ContextMF: A Fast and Context-aware Embedding Learning Method for Recommendation Systems

Junfei Wang* Darshan Bagul† Sargur Srihari‡

Abstract

In location based social network systems, it is essential to recommend place of interests(POIs) to users. With the development of mobile devices and apps, POI recommendation becomes a prevalent topic. Current POI recommendation models suffer from either the problem of lacking contextual signals or high computational complexity. In this paper we propose ContextMF, a novel linear embedding method for sparse contextual features, thus, allowing us to quickly convert these sparse features to latent vector space while still preserving the comparable embedding quality to neural network embedding models. In particular, we use mathematical add and dot-product operations instead of expensive matrix concatenation and nonlinearities. We conduct our experiment on the Foursquare check-in dataset, the mined densely embedded features are input to a gradient boosting decision tree (GBDT) based pairwise scoring model, which is trained by another portion of check-in data, to make POI recommendation. The experiment results prove the effectiveness and robustness of our method.

1 Introduction

With the rapid growth of online business websites and social networks; recommender systems which aim at recommending structured locations to persons, are receiving much attention in machine learning research.

Contextual signals are data sources associated with users and places involved in POI recommendation systems. Some examples are place category, user home location, check-in time, etc. The explosive growth of social networks (SNS) has generated a huge amount of contextual signals, and they have potential to be used in POI recommendation systems.

Recommendation systems also need to be fast enough to give real-time response to users. For example, Netflix [1] requires 200ms maximum response time for every single query. The goal of better incorporating contextual signals into large-scale model-based recommendation systems rises from the demand for fast and robust recommendation model.

Most recommendation systems use matrix factorization methods to calculate the latent vector of each user and item given the existing user-item ratings and use these embedded vectors to predict the future ratings, [7]. However, this kind of approach fails to account for the contextual signals that are ubiquitous in recommendation dataset. More recently, some context-aware neural network models have been proposed to incorporate the contextual information into recommendation systems [5]. However, neural network based recommendation systems always receive sparse features as a huge part of their input. Directly inputting sparse features into the prediction model always leads to poor performance because the sparsity prevents the model from learning useful info from these features. Moreover, neural network models suffer from high computational complexity caused by multiple levels of abstraction and nonlinearities.

In this paper, we propose a fast and context-aware bag of features model for POI recommendation. In particular, we first combine different contextual sparse features for every user and every place using mathematical add operation. We then use dot-product to calculate the similarity between each user and place. In the end, the embeddings for each sparse feature is trained using traditional stochastic gradient descent. In order to validate our embedding quality, a GBDT based pairwise scoring model is trained using our generated user and place embedding vectors as features. 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. Additionally, we compare the run-time of our model and traditional neural network model. The results show that our approach has significant advantage over neural network models regarding computational complexity.

The rest of this paper is organized as follows. Chapter 2 introduces some related work in this domain. Chapter 3 describes our proposed transfer learning method in details. The experiment process and results

*State University of New York at Buffalo.
†State University of New York at Buffalo
‡State University of New York at Buffalo

are in Chapter 4. Chapter 5 draws our conclusion and states future plan.

2 Related Work

There are two kinds of recommendation systems, collaborative filtering based and model-based [12]. 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 [12]. More importantly, this kind of recommendation system always fails to incorporate the contextual information associated with the entities involved in the dataset.

In contrast, content-based recommendation systems can make use of meta-data that is always associated with users and items. Neural networks are widely deployed in content-based recommendation systems. For example, Google has successfully built a neural network model to recommend videos to users on Youtube [5]. Despite the proved advantages over traditional methods, neural network models always suffer from the problem of high computational complexity and may not be served online without powerful hardware [13].

Some advances have been made in the field of POI recommendation as well. For instance, [4] proposed a model to predict user's next check-in using various features such as check-in count and check-in history. [17] proposed a hybrid model to model both spatial and temporal locations. In [6], 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) [9] and hidden Markov models (HMM) [10]. Some personalized approaches have also been proposed, such as personalized metric embeddings in [6] and personalized gradient boosting decision trees (GBDT) based recommendation [14].

