DeepStore: an interaction-aware Wide&Deep model for store site recommendation with attentional spatial embeddings

Yan Liu, Bin Guo, Nuo Li, Jing Zhang, Jingmin Chen, Daqing Zhang, Yinxiao Liu, Zhiwen Yu, Sizhe Zhang, Lina Yao

DeepStore: An Interaction-Aware Wide&Deep Model for Store Site Recommendation With Attentional Spatial Embeddings

Yan Liu, Bin Guo, *Senior Member, IEEE*, Nuo Li, Jing Zhang, Jingmin Chen, Daqing Zhang, *Fellow, IEEE*, Yinxiao Liu, Zhiwen Yu, *Senior Member, IEEE*, Sizhe Zhang, and Lina Yao, *Member, IEEE*

***Abstract*—Store site recommendation is one of the essential business services in smart cities for brick-and-mortar enterprises. In recent years, the proliferation of multisource data in cities has fostered unprecedented opportunities to the data-driven store site recommendation, which aims at leveraging large-scale user-generated data to analyze and mine users' preferences for identifying the optimal location for a new store. However, most works in store site recommendation pay more attention to a single data source which lacks some significant data (e.g., consumption data and user profile data). In this paper, we aim to study the store site recommendation in a fine-grained manner. Specifically, we predict the consumption level of different users at the store based on multisource data, which can not only help the store placement but also benefit analyzing customer behavior in the store at different time periods. To solve this problem, we design a novel model based on the deep neural network, named DeepStore, which learns low- and high-order feature interactions explicitly and implicitly from dense and sparse features simultaneously. In particular, DeepStore incorporates three modules: 1) the cross network; 2) the deep network; and 3) the linear component. In addition, to learn the latent feature representation from multisource data, we propose two embedding methods for different types of data: 1) the filed embedding and 2) attention-based spatial embedding. Extensive experiments are conducted on a real-world dataset including store data, user data, and point-of-interest data, the results demonstrate that DeepStore outperforms the state-of-the-art models.**

***Index Terms*—Attention mechanism, data analytics, deep learning, spatial embedding, store site recommendation.**

Manuscript received March 14, 2019; revised May 3, 2019; accepted May 7, 2019. Date of publication May 10, 2019; date of current version July 31, 2019. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1001800, in part by the National Natural Science Foundation of China under Grant 61772428 and Grant 61725205, and in part by the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University under Grant CX201958. *(Corresponding author: Bin Guo.)*

Y. Liu, B. Guo, N. Li, J. Zhang, and Z. Yu are with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China (e-mail: liuyan.emily@mail.nwpu.edu.cn; guob@nwpu.edu.cn; 2326375288@qq.com; 1107141724@qq.com; zhiwenyu@nwpu.edu.cn).

J. Chen, Y. Liu, and S. Zhang are with the Alibaba Group, Hangzhou 311121, China (e-mail: jingmin.cjm@alibaba-inc.com; yinxiao.lyx@alibaba-inc.com; jincheng.zsz@alibaba-inc.com).

D. Zhang is with the Département Réseaux et Services Multimédia Mobiles, Institut Mines-Télécom/Télécom SudParis, Évry 91011, France (e-mail: daqing.zhang@telecom-sudparis.eu).

L. Yao is with the School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia (e-mail: lina.yao@unsw.edu.au).

Digital Object Identifier 10.1109/JIOT.2019.2916143

I. Introduction

In RECENT years, with the rapid development of Internet technology and mobile devices, the amount of user-generated data from various sources in cities has grown explosively, such as social media, electronic commerce Websites, mobile devices, sensor networks, etc. Urban computing exploits those big data generated by a diversity of sources in smart cities [1], [2] to tackle the major issues that cities face and provide the high-quality and beneficial services for the citizens [3], such as the fields of traffic [4], business [5], and so on. Store site recommendation is one of the essential business services in smart cities for brick-and-mortar enterprises (e.g., the retail store). Choosing a good location when opening a new store is crucial for the future success of a business, since 94% of retail sales are still transacted in physical stores [6]. Therefore, an effective site recommendation system becomes necessary to brick-and-mortar enterprise managers.

