Mining Place-Time Affinity to Improve POI Recommendation

Junfei Wang, Darshan Bagul, Jun Chu, Lu Meng, Sargur Srihari
Department of Computer Science and Engineering
State University of New York at Buffalo
Buffalo, USA
e-mail: junfeiwa, dbagul, jchu6, lumeng, srihari@buffalo.edu

Abstract—**A Point of Interest(POI) is a location that one may find useful or interesting. POI recommendation is a key feature in location-based social networks (LBSNs). With the development of mobile devices and apps, POI recommendation becomes a very popular topic and it includes humongous data. Current models always suffer from the problem of data sparsity. In this paper we propose a novel transfer learning model to learn affinity between the time and places, and use the mined features to improve the performance of a content-based POI recommendation system. In particular, we use check-in data to learn latent vectors for time and place category features by non-negative matrix factorization. Then, the mined densely embedded features are input to a gradient boosting decision tree (GBDT) based pairwise scoring model, which is trained by the check-in data of another city, to do POI recommendation. We conduct our experiment on the Foursquare check-in dataset, and discover that the learned latent vectors can dramatically improve the performance of a POI recommendation system.**

Keywords-embedding; matrix factorization; recommendation system; transfer learning

I. Introduction

The explosive growth of location-based social networks (LBSNs) has inspired a lot of advances in the field of POI recommendation. The aim of POI recommendation systems is to recommend a structured location to a user, given multiple input features such as the hour of a day, the day of a week, distance, place category, etc.

Place visits are highly correlated to temporal features, such as the hour of a day, and the day of a week. For example, one will usually have a fixed schedule of working out, such as going to the gym during the weekend, or during specific time periods on weekdays. Figure 1 shows the number of visits to a gym in New York City during one weekday, by hours. To make the prediction, models must discover the relationship between place and time, sparse features, such as place category or the hour of a day, have to be introduced into our model. However, many current approaches just directly introduce the sparse features into the models, which is not an efficient method because sparse features do not usually bear a lot of information. Embedding, a method to convert a sparse vector to a lower dimensional latent vector is the viable approach to this problem and improving our POI recommendation system performance.

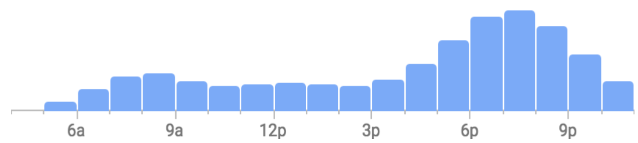

Figure 1. The popularity of a gym during a weekday, by hours. According to this graph, most people go to gym during 7 am to 10 am in the morning and 3 pm to 10 pm in the afternoon. This figure shows that place visits are highly correlated with time.

In this paper, we propose a novel hybrid model for POI recommendation. In particular, we first construct the time-category matrix whose entries are the check-in counts for a specific category at a given hour of a day, a day of a week. After that, we use non-negative matrix factorization to find latent vectors for the temporal features and the place category features, using a check-in dataset. Finally, with the learned dense latent vectors introduced, another non-overlapping check-in dataset is used to train a tree-based scoring model for scoring multiple POI candidates. We conduct our experiment on the Foursquare check-in dataset. We find out that with the embedded latent vectors, the POI recommendation system has significant performance gain over the baseline model.

The rest of this paper is organized as follows. Chapter 2 introduces some related work. Chapter 3 describes our proposed transfer learning method in details. The experiment process and results are in Chapter 4. Chapter 5 draws our conclusion and future plans.

II. Related Work

Recommender systems or recommendation systems are one kind of information filtering system that seek to predict the 'rating' or 'preference' that a user would give to an item. After the prediction process, we can then make a ranking based on the predicted ratings or preference, recommend those with highest ratings to users.

Broadly speaking, there are two kinds of recommendation systems, collaborative filtering based and model based [2]. Most collaborative filtering systems use user-item ratings as the only input; However, this approach suffers from the cold start problem and the scalability problem [2]. For example, Singular Value Decomposition (SVD) [3] is one of the matrix factorization methods that finds the latent vector for each user and item and uses the generated latent vectors to do prediction [4]. This kind of approach fails to address the case when a new user or item has been introduced to our system, due to its inability to address the system's new products and users [5]. Also, in a large scale recommendation system, the numbers of users and items are becoming overwhelmingly great, in this case, the computational cost of collaborative filtering based approaches is always beyond the acceptable level [6].