In the context of sparse feature embeddings, the unsupervised approaches [8] [3] use unsupervised learning methods, such as neural network based autoencoders, 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 [11] are mainly used in natural language processing.

3 Our ContextMF Model

This section describes our ContextMF model in detail.

3.1 Data Preprocessing The first step is to prepare the check-in dataset. For every single check-in c_i, we use its associated user id and place id to generate a tuple: $\{u_i : p_i\}$, Where u_i stands for the user id, and p_i stands for the place id. For example, tuple 0:1 means user 0 checked in at place 1.

In the second step, for every user and place, we find the associated contextual data. The features that we use are shown below. We use Apache Hive [2] to do

User Features	Candidate Location Features
User ID	Place ID
Gender	The hour of a day
	The day of a week
	Place category

data preprocessing. An example of our resulted data is shown in Listing 1.

```
"user_id" : 04019569,
"place_id" : 04019569,
"user_gender" : 'Male',
"place_category" : 'Restaurant',
"hour_of_a_day" : 01,
"day_of_a_week" : 06
```

Listing 1: An example of our resulted data. This means that male user with id 04019569, checked in at place with id 04019569, at 1am, on a Sunday.

3.2 Linear Bag of Features Embedding Model

We now describe our linear bag of features embedding model in detail. In our model, for every user-page pair, the embeddings of every sparse feature are combined using add operation, where c_i is embedding of i-th sparse feature for a page. After combining the bag of features, the similarity between a pair of locations is computed as the inner product of the combined latent feature vectors. The square loss is used between the similarity values and the ground truth labels as follows.

$$\mathbf{L} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y_i}) \tag{3.1}$$

Where n is the total number of training samples, Y_i stands for the correct label of i-th sample. The structure of our model is shown in Figure 1. Finally, our model is trained using back-propagation and gradient descent. Without the expensive multiple levels of nonlinearities and matrix concatenations, we can achieve much faster training and inference.

3.3 GBDT-based Scoring Model Now we introduce our gradient boosting decision Tree (GBDT) based scoring model. We build a binary classification model

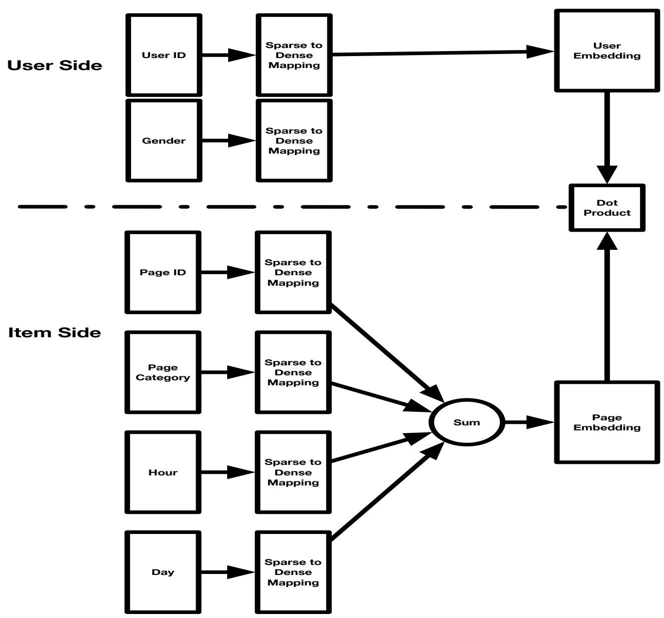

Figure 1: The architecture of our online embedding model. We use simple mathematical add and dot product to ensure the training can be finished in real-time.

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.

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 nearby places as counter-examples. We experimented with several different settings and finally chose nine closest places to the positive check-in place as negative samples.

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

Table 1: 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 a user will go to a candidate place.

4 Experiment

In this section, we evaluate the influence of sparse embedded 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 [17] [15] [16] 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.

4.1 Detailed Experiment Setup We now explain our experiment procedure 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.

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:

$$MRR = \frac{1}{\mid Q \mid} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{4.2}$$

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.