The recent proliferation of multisource data in cities has fostered unprecedented opportunities to the data-driven store site recommendation, which aims at leveraging large-scale user-generated data (e.g., check-in data and rating data) to analyze and mine users' preferences for identifying the optimal location for a new store. Compared to traditional methods which generally conduct surveys to assess the value of store locations which is time-consuming and do not scale up well, the data-driven method can mine user behavior and extract knowledge from geographic datasets via data mining and machine learning techniques. For example, a lot of works [7]–[11] study the optimal location problem from location-based social networks, and most of them learn the regression model based on extracted features to predict the check-in numbers at given locations.

However, there still are some limitations in present data-driven methods for store site recommendation. First, most works just analyze the popularity of the store location based on a single data source (e.g., check-in data), which lacks other significant data (e.g., real-word consumption data) affecting the store placement. Second, they rely on expertise feature engineering to characterize sophisticated influences, but it comes with a high cost to obtain high-quality features and hardly generalizes to other situations. Third, most models fail to learn complex feature interactions from multisource data in complicated problems. Therefore, in this paper, we aim to tackle

above-mentioned issues, and we study the problem of store site recommendation in a fine-grained manner.

Generally, the objective of the store site recommendation problem is to select the optimal location which can maximize sales of the store. Intuitively, different users play an important role in store site recommendation, since the overall sales of the store are decided by the consumption of each user. Therefore, to understand consumption behaviors of potential consumers at the store and help the selection of store location from the fine-grained perspective, we aim to predict the consumption level of different users at the store based on multisource data (e.g., store data and user profiling data). Furthermore, we can infer whether the candidate location is appropriate to open the store in the long term in view of nearby customers.

The key challenge to solve this problem is modeling customer behaviors from multisource data, including store data, user data, and point-of-interest (POI) data. On the one hand, we need to extract valuable features from multiple types of data, such as user profile data and geographical data. On the other hand, we need to learn feature interactions among multiple features, since customer behavior is usually affected by various factors simultaneously.

Recently, deep neural networks (DNNs) [12] have experienced great success in many fields because of their great power of feature representation learning, such as computer vision [13] and natural language processing [14]. More importantly, DNNs have the ability to learn sophisticated feature interactions from raw data. More and more works have been trying to leverage DNNs to learn latent representations from raw data. For example, Lian *et al.* [15] proposed a deep fusion model (DFM) including an inception module and an attention mechanism for feature-aware representation learning. Zhang *et al.* [16] studied feature representations and proposed factorization-machine supported neural network (FNN). However, these works mainly learn the feature representation from categorical data, which are not suitable to learn features from geographical data in our problem. In order to learn latent feature representation from multiple types of data in this paper, we propose two embedding methods: the field embedding to reduce the dimensionality of sparse features (e.g., categorical data), the attention-based spatial embedding to learn the importance of spatial features.

A lot of works have been studying feature interactions without manual engineering. Factorization machines (FM) is the typical framework which models the feature interactions as the inner product of latent vectors between different features. Although FM can achieve higher-order feature interactions theoretically, most FM methods only order two feature interactions due to high complexity. In view of the learning ability of neural networks, DNNs are used to model the high-order feature interactions implicitly [17]. To model both low- and high-order feature interactions, Cheng *et al.* [18] proposed a typical hybrid network structure (Wide&Deep) that contains a linear ("wide") model and a deep model, and it combines the benefits of memorization and generalization for recommender systems. However, the input of *wide part* still relies on expertise feature engineering. Further, some models are proposed to reduce the manual feature engineering, such as DeepFM [19], DCN [20], and xDeepFM [21]. For example, DCN contains a cross network (CrossNet) which can efficiently capture feature interactions of bounded degrees. xDeepFM not only can learn explicit and implicit high-order feature interactions, but also can generate feature interactions at the vector-wise level. However, the major downside of these models is that they fail to learn latent feature representation and feature interactions from multisource data. For instance, they do not consider the spatial distribution of POI data, and the effective interaction between dense features and sparse features.

In this paper, we design a unified interaction-aware model, named DeepStore, which can process different types of data from multiple data sources for store site recommendation. Our model is based on the Wide&Deep framework, which aims to efficiently capture low- and high-order feature interactions simultaneously. In particular, we design a novel CrossNet to learn high-order feature interactions explicitly from dense and sparse features. In addition, following the spirit of the DCN and xDeepFM models, we combine the explicit high-order interaction module (the CrossNet) with implicit interaction module (the deep network) and traditional linear module.

