

# 基于 LSTM 的 POI 个性化推荐框架<sup>①</sup>

王立, 张 谧

(复旦大学 软件学院, 上海 201203)

(复旦大学 上海市智能信息处理重点实验室, 上海 201203)

**摘 要:** 近年来, 随着基于位置的社会网络 (Location-Based Social Network, LBSN) 热度的不断增加, 为用户推荐下一个 POI (Point-Of-Interests) 也显得越来越重要. 而对应的各种应用搜集到的用户的行为时间、地理、好友以及标签等信息的增多, 使得 POI 推荐变得更加容易. 目前针对 POI 推荐, 已经有部分算法提出, 但是他们受限于自身的局限性, 还都不能很好的解决这个问题, 例如, 个性化马尔科夫链 (Factorizing Personalized Markov Chain, FPMC)、张量分解 (Tensor Factorization, TF)、RNN (Recurrent Neural Networks) 等. 但是, 这些模型由于其本身缺陷, 都不能完美的糅合 POI 场景中的所有信息. 在这篇文章中, 我们扩展了长短时记忆循环神经网络 (Long-Short Memory recurrent neural networks, LSTM), 提出一种全新的推荐框架 POI-LSTM 来解决 POI 推荐问题. POI-LSTM 借鉴 Embedding 的思想, 对用户信息、好友关系、POI 信息和评论信息进行向量化后, 输入到神经网络中, 同时利用 LSTM 捕捉用户的兴趣特征和兴趣的变化趋势, 最终能够在不同的输入层拟合社交网络信息和语义信息, 同时利用用户历史行为的时间和地理位置信息来为用户推荐下一个兴趣点.

**关键词:** 推荐系统; LSTM; POI Embedding; POI 推荐; LBSN

引用格式: 王立, 张谧. 基于 LSTM 的 POI 个性化推荐框架. 计算机系统应用, 2018, 27(12): 56-61. <http://www.c-s-a.org.cn/1003-3254/6530.html>

## LSTM-Based Neural Network Framework for Next POI Recommendation

WANG Li, ZHANG Mi

(School of Software, Fudan University, Shanghai 201203, China)

(Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai 201203, China)

**Abstract:** As Location-Based Social Network (LBSN) services become increasingly popular, next Point-Of-Interests (POI) recommendation emerges as one of many important applications of LBSNs. With the growing ability of collecting information, more and more temporal, spatial, social contextual and semantic tags information is collected in systems, which makes the location prediction problem becomes feasible. Some works, like Factorizing Personalized Markov Chain (FPMC), Tensor Factorization (TF), Recurrent Neural Networks (RNN), etc., have been proposed to address this problem, but they all have their limitations. In this study, we extend Long-Short memory recurrent neural networks (LSTM) and propose a novel method called POI-LSTM. POI-LSTM can model social contextual and semantic tags information in each layer, and employ temporal and spatial contexts in more efficient way. Experimental results show that the proposed POI-LSTM model yields significant improvements over the competitive compared methods on two typical datasets, i.e., Yelp and Foursquare dataset.

**Key words:** recommender system; LSTM; POI Embedding; POI prediction; Location-Based Social Network (LBSN)

<sup>①</sup> 收稿时间: 2018-01-17; 修改时间: 2018-02-09; 采用时间: 2018-02-28; csa 在线出版时间: 2018-12-03

近年来,基于位置的社交网络(Location-Based Social Network, LBSN)应用飞速增长,典型的基于位置的社交网络有大众点评、微博、Foursquare、Yelp等,而人们也越来越习惯利用在线应用获取信息.这使得我们可以更方便的收集到用户的好友信息和历史行为序列信息,这些社交、地理位置、时间信息都为精细化推荐提供了更大的可能性.然而,要根据有限的信息完全准确的预测出单个用户在一个具体的时间节点出现在特定位置上的特定商家还是一件极具挑战性的事情.但是,一旦我们可以准确的预测出用户的行为序列,除了将推荐结果应用在现有的情景外,还可以推广到其余众多的领域,例如预测交通状况,根据人们的行为趋势,可以精准的定位交通堵塞发生的时间和地点,并提前启动预防机制.

