S.; Cha, M.; Quercia, D.; and Almeida, V. 2015. Dawn of the selfie era: The whos, wheres, and hows of selfies on Instagram. In *COSN*.

Ye, M.; Shou, D.; Lee, W.-C.; Yin, P.; and Janowicz, K. 2011. On the semantic annotation of places in location-based social networks. In *KDD*.

Zhang, Y., and Pang, J. 2015. Distance and friendship: A distance-based model for link prediction in social networks. In *APWeb*.

Zheng, Y.; Capra, L.; Wolfson, O.; and Yang, H. 2014. Urban computing: Concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology 5(3)*.

Zhong, Y.; Yuan, N. J.; Zhong, W.; Zhang, F.; and Xie, X. 2015. You are where you go: Inferring demographic attributes from location check-ins. In *WSDM*.