In summary, we make the following contributions.

1) We formulate a problem for store site recommendation, which aims to predict the consumption level of different users at the store based on multisource data.
2) We propose a unified interaction-aware model based on the Wide&Deep network, named DeepStore, including the CrossNet module, the deep network module, and linear module. Unlike general Wide&Deep models [18], [19], it can jointly learn low- and high-order feature interactions explicitly and implicitly from multimodal features.
3) We design a new CrossNet to fuse multimodal feature interactions explicitly. Specifically, it can model hybrid feature interactions, which makes sparse features interact at the vector level and dense features interact at the bite level.
4) We propose two embedding methods in DeepStore, the filed embedding and attention-based spatial embedding, to learn latent feature representation from multimodal data. In particular, two-level attention is designed in the attention-based spatial embedding to learn the importance of different spatial features in view of user profile and distance simultaneously.
5) We evaluate our proposed model based on a real-world dataset, including store data, user data, and POI data. Extensive experiments are conducted from different perspectives, the results demonstrate that our DeepStore outperforms several state-of-the-art models.

The remainder of this paper is organized as follows. In Section II, we review the relevant work. Section III presents an overview of our proposed framework to solve the store site problem. In Section IV, we elaborate the proposed model, named DeepStore. The experiments based on real-world dataset are conducted in Section V. Finally, we conclude this paper in Section VI.

II. Related Work

In this section, we review the related work, including location recommendation, deep learning for feature interactions, and attention mechanism.

A. Location Recommendation

In recent years, location recommendation has been a trending research area, since the rapid increase of the availability of big data provides researchers with the possibility to access users' location, such as check-in data. *POI recommendation* and *optimal site recommendation* are two main research problems in location recommendation.

POI recommendation [22], [23] has become an important way to help people discover attractive and interesting places, such as restaurants, hotels, and so on. Yin *et al.* [24] proposed a probabilistic generative model TRM for joint modeling of users' check-in behaviors by exploiting the semantic, temporal, and spatial patterns in a unified way. Liu *et al.* [25] proposed an recurrent neural network (RNN)-based neural network solution by modeling the user's historical POI visits in a sequential manner. Feng *et al.* [26] proposed a POI latent representation model, named POI2Vec, which incorporates the geographical influence of POIs to predict the potential visitors for a location in the next few hours. Qian *et al.* [27] proposed a spatiotemporal context-aware and translation-based recommender framework (STA) to model the third-order relationship among users, POIs, and spatiotemporal contexts for large-scale POI recommendation. Yin *et al.* [28] proposed a spatial-aware hierarchical collaborative deep learning (SH-CDL) model, which models jointly performs deep representation learning for POIs from heterogeneous features and hierarchically additive representation learning for spatial-aware personal preferences.

Different from POI recommendation which recommends a place to people, site recommendation provides the optimal location for the enterprise to open the store. Early studies are based on dedicated models for store site recommendation. Sevtsuk [29] analyzed location patterns of retail and food establishments. It tests five hypotheses about retail locations found in previous literature using an economic model, and estimates the impacts of different location characteristics for store placement. Li and Liu [30] presented a modified Huff model to estimate the potential sales of individual Kmart and Walmart stores. Roig-Tierno *et al.* [31] presented a methodology for retail site location decision, which takes both geographic information systems and the analytical hierarchy process into consideration. However, those dedicated models mainly rely on domain expert knowledge or traditional data, which hardly learn related knowledge effectively under complex factors.

Recently, with the emergence of large-scale urban data, there is a potential to leverage these data to analyze and mine users' preferences for store site recommendation. For example, Karamshuk *et al.* [8] demonstrated the power of geographic and user mobility features in predicting the best placement of retail stores based on check-in data. In [32], three types of features are incorporated into a regression model to predict the number of check-ins at a candidate location, including review-based market attractiveness features, review-based market competitiveness features, and geographic features. Lin *et al.* [33] analyzed the popularity of a business location using Facebook data, and proposed a model based on gradient boosting machine to estimate the popularity of a given target location. Xu *et al.* [34] proposed a demand distribution driven store placement (D3SP) framework for store location selection via mining search query logs of Baidu Maps. Guo *et al.* [11] proposed a twofold knowledge transfer framework based on collaborative filtering to solve the cold-start problem, which can transfer chain store knowledge from semantically relevant domains.