在机器学习领域,针对基于地理和时间信息 POI 推荐,不断有新的推荐方法提出.这些推荐主要分为四类,分别是马尔科夫模型、张量分解模型、RNN 深度学习模型和地点向量化模型.FPMC 是普通马尔科夫模型的扩展,能够针对不同个体提供个性化推荐,FPMC 通过为每个用户设置马尔科夫链俘获用户的长期喜好和近期关注,同时可以通过张量分解解决转移矩阵数据匮乏的问题.基于其特性,这种模型能够很好的对序列进行建模,FPMC 也被成功的应用在 POI 推荐上<sup>[1-5]</sup>.TF 张量分解模型也在解决 POI 问题上得到应用,TF 假设每一维信息为一个张量,例如用户就是 U 矩阵,商户是 V 矩阵,时间是 T 矩阵等,通过训练得到这些矩阵就可以获得用户、商户、时间等不同维度之间的关系.因为 TF 可以将很多非结构化的信息键入模型,所以在基于时间<sup>[6]</sup>和基于地理位置信息<sup>[7,8]</sup>的模型在 POI 推荐上都有广泛应用.而在深度学习在 NLP 和图像领域大放异彩的时候,RNN 在 POI 推荐问题上也有所应用<sup>[9]</sup>,RNN 模型能够很好的对序列信息建模,同时还不需要强先后依赖关系假设,在 POI 推荐问题上相比 FPMC 有更好的表现.此外,很多学者<sup>[10-12]</sup>也尝试利用 Word Embedding 类似的方式解决 POI 推荐问题,将商户等转化成为低维空间向量,然后基于向量之间的相似度为用户推荐下一个兴趣点.

虽然上述模型在一些特定的场景内取得了令人惊喜的效果,但是它们并不能处理社交信息,也不能衡量用户和 POI 之间的语义相似度.譬如,一个用户会根据商户的标签信息以及朋友圈内朋友的评价信息进行选择.

本文提出一种基于 LSTM 对 POI 和用户同时做

Embedding 的框架 POI-LSTM,将用户的评价信息、好友关系、历史访问信息转化成为向量后,输入到 LSTM 内预测用户下一个兴趣点.相比现有的向量化模型而言,能够捕捉更多的序列依赖信息;而相比现有的 RNN 模型,能够更好的处理复杂的社交关系和语义信息.

后文组织如下:第 1 节介绍 LSTM 算法及 Embedding 的相关背景知识;第 2 节介绍 POI-LSTM 框架的实现细节;第 3 节介绍数据集以及实验结果;第 4 节进行总结.

## 1 相关背景

本节主要介绍和本模型密切相关的 LSTM 网络结构以及 Embedding 算法.

### 1.1 LSTM 网络结构介绍

Long Short Term Memory 网络是一种 RNN 特殊的类型,可以学习长期依赖信息.LSTM 自提出以来,已经被应用很多场景中,也取得了巨大的成功.

LSTM 和 RNN 一样,通过重复神经网络模块的链式形式来对序列数据进行学习.同时为了避免 RNN 的梯度爆炸和梯度弥散问题,LSTM 通过在 RNN 的重复模块中添加遗忘门、输入门和输出门来增加长序列的记忆问题.LSTM 中重复模块的结构如图 1.

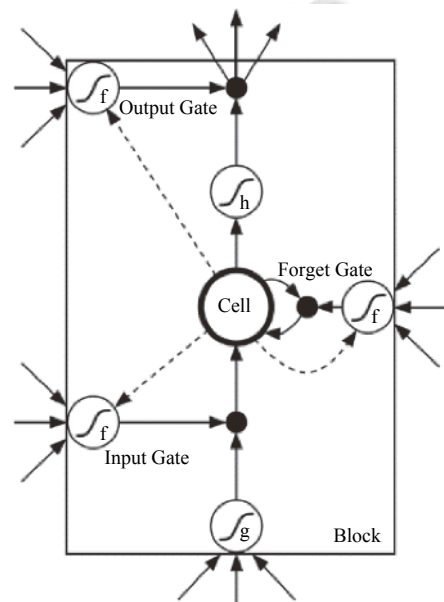


图 1 LSTM 模块结构

图 1 中,Cell 是 LSTM 模型维护的状态,包含了当前时间节点以前所有的信息;Input Gate 用来控制当前

输入信息保留概率; Forget Gate 用来控制根据当前的输入和状态来看, Cell 含有的信息保留的概率; Output Gate 用来控制当前时刻的信息的输出概率; 通过三个控制门的控制, LSTM 可以不断的吸收输入信息, 更新自己的状态, 并控制输出. LSTM 的向前传播公式如下:

$$\text{Input Gate: } I_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\text{Forget Gate: } F_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$\text{Output Gate: } O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$\text{Kept State: } C'_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

$$\text{Cell State: } C_t = C_{t-1} * F_t + C'_t * I_t$$

$$\text{Output: } H_t = C_t * O_t$$

其中,  $\sigma$  表示 sigmoid 函数,  $W$  表示神经网络中的权重,  $b$  表示偏置项,  $\tanh$  为输出的激活函数.

LSTM 可以采用梯度下降法进行更新, 最终使得损失函数值最小.

## 1.2 Embedding 模型介绍

Word Embedding 算法, 也被称为 Word2Vec 算法<sup>[11]</sup>, 是由 Bengio 等于 2003 年提出的一个三层神经网络的自然语言模型, 主要分为 Skip-Gram 和 CBOW 两种模型. Word2Vec 通过神经网络拟合一个词序列的条件概率  $p(w_t | w_1, w_2, \dots, w_{t-1})$ , 从而将文本词语转化为向量空间中的向量, 在模型训练完成后, 词向量之间的相似度具有一定的语义相似度. 整个模型的网络结构如图 2 所示.

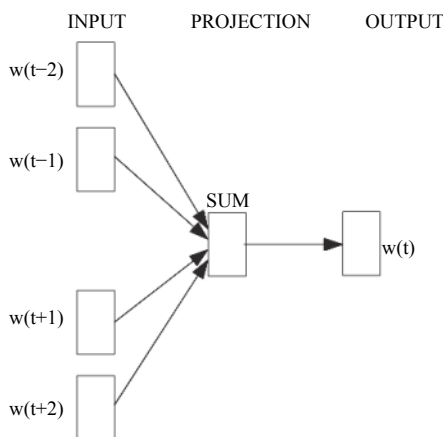


图2 CBOW 模型网络结构

从图中可以看出, Word2Vec 模型主要分成两部分, 前半部分在词向量矩阵中根据词语的序号寻找词向量, 后半部分将词向量经过  $\tanh$  隐藏层和 softmax 的激活输出层得到输出的词概率估计.

训练好的 Word2Vec 模型具有一定的语义相似度, 例如  $China - Beijing \approx French - Paris$ . 正是因为这样的相似度, 我们可以根据预测得到的向量和已有兴趣点向量之间的相似度来预测下一个兴趣点.

## 2 POI-LSTM 模型

现有的 LSTM 模型和 Embedding 模型主要问题在于都只能模拟单源信息. 在 POI 推荐场景中, 包含大量的社交信息和语义信息, 用户并非单纯的从一个地点转移到另外一个地点, 用户除了遵从自身的喜好外, 还可能会受到朋友的影响, 而且用户对以前访问过的 POI 并非都持有喜爱的态度, 这时候用户以往的评论信息就显得至关重要. 单纯的 LSTM 模型只能对用户的行为序列进行建模, 将用户访问过的地点信息输入到网络中, 只能捕捉到用户的访问地点兴趣信息, 不加入用户的历史评论信息就不能精细描述用户的兴趣变化, 同时也不能接受用户好友关系对于用户选择的影响. 而单纯的使用 Embedding 的话, 现有的推荐模型能够独立将用户和好友、POI 以及评论分别进行向量化, 应用到 POI 推荐问题上. 单独使用其中一种信息都不能很好的进行精细化推荐, 但是要是训练三种向量的话, 怎么均衡三者之间的关系, 即使用权重和比例上很难把握, 往往会使得模型训练坍塌, 就算能够训练成功, 也很难在测试集合上成功应用. 本模型则采用端对端的训练方式, 将用户及其好友关系作为固有属性, 将其历史评论信息向量化后结合当时访问的 POI 信息一同输入 LSTM 网络中得到用户兴趣信息作为动态属性, 然后三者结合为用户推荐下一个地点.