4.2 Experiment Results As our baseline model, we directly feed in our sparse features as listed in Table 1, train and evaluate our GBDT model on 75% and 25% random chosen NYC and Tokyo check-in dataset with generated negative data entries. The MRR score of our model improved from 0.46 to 0.713, details are shown in Table 2. We also calculated the precision-recall curve for our pairwise scoring model with embedding dimension k = 32; it is shown in Table 2. The AUC increases from 0.15487 to 0.22832. Additionally, with embedding dimension k = 32, we create a 5-layer neural network model and compare its training time with that of our model. Our model takes 32.7 seconds to finish 1 epoch whereas the neural network model takes 90.1 seconds.

5 Conclusion and Future Plan

We propose a novel linear bag of features embedding learning model that provides better embedding quality

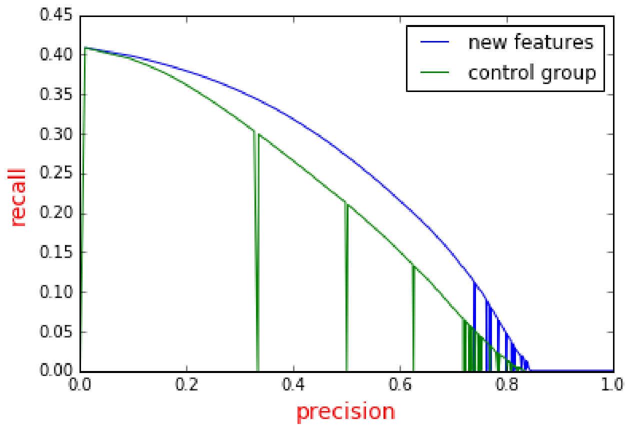

Figure 2: The precision-recall curve of our pairwise scoring model compared to the control group.

Model	Mean Reciprocal Rank(MRR)
Baseline	0.33
k=8	0.493
k=16	0.540
k=32	0.662

Table 2: MRR for the baseline model and our approach with different embedding dimension k. We can get the best performance when k=32.

than traditional matrix factorization method and better computational complexity than neural network models. According to the experiment results, our approach outperforms the baseline model that directly accept unembedded sparse features as input. Our future plan involves comparing the embedding quality of our model and neural network based models.

References

[1] Netflix Technology Blog. System architectures for personalization and recommendation, Mar 2013.

[2] Dhruba Borthakur, Jonathan Gray, Joydeep Sen Sarma, Kannan Muthukkaruppan, Nicolas Spiegelberg, Hairong Kuang, Karthik Ranganathan, Dmytro Molkov, Aravind Menon, Samuel Rash, et al. Apache hadoop goes realtime at facebook. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 1071–1080. ACM, 2011.

[3] Y-lan Boureau, Yann L Cun, et al. Sparse feature learning for deep belief networks. In *Advances in neural information processing systems*, pages 1185–1192, 2008.

[4] Jonathan Chang and Eric 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*, pages 74–80, 2011.

[5] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 191–198. ACM, 2016.

[6] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. Personalized ranking metric embedding for next new poi recommendation. In *IJCAI*, pages 2069–2075, 2015.

[7] Stephen Gower. Netflix prize and svd. 2014.

[8] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[9] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*, pages 194–200, 2016.

[10] Wesley Mathew, Ruben Raposo, and Bruno Martins. Predicting future locations with hidden markov models. In *Proceedings of the 2012 ACM conference on ubiquitous computing*, pages 911–918. ACM, 2012.

[11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[12] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.

[13] Hokchhay Tann, Soheil Hashemi, R Bahar, and Sherief Reda. Runtime configurable deep neural networks for energy-accuracy trade-off. In *Proceedings of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, page 34. ACM, 2016.

[14] Le Hung Tran, Michele Catasta, Lucas Kelsey McDowell, and Karl Aberer. Next place prediction using mobile data. In *Proceedings of the Mobile Data Challenge Workshop (MDC 2012)*, number EPFL-CONF-182131, 2012.

[15] Dingqi Yang, Daqing Zhang, and Bingqing Qu. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(3):30, 2016.

[16] Dingqi Yang, Daqing Zhang, Bingqing Qu, and Philippe Cudré-Mauroux. Privcheck: privacy-preserving check-in data publishing for personalized location based services. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 545–556. ACM, 2016.

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