This paper differs from previous works in the following aspects: first, the problem definition is different, we study the problem of store site recommendation in a fine-grained manner. Specifically, we aim to predict the consumption level of different users in the store, which not only can help the store placement, but also benefits to study customer behavior in the store at different time periods; second, the dataset is different, we obtain real-world and fine-grained data to study customer behavior, including store data, user data, and POI data. Third, the model for store site recommendation is different, we propose a unified interaction-aware model based on the neural network which can learn complex relations from multisource data.

In this paper, we study customer behaviors in retail enterprises for store site recommendation. Different from traditional consumption behaviors in bricks-and-mortar stores, the retail enterprise in our research is a new business form, which combines online and offline business. That is to say, users could consume online and offline in the store simultaneously. The key challenge to solve this problem is modeling customer behaviors from multisource data, including store data, user data, and POI data. On the one hand, we need to extract valuable features and model feature interactions from multimodal data. On the other hand, we need to model the influence of POIs with the spatial distribution. Therefore, in the following, we review some techniques to solve above-mentioned two issues, including deep learning for feature interaction and attention mechanism.

B. Deep Learning for Feature Interactions

Deep learning techniques have achieved great success in computer vision, speech recognition, and natural language understanding. As a result, an increasing number of researchers are interested in employing DNNs to extract representative features from raw data and model deep interaction of features [35], [36].

Feature interaction based on deep learning has been widely studied in recommender system, especially in click-through rate (CTR) prediction. CTR prediction plays an important role in recommender systems, which aims at predicting the probability a user will click on a recommended item in view of complex factors. It is important for CTR prediction to model complex feature interactions behind user click behaviors, thus learning to extract features without manual engineering is necessary. To model both low- and high-order feature interactions, Cheng *et al.* [18] proposed a typical

hybrid network structure (Wide&Deep) that combines a linear (wide) model and a deep model. In this model, two different inputs are required for the "wide part" and "deep part," respectively. However, the input of wide part still relies on expertise feature engineering. In order to reduce manual feature engineering, Guo *et al.* [19] proposed a new neural network model DeepFM, which integrates the architectures of FM and DNNs. It can model low-order feature interactions like FM and models high-order feature interactions like DNN. However, Wide&Deep and DeepFM just model high-order feature interactions implicitly, since the function learned by DNNs can be arbitrary. Furthermore, Wang *et al.* [20] proposed the deep and cross network (DCN) model, which can learn high-order feature interactions implicitly and explicitly. Particularly, DCN contains a CrossNet that can capture feature interactions of bounded degrees. However, DCN models feature interactions at the bit-wise level, which is different from the traditional FM framework which models feature interactions at the vector-wise level. The details about bit-wise and vector-wise feature interaction will be introduced in Section IV. Recently, Lian *et al.* [21] proposed a new model, named xDeepFM, which can learn explicit and implicit high-order feature interactions effectively. In particular, they designed a compressed interaction network (CIN), which generates feature interactions in an explicit fashion and at the vector-wise level.

The above-mentioned deep models, however, are mainly designed to learn representation and feature interactions from the data of a single modality. With the proliferation of multi-source data in cities, many efforts have been made to fuse multimodal data and learn interactions among multimodal features in the deep network. Nie *et al.* [37] presented the multisource mono-task learning model and its application in volunteerism tendency prediction based on aggregation of multiple social networks. In particular, they introduced multisource dataset construction and how to effectively and efficiently complete the item-wise and block-wise missing data. Yin *et al.* [28] presented a late feature fusion strategy into the SH-CDL model to deal with the multimodal heterogeneous features of the POIs. Du *et al.* [38] proposed a priority-based fusion method, which use exponential weights to model the overwhelming influences from the stronger content features. Wu and Han [39] proposed a new module of multimodal circulant fusion to fully exploit interactions among multimodal features. In particular, they defined two types of interaction operations between original feature vectors and the reshaped circulant matrices.