In contrast, content-based recommendation systems are able to make use of meta-data that are always associated with users and items, and scale to desired level with reasonable computational cost, thus are gaining more attentions. For example, Google has successfully built a neural network model to recommend videos to users on YouTube [7]. Facebook has deployed a hybrid model that combines a gradient boosted decision tree (GBDT) and logistic regression to predict ads clicks [8].

Many POI recommendation systems use content-based models. For example, [9] proposed a model to predict user's next check-in using various features such as check-in count and check-in history. [1] proposed a hybrid model to model both spatial and temporal locations. In [10], authors studied point of interests (POIs) recommendation based on user's current locations. They used metric embeddings to learn the transition between user's current and next location. Other efforts to implement location recommendation include using recurrent neural networks (RNN) [11], hidden Markov models (HMM) [12]. Some personalized approaches have also been proposed, such as personalized metric embeddings in [10] and personalized gradient boosting decision trees (GBDT) based recommendation [13].

In the context of sparse feature embeddings, the unsupervised approaches [14], [15] use unsupervised learning methods, such as neural network based auto encoders, and restricted Boltzmann machines to reconstruct the input using proper loss functions. Since the input and output are just one single sparse feature, this method always fails to learn the pairwise relationship between two sparse features. The n-gram based models [16], [17] are mainly used in natural language processing. An n-gram is a contiguous sequence of n items from a given sequence of text or speech [18]. This kind of model uses the neural network to map a single word to multiple words in an n-gram and thus it can learn similarities between two words. However, this kind of model suffers from long running times.

III. Our Hybrid POI Recommendation Model

A. Constructing Time-Category Matrix

This section describes the method to construct time-place category matrix as input to NMF algorithm. We generate the time-category matrix from check-ins in one city and apply to another city. The first step is to bucket the time in hours. For every single check-in ci, we use its associated time stamp to generate a tuple: {di : hi}, where 0 <= di <= 6 stands for the day of a week, and 0 <= hi <= 23 stands for the hour of a day. For example, tuple 0:23 means 11 pm at Sunday night.

In the second step, for every place category, we count the number of check-ins for each category and time pairs. We end up with a matrix of size n x m, where n = 168, which is the number of hours in a week, and m is the number of categories in our dataset.

B. Generate Embeddings Using NMF

We now apply NMF algorithm to decompose the time-category matrix into two matrices with reduced sizes, whose product can approximate the original matrix. The resulting two matrices can then be used as embedding matrices for time and category features. The approximation process involves minimizing the Frobenius norm of $\| \mathbf{R} - \mathbf{N} \|$. $\mathbf{N}$ is a rank-k approximation of $\mathbf{R}$. The Frobenius norm of a m x n matrix $\mathbf{R}$ is defined as follows:

$$||\boldsymbol{R}||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} R_{ij}^2}$$

We choose the multiplicative method proposed in [23] to do the learning. In a nutshell, after random initialization of W and H, we iteratively update them using the following two equations until convergence.

$$W \leftarrow W \circ \frac{RH'}{WHH'}$$

$$H \leftarrow H \circ \frac{W'R}{W'WH}$$

C. GBDT-Based Pairwise Scoring Model

Now we introduce our gradient boosting decision Tree (GBDT) based scoring model. We build a binary classification model to predict if a user will go to a certain POI or not given various kinds of features of user and candidate locations. The features that we use are shown in the Table 1. We manually calculated some of these features, such as the check-in count of every place, and distance to the user.