本节将从数据的问题定义、模型网络结构以及损失函数三方面进行描述.

### 2.1 问题定义

针对 POI 推荐问题, 我们定义  $P$  为用户集合,  $Q$  为 POI 集合,  $T$  为标签集合. 另外指定  $p_u \in R^d$ ,  $q_v \in R^d$ ,  $s_k \in R^d$  表示用户  $u$ 、兴趣点  $v$  和标签  $s$  的  $d$  维向量. 每个兴趣点  $v$  都有它的经纬度信息  $\{x_v, y_v\}$  和标签集合  $\{s_1^v, s_2^v, \dots\}$ . 每个用户  $u$  绑定有其标签集合  $\{s_1^u, s_2^u, \dots\}$ , 朋友列表  $\{f_1^u, f_2^u, \dots\}$  以及历史访问记录  $\{q_{t_1}^u, q_{t_2}^u, \dots\}$ , 其中  $q_{t_i}^u$  表示用户  $u$  在时间  $t_i$  访问了兴趣点  $q$ . 所有用户的历史行为序列表示为  $Q^u = \{Q^{u_1}, Q^{u_2}, \dots\}$ . 给定了用户的历史行为记录信息和相关的好友信息, 模型的任务就是预测下一次用户最有可能去的兴趣点. 数据源结构如图 3.

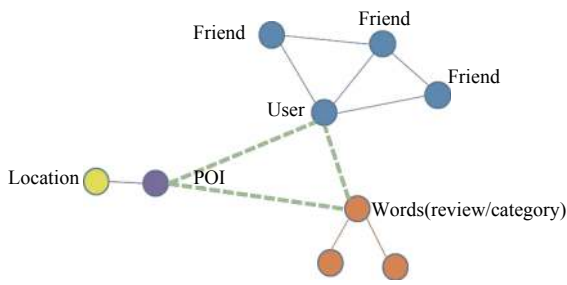


图3 数据源结构

## 2.2 模型框架

模型整体网络架构如图四所示, 其中直角方块部分代表了模型的四种源输入, 最左侧方框代表用户的向量输入, 中间的方框部分代表用户的朋友向量输入, 右上侧方框代表用户对兴趣点的评价信息, 右侧中间方框包含了兴趣点的向量和地理位置 (经纬度分别归一化到正负一之间) 输入. 下侧圆角虚线方框代表了模型的经过卷积和 LSTM 层提取的高维特征. 最右下方的圆角实线方块则代表了模型的输出部分.