C. Attention Mechanism

The attention mechanism has been successfully adopted in various machine learning tasks, which could improve the performance of the model. The main reason is that it can select valuable parts of the whole feature space. Xiao *et al.* [40] proposed attentional FM, which learns the importance of each feature interaction from data via a neural attention network. Chen *et al.* [41] introduced the attention mechanism in CF to address the challenging item- and component-level implicit feedback in the multimedia recommendation. They proposed an attention model including two attention components: 1) the component-level attention module and 2) the item-level attention module.

Visual attention is an effective method in computer vision [42], [43], which can learn representative spatial features from geographical data. Xu *et al.* [44] proposed the first visual attention model in image captioning, which automatically learns to describe the content of images. Xiao *et al.* [45] applied visual attention to fine-grained classification task using the DNN. Chen *et al.* [46] proposed a convolutional neural network, named SCA-CNN, which combines spatial and channel-wise attentions in a CNN. Chen *et al.* [47] proposed an attention-based configurable convolutional neural network (ABC-CNN) for visual question answering task, to locate the question-guided attention based on input queries.

In this paper, we will take full advantage of the deep models for feature interactions [20], [21]. In addition, inspired by the attention model in vision attention [46], [47], we also leverage the attention mechanism to learn better latent representations from a large number of geospatial data. In general, our model distinguishes the models in above-mentioned works in the following aspects. First, we design a unified interaction-aware model based on the Wide&Deep framework, which can process different types of data from multiple data sources. Second, we apply two embedding methods to learn dense and valuable features from categorical data and geographical data, respectively. Third, we propose two-level attention in the attention-based spatial embedding to learn the importance of different spatial features. Finally, we jointly learn low- and high-order feature interactions explicitly and implicitly from dense and sparse features simultaneously.

III. Overview

In this section, we first present the problem formulation, and describe the dataset we used, then analyze the complex factors that may influence the selection of store location. Finally, we illustrate the system framework of this paper.

A. Problem Formulation

1) Store Site Recommendation Problem: Given some candidate locations to open a new store, the objective of the store site recommendation problem is to select the optimal location which can maximize sales of the store. On the one hand, from the coarse-grained perspective, we need to predict the overall sales of the store at the candidate location. On the other hand, from the fine-grained perspective, the consumption of different customers should be predicted to assist store site selection. Intuitively, different users play an important role in store site recommendation, since the overall sales of the store are decided by the consumption of each user. Therefore, to understand the consumption behaviors of potential consumers at the store and guide the selection of store location from the fine-grained perspective, we aim to predict the consumption situation of different users at the store. Further, we can infer whether the candidate location is appropriate to open the store in the long term in view of nearby customers.

TABLE I
STATISTICS OF THE REAL-WORD COMMERCIAL DATASET

Data	Statistic information		
Store data	*The number of stores*	*The number of cities*	*Observation period*
	49	13	15.01.*2016* - 15.09.2018
User data	*The number of users*	*The number of communities*	*The number of user profile attributes*
	37,159,703	33,251	11
POI data	*The* number *of POI categories*	The *set of POI category*	
	10	Shopping, food, transport, company, education, sport, service, medical, hotel, scene	

2) User Consumption Behavior Prediction Problem: In general, the consumption behavior of a single user has a certain degree of randomness to some extent, which is affected by sophisticated factors in real situations. Therefore, we aim to predict the overall consumption of a group of people instead of predicting a single user's consumption. Considering that geographical factor is one of the main impacts of consumer behavior in bricks-and-mortar stores, thus, we consider a group of people as the *location-based community*, which shares a sense of place that is situated in a given geographical area (e.g., a neighborhood). More specifically, *a community in our dataset means a housing estate, which is a group of homes and other buildings built together*.

Assume that there are m nearby communities around the candidate location L_q of the store, denoted as $C_q = \{C_q^1, C_q^2, \ldots, C_q^i, \ldots, C_q^w\}$. In each community, there are some users that have the possibility to consume in the store, denoted as $C_q^i = \{u_1^i, u_2^i, \ldots, u_j^i, \ldots, u_n^i\}$. The user consumption behavior prediction problem aims to predict the users' consumption of each community in the store at the candidate location L. In view of users' privacy, we just predict the level of users' consumption rather than the amount of users' consumption. Specifically, users' consumption is divided into s levels based on equidistance partition, denoted as $\{y_1, y_2, \ldots, y_k, \ldots, y_s\}$. Therefore, user consumption behavior prediction problem can be defined as the classification problem, which predicts the level of community users' consumption in the store at the candidate location L during a given period of time (e.g., a month).