TABLE I. Features Used in our GBDT Scoring Model. Our GBDT Based Pairwise Scoring Model Takes These Features as Input and Calculate a Score that Indicates How Likely an User will go to a Candidate Place

User Features	Candidate Location Features
User ID	Place ID
Gender	The hour of a day
Twitter friend count	The day of a week
Twitter follower count	Place category
	Check-in count
	Distance to the user

Our original check-in dataset only contains positive samples that are check-ins. Our model is a scoring model, so we need negative samples for training and evaluation process. Strong negative samples are essential to the success of our approach. Since location is one of the most important factors to consider before visiting, we use neighbouring places as counter-examples. We experimented with several different settings and finally chose 9 closest places to the positive check-in place as negative samples. An example of our negative data generation process is shown in Figure 2.

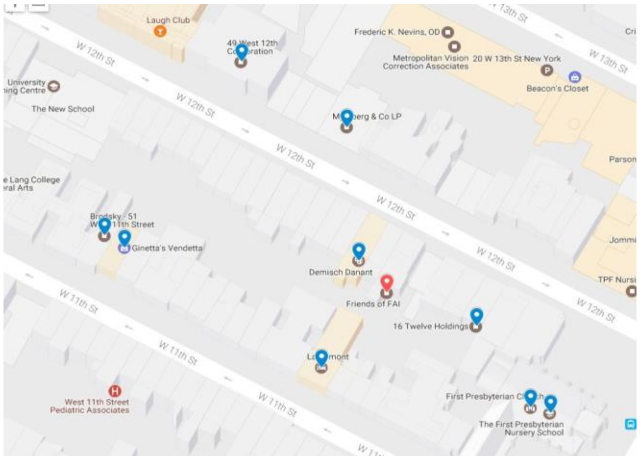

Figure 2. This figure explains our method to generate negative data for our scoring model. In this figure, the place with a red pin is the positive check-in location, and other 9 places (in blue) are negative samples generated.

IV. EXPERIMENT

In this section, we evaluate the influence of embedded sparse features on our POI recommendation model, with different embedding dimensions. We also implement a baseline model, that is, the GBDT model that directly takes sparse time and place category as its input. We use the Foursquare dataset [1], [24], [25] throughout this paper. This dataset contains check-ins in NYC and Tokyo collected from April 2012 to February 2013. It contains 227,428 check-ins in New York City and 573,703 check-ins in Tokyo. Each entry is associated with a check-in timestamp, its raw GPS location and its category, such as restaurant, park, etc. proceedings. proceedings.

A. *Detailed Experiment Setup*

We now explain our experiment process in detail. Our pairwise scoring model models the relationship between users and POIs using their features. We use the mean reciprocal rank (MRR) as the metric to quantify the quality of our generated embeddings. following.

MRR is a metric to evaluate any process that produces a list of possible responses to a sample of queries, ordered by the probability of correctness. The MRR for queries Q can be calculated by the following equation:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

where $rank_i$ refers to the rank position of the first relevant document for the i-th query. To evaluate our model performance, we split the original training set into a training set and a test set. During the evaluation process, we score all ten candidates, sort them by the scores, and then calculate the MRR. We also calculate Precision-Recall curve for our model, then compute the area-under-curve metric (AUC) as our second metric, as shown in Figure 3.

In our experiment, we use two non-overlapping datasets for two stages of our model: Tokyo for NMF, and NYC for the scoring model. Using non-overlapping dataset for each step guarantees the robustness of our model. First of all, we use Tokyo check-in data to generate a time-category matrix, then use NMF to produce the embeddings. We choose various embedding dimensions, which are 8, 16 and 32. An example is shown in Listing 1.

After the matrix factorization stage, we end up with a latent representation of time feature and place category features. Now we are ready to evaluate their impact on our pairwise scoring model.

```
"hour of a day": 01,
"day of a week": 06,
"embedding": [0.15729017555713654 ,0.514305
7107925415,
0.4689531624317169 ,0.3259933888912201,
0.8134676218032837 , 1.0409616231918335 ,
0.31192076206207275 ,0.13485203683376312]
```

Listing 1. An example of our time features

TABLE II. AUC AND MRR FOR THE BASELINE MODEL AND OUR APPROACH WITH DIFFERENT EMBEDDING DIMENSION K. WE CAN GET THE BEST PERFORMANCE WHEN K=32.