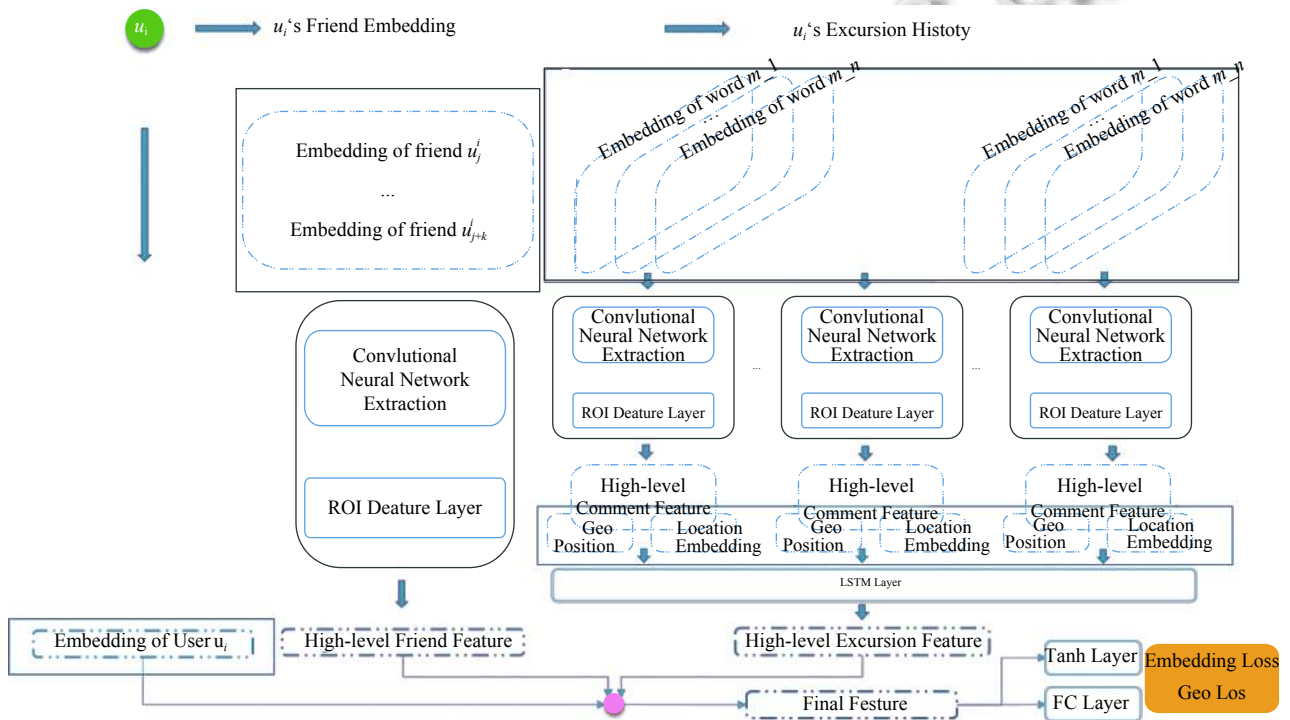


图4 POI-LSTM 网络结构

用户的向量输入和 Word2Vec 的输入相似, 输入 one-hot 的向量作为输入, 然后经过 Embedding 层得到该用户的向量. 相似的, 朋友的向量是  $k$  个 one-hot 向量作为输入, 然后得到  $k$  个朋友的向量作为输入, 在实现的时候直接输入  $k$ -hot 的向量, 得到代表关系网的向量作为该层的输入. 对于该用户对每个兴趣点的评价信息, 使用单词对应的预训练 (基于 glove 数据集) 好的向量作为输入, 然后经过卷积层和 ROI 池化层将评价信息压缩到一维向量作为 LSTM 的部分输入. 此向量和兴趣点的地理位置向量以及兴趣点的向量拼接起来将作为 LSTM 层的输入.

不同源的输入经过上述的过程, 已经提取成为高

维的特征, 分别是用户自身的特征、社交特征、评价和行为序列特征. 这些特征拼接起来后, 经过激活层即可得到下一次将要访问的兴趣点的向量表示. 同时为了保证模型的准确性, 我们还将高维的特征经过全连接层输出一个地理位置信息和访问的排序信息, 以确保访问的地点和下一次访问的地点之间地理位置偏差较小.

特别说明, 对于不同源的低维度特征模型采用不同的神经网络层来提取高维特征. 用户特征直接接入最后的全连接和激活层. 社交特征即用户的好友的信息通过 CNN 提取高维特征后经过 ROI 池化层转化为固定长度的高维特征. 用户对兴趣点的评价特征, 和

社交信息一样,需要经过 CNN 层提取高维特征后经过 ROI 池化层得到固定长度的高维特征,之后<评论特征,地理位置,兴趣点特征>将作为一个整体输入 LSTM 网络中作为用户的历史行为特征,因为 LSTM 可以很好的表示用户的长期兴趣,还能够精确的捕捉到用户近期的兴趣特征.高维特征经过 tanh 激活层即可得到用户现有的兴趣特征,这个特征就表达了用户的下一个感兴趣的兴趣点;同时,这个高维特征内还应该能表达用户的活动范围,利用全连接层可以提取高维特征中相应的地理位置信息.