B. Dataset Description and Factors Analysis

In this paper, we choose a retail enterprise for a case study, which is a chain retail enterprise owning multiple bricks-and-mortar stores in some cities across China. Different from traditional bricks-and-mortar stores, this retail enterprise is a new business form which combines online and offline business, known as the new retail [48]. Therefore, the consumption behaviors of users in these stores distinguish traditional consumption behavior. Fortunately, the combination of online business and offline business provides unique opportunities to analyze relationships between customer behaviors and multisource data, such as online sales data and geospatial data. However, it also brings some new challenges, for example, how to analyze the complex factors that may influence the consumption of users in view of multisource data, how to quantify the influence of multiple types of data on customer behavior, etc.

1) Dataset Description: The dataset in this paper is the real-world commercial data, which is unique among existing works. Specifically, it includes three types of data for this paper, including store data (e.g., location, sale information, and customer visiting information), user profile data (e.g., age and gender), and POI data. Store data and user data sourced from the retail company, and POI data obtained from a Map platform. Table I gives a summary of the statistics of multisource dataset we used in this paper.

a) Store data: We mainly use two types of data in store dataset, including the profile information and historical sales data of the store. For the profile information, it contains the shop name, city, location (e.g., longitude and latitude), and opening time (e.g., year and month). For the historical sales data, each record contains customer id, shop name, and customer behavior (e.g., the level and time of consumption). Note that, in view of user privacy, we just obtain the overall consumption level of users in a community during a given period of time as the ground truth to evaluate the prediction result. For example, the consumption level of users in the community C^i in the store S_j is $y_k^{i,j}$ in February.

b) User data: Benefiting from the online business, we obtain user data that can establish the relationship with store sales data, which is unique comparing to existing store site recommendation works. Specifically, user data includes user location information and profile information. Similarly, to protect user privacy, we make statistics on user information in the community. For the location information, we obtain the number of people in a community and the location of the community. For the user profile, we have the following information: the number of men or women in the community, the number of people in different age groups, etc.

c) POI data: We rely on POI dataset to characterize the geographical environment of different places and further analyze the impact of the surrounding environment of bricks-and-mortar stores on customer behavior. In this paper, POI data contains the information (e.g., name, location, and category) of multiple categories of POI related to the retail business, such as shop, food, transport facilities, and so on.

2) Multifactor Analysis: There are various complex factors that may influence the consumption behavior of users in the store. Considering that the number of existing stores is small, in order to analyze customer behavior in detail, we treat each

nearby community around the store as an instance instead of a store. Therefore, we measure the consumption level of users in each community in the store, which is used as the ground truth for the prediction model. In general, we take a community as an entity and analyze different factors which may influence customer behavior in the store based on three datasets. Specifically, these factors are classified into three categories, including user factors, geographic factors, and time factors.

a) User factors: Whether the user consumes in the store depends, to a great extent, on the user profile. For example, there is a higher possibility for women to consume in the clothing shop compared to men, families with babies would like to consume in the baby store. In addition, the amount of consumption of the user in the store is always associated with the user's income level. Therefore, we make statistics on the number of people with different profiles in each community to characterize users in different communities. Specifically, user factors contain the number of men or women in the community, the number of people in different age groups, the number of people with different professions, the number of people on different income levels, etc.

b) Geographic factors: The geographic factors [8], [28] we introduced assess spatial characteristics around the place where the store resides and the nearby communities. More specifically, we measure the following geographic factors of surrounding areas, which lie in a disk of radius r around the store and the community.