Model	**Area Under Curve(AUC)**	**Mean Reciprocal Rank(MRR)**
Baseline	0.541	0.46
k=8	0.591	0.538
k=16	0.610	0.604
k=32	0.655	0.713

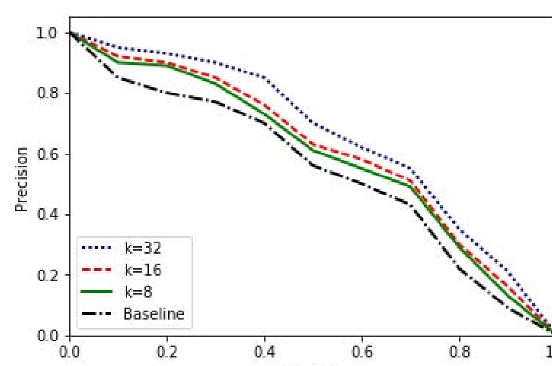

Figure 3. Precision-recall curve for our baseline model and our proposed method with different embedding dimension k. With embedded time and place category features, our model significantly improves the recommendation performance.

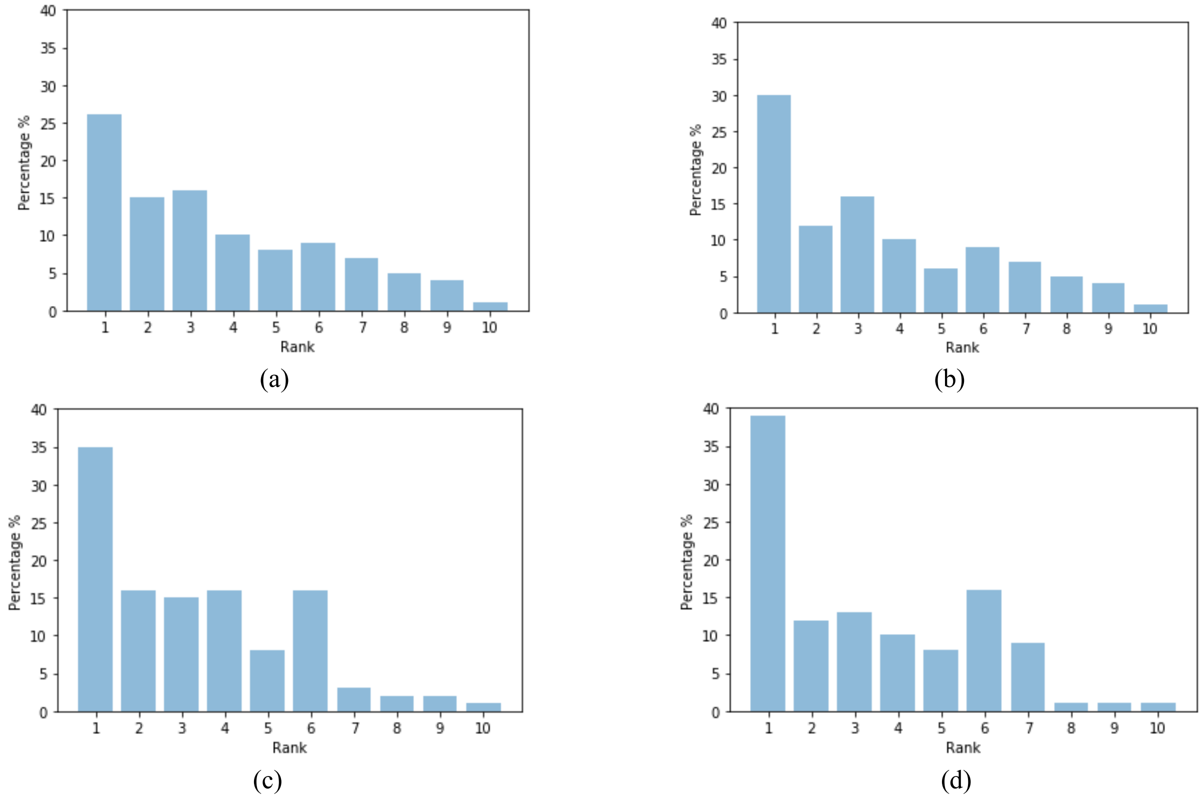

Figure 4. Rank of the positive label in our evaluation set for (a) baseline mode, (b) k=8, (c) k=16 (d)k=32. With increasing embedding dimension k, more entries in evaluation set are ranked at 1st place. The generated embeddings carry more information and thus the scoring model performs better compared to the baseline.