### 2.3 损失函数

模型的损失函数分为两部分,预测兴趣点的向量和真实向量之间的差距以及预测地理位置和真实地理位置之间的距离.预测向量和真实向量之间的差距可以刻画预测得到的向量的准确度,是损失函数最重要的部分.而预测地理位置和真实位置之间的差距作为调整的一部分也被加入到损失函数中,这是考虑到用户本身的活动范围有限,预测得到的 POI 不应该距离用户的常规活动范围有太大偏差,不然这样的推荐将变的毫无意义.具体的损失函数如下:

预测得到的向量和真实向量之间的差距采用余弦相似度衡量,预测得到的兴趣点向量为  $\tilde{q}_t$ ,我们采用当前  $t$  时刻之后的用户访问的  $k$  个兴趣点来判定预测准确度:

$$Loss_{sim} = \sum_{i=1}^k \frac{\tilde{q}_t \cdot q_{v_{t+i}}}{|\tilde{q}_t| * |q_{v_{t+i}}|} \quad (1)$$

地理位置之间的相似度采用欧式距离,预测得到的地理位置设置为  $\{\tilde{x}_t, \tilde{y}_t\}$ ,同样测量其和未来  $k$  个兴趣点之间的距离.损失函数如下:

$$Loss_{sim} = \sum_{i=1}^k \sqrt{(\tilde{x}_t - x_{v_{t+i}})^2 + (\tilde{y}_t - y_{v_{t+i}})^2} \quad (2)$$

最终,模型的损失函数如下:

$$Loss_{total} = Loss_{sim} + Loss_{loc} \quad (3)$$

## 3 实验分析

我们使用的数据集为美国最大的点评网站 Yelp 公开的内部数据集以及基于用户地理位置信息的分享网站 Foursquare 公开的数据集.两份数据集都采用数据标注城市为纽约的兴趣点,以及评价过这些兴趣点的用户,数据规模如表 1 所示:

表 1 数据集

Yelp	Foursquare	
用户数	2426	4038
POI 数	5672	13 042
Check-in 数	189 208	342 721

### 3.1 评测标准

由于是预测下一个兴趣点,所以我们的评测标准是 Precision@N,表示预测出 Top N 个作为推荐列表,其中推荐对的占推荐列表中所有的被推荐的兴趣点的比例.

对于每份数据集,我们都按照 1:4 的比例分配测试集和训练集,即每个用户的访问序列的前 80% 数据用来训练,后 20% 数据用来测试.

选取的对比实验有 FPMC<sup>[1]</sup>, Rank-GeoFM<sup>[8]</sup>, POI2Vec<sup>[12]</sup>以及 ST-RNN<sup>[9]</sup>. FPMC 是个性化马尔可夫链模型的代表,该模型为每个用户建立一个转移矩阵,形成转移矩阵立方体,然后使用张量分解模型对转移立方体进行学习,弥补其数据稀疏的问题. Rank-GeoFM 是一种基于时间序列的张量分解模型,将时间分段后获取用户的对兴趣点的多个评分矩阵,然后使用张量分解学习这些评分矩阵. POI2Vec 采用了 Embedding 的思想,将用户访问过的地点序列比拟作为 NLP 中的语句,然后构建兴趣点之间的后验概率. ST-RNN 则利用了循环神经网络的思想,基于兴趣点的维度构建 RNN 预测模型,输出的兴趣点向量,然后参照张量分解的思想和用户向量相乘作为用户访问该兴趣点的概率.选用的四个对比实验涵盖了当前比较主流的 POI 推荐模型种类,并且采用的比较新的模型结构.

### 3.2 实验结果和分析

经过多次实验比较,我们分别测试了 Precision@1、Precision@3、Precision@5.特别注意,由于每个用户访问的序列长度不同,我们会筛选测试序列长度大于等于 N 的部分.所以 N 越大,测试数据集的数量越小.实验结果见图 5 及图 6.