As our baseline model, we directly feed in our sparse features as listed in Table 1, train and evaluate our GBDT model on NYC check-in dataset with generated negative data entries. The precision-recall curve is shown in Figure 3. The AUC improved from 0.541 to 0.655 in this experiment. The MRR score of our model improved from 0.46 to 0.713, details are shown in Table 2. We also plotted the ranking distribution of our baseline and the proposed model; it is shown in Figure 4.

According to our experiment, we conclude that our generated embeddings can significantly improve the overall performance of the location recommendation system.

V. Conclusion and Future Work

We propose a novel transfer learning model that provides better POI recommendations than traditional content-based models. In particular, we first convert our sparse features to dense vectors by non-negative matrix factorization and then feed the resulting vectors into the scoring model as features. According to the experiment results, our approach outperforms the baseline model that directly accept un-embedded sparse features as input. Our future plan involves embedding model sparse features other than time and place categories. Also, we want to try embedding by the neural network and compare its performance with our current model.

References

[1] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, p. 4, 2009.

[2] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular value decom- position and principal component analysis," in *A practical approach to microarray data analysis*. Springer, 2003, pp. 91–109.

[3] S. Gower, "Netflix prize and svd," 2014.

[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, 2009.

[5] M. Papagelis, I. Rousidis, D. Plexousakis, and E. Theoharopoulos, "Incremental collaborative filtering for highly-scalable recommendation algorithms," in *International Symposium on Methodologies for Intelli- gent Systems*. Springer, 2005, pp. 553–561.

[6] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 2016, pp. 191–198.

[7] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers *et al.*, "Practical lessons from predicting clicks on ads at facebook," in *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. ACM, 2014, pp. 1–9.

[8] J. Chang and E. Sun, "Location 3: How users share and respond to location-based data on social networking sites," in *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 2011, pp. 74–80.

[9] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 129–142, 2015.

[10] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new poi recommendation." in *IJCAI*, 2015, pp. 2069–2075.

[11] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts." in *AAAI*, 2016, pp. 194–200.

[12] W. Mathew, R. Raposo, and B. Martins, "Predicting future locations with hidden markov models," in *Proceedings of the 2012 ACM confer- ence on ubiquitous computing*. ACM, 2012, pp. 911–918.

[13] L. H. Tran, M. Catasta, L. K. McDowell, and K. Aberer, "Next place prediction using mobile data," in *Proceedings of the Mobile Data Challenge Workshop (MDC 2012)*, no. EPFL-CONF-182131, 2012.

[14] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[15] Y.-l. Boureau, Y. L. Cun *et al.*, "Sparse feature learning for deep belief networks," in *Advances in neural information processing systems*, 2008, pp. 1185–1192.

[16] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification." in *ACL (1)*, 2014, pp. 1555–1565.

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[18] Wikipedia, "N-gram — Wikipedia, the free encyclopedia," http://en.wikipedia.org/w/index.php?title=N-gram, 2017, [Online; accessed 20-April-2017].

[19] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[20] K. Chen, R. Lu, C. Wong, G. Sun, L. Heck, and B. Tseng, "Trada: tree based ranking function adaptation," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 1143–1152.

[21] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[22] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[23] Wikipedia, "Non-negative matrix factorization — Wikipedia, the free encyclopedia," http://en.wikipedia.org/w/index.php?title=Non-negative%20matrix%20factorization&oldid=776644650, 2017, [Online; accessed 09-May-2017].

[24] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[25] D. Yang, D. Zhang, and B. Qu, "Participatory cultural mapping based on collective behavior data in location-based social networks," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 7, no. 3, p. 30, 2016.

[26] D. Yang, D. Zhang, B. Qu, and P. Cudré-Mauroux, "Privcheck: privacy- preserving check-in data publishing for personalized location based ser- vices," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 545–556.