图 5 表示在 Foursquare 上的精确度比较,图 6 表示在 Yelp 上的精确度比较.可以看到 POI-LSTM 在两个数据集上都表现抢眼,这说明向 LSTM 模型中加入用户的社交信息以及评论和地理位置信息对提升兴趣点预测的准确度有明显作用.而模型在 Foursquare 上相比其它模型效果较 Yelp 上效果提升更明显,这是因

为 Foursquare 数据集比 Yelp 的数据集平均序列长度更短, 使用额外的非序列信息对于提高准确度帮助更大, 说明我们的模型在应对冷启动问题上也有很大的效果。

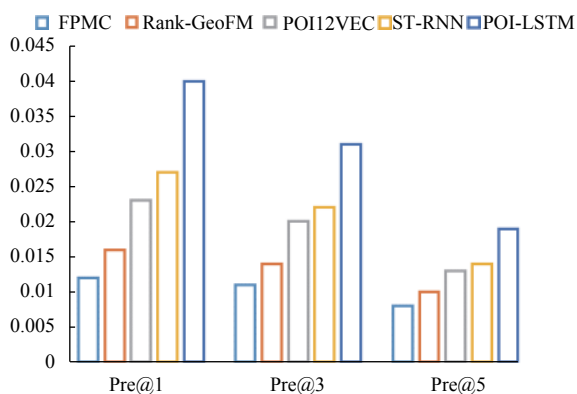


图5 Foursquare 数据集 Pre@N

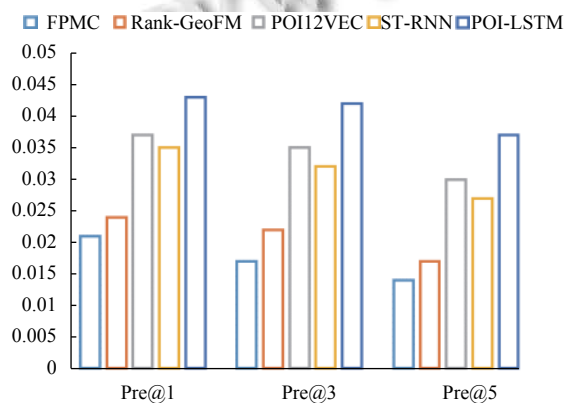


图6 Yelp 数据集 Pre@N

#### 4 结语

本文成功的将 LSTM 和 Embedding 算法结合, 并成功应用在兴趣点推荐场景下, 通过 Embedding 的思想把用户的社交信息和标签、评论信息变成向量输入到循环神经网络中捕捉用户的长期特征和短期爱好. 实验表明, 该方法可以有效的提升兴趣点预测的准确度, 并能部分解决推荐场景中的冷启动问题。

#### 参考文献

- Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized Markov chains for next-basket recommendation. Proceedings of the 19th International Conference on World Wide Web. Raleigh, NC, USA. 2010. 811–820.
- Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer, 2009, 42(8): 30–37. [doi: 10.1109/MC.2009.263]
- Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model. INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association. Makuhari, Chiba, Japan. 2010. 1045–1048.
- Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 1994, 5(2): 157–166. [doi: 10.1109/72.279181]
- Schäfer A M, Udluft S, Zimmermann H G. Learning long term dependencies with recurrent neural networks. Kollias SD, Stafylopatis A, Duch W, et al. Artificial Neural Networks - ICANN 2006. Berlin, Heidelberg: Springer, 2006.
- Koren Y. Collaborative filtering with temporal dynamics. Communications of the ACM, 2010, 53(4): 89–97. [doi: 10.1145/1721654]
- Lathia N, Hailes S, Capra L. Temporal collaborative filtering with adaptive neighbourhoods. Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval. Boston, MA, USA. 2009. 796–797.
- Li XT, Cong G, Li XL, et al. Rank-GeoFM: A ranking based geographical factorization method for point of interest recommendation Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. Santiago, Chile. 2015. 433–442.
- Liu Q, Wu S, Wang L, et al. Predicting the next location: A recurrent model with spatial and temporal contexts. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. Phoenix, AZ, USA. 2016. 194–200.
- Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. arXiv preprint arXiv: 1301.3781, 2013.
- Cheng C, Yang HQ, Lyu MR, et al. Where you like to go next: Successive point-of-interest recommendation. Proceedings of the 23rd International Joint Conference on Artificial Intelligence. Beijing, China. 2013. 2605–2611.
- Cheng C, Yang HQ, King I, et al. Fused matrix factorization with geographical and social influence in location-based social networks. Proceedings of the 26th AAAI Conference on Artificial Intelligence. Toronto, Ontario, Canada. 2012. 17–23.