Distance: We consider the distance between the store located at l_i and the community located at l_j, as formulated in (1). The farther away from the distance, the less likely customers go to the store

$$x_{ij}^{\text{distance}} = \text{Dist}(l_i, l_j). \tag{1}$$

Traffic convenience: We use the number of transportation stations of multiple categories Γ_t (including bus stations and subway stations) in the surrounding area to denote the traffic convenience of the place at l_j, which is defined as (2). Here, $N^{c_i}(l_j, r)$ is the number of transportation stations of category $c_i \in \Gamma_t$ in the surrounding area, which is a disc centered at l_j with radius r

$$x_j^{\text{transport}} = \sum_{c_i \in \Gamma_t} N^{c_i}(l_j, r). \tag{2}$$

Density: We calculate the total number of stores of different categories in surrounding areas to assess what extent the popularity of the place at l_j, as define in (3). Intuitively, a denser area could have a higher likelihood of attracting more users

$$x_j^{\text{density}} = N(l_j, r). \tag{3}$$

Neighbors entropy: We apply the entropy measure from information theory to measure the spatial heterogeneity and diversity of a place, which is defined as (4). We denote the number of place neighbors of type c_i with $N^{c_i}(l_j, r)$, where $c_i \in \Gamma$ is one of the categories in POI data

$$x_j^{\text{diversity}} = -\sum_{c_i \in \Gamma} \frac{N^{c_i}(l_j, r)}{N(l_j, r)} \times \log \frac{N^{c_i}(l_j, r)}{N(l_j, r)}. \tag{4}$$

Competitiveness: We consider the competitive relationship between the stores belonging to the same category, which is measured as the proportion of neighboring places of the same type c with respect to the total number of nearby places, as formulated in the following equation:

$$x_j^{\text{competitiveness}} = -\frac{N^c(l_j, r)}{N(l_j, r)}. \tag{5}$$

Complementarity: We consider the complementarity relationship of spatial interactions between different categories in the same area. We employ, Jensen quality, defined by Jensen [7], to assess the complementarity relationship of spatial interactions of places with respect to their ability to attract other places of certain types. Specifically, it first uses a utility intertype coefficient to quantify the dependency between different POI category pairs in the city. Then, the overall complementarity of the given place is computed based on the POI category distribution of the region, which is defined as

$$\rho_{\gamma_p \to \gamma_l} = \frac{N - N_{\gamma_p}}{N_{\gamma_p} \times N_{\gamma_l}} \sum_p \frac{N_{\gamma_l}(p, r)}{N(p, r) - N_{\gamma_p}(p, r)} \tag{6}$$

$$x_j^{\text{com}} = \sum_{\gamma_p \in \Gamma} \log\left(\rho_{\gamma_p \to \gamma_l}\right) \times \left(N_{\gamma_p}(l, r) - \overline{N_{\gamma_p}(l, r)}\right) \tag{7}$$

where γ_p is the type of the place p, $N_{\gamma_l}(p, r)$ is the number of places of type γ_l in the surrounding area, which is a disc centered at p with radius r, and $\overline{N_{\gamma_p}(l, r)}$ denotes how many places of type γ_p are observed on average around the places of type γ_l.

POI set: We also consider the number of POIs of related categories in the surrounding area of the place located in l_j, which is defined as (8). In order to reflect the distribution of POIs, we divide the surrounding area into a set of location grids, and calculate the number of POIs in different location grids, where $\mathbf{M}^{c_1}(l_j, r)$ is a matrix to represent the number of POIs of category c_1 in different location grids

$$\mathbf{X}_j^{\text{POI}} = \left\{\mathbf{M}^{c_1}(l_j, r), \mathbf{M}^{c_2}(l_j, r), \ldots, \mathbf{M}^{c_n}(l_j, r)\right\}. \tag{8}$$

c) Time factors: The time factors we explore attempt to capture customer behavior during different time periods. Brand awareness is one of the key factors that influence customer behavior in the store, and it constantly changes at different stages. Intuitively, we use the opening time of the store (e.g., the year) and the number of existing stores to evaluate the popularity of the brand in users. In addition, most of the user behaviors have the obvious seasonal characteristic, so we regard the month when users consume in the store as a factor to reject the consumption habits. Moreover, the holiday is also an important factor which could stimulate consumption, such as New Year's Day and National Day. In addition to the time when users consume in the store, the opening time of the store.

C. DeepStore Framework

The framework of DeepStore is illustrated in Fig. 1, which mainly consists of five components: 1) input features; 2) embedding layer; 3) CrossNet; 4) deep network; and 5) combination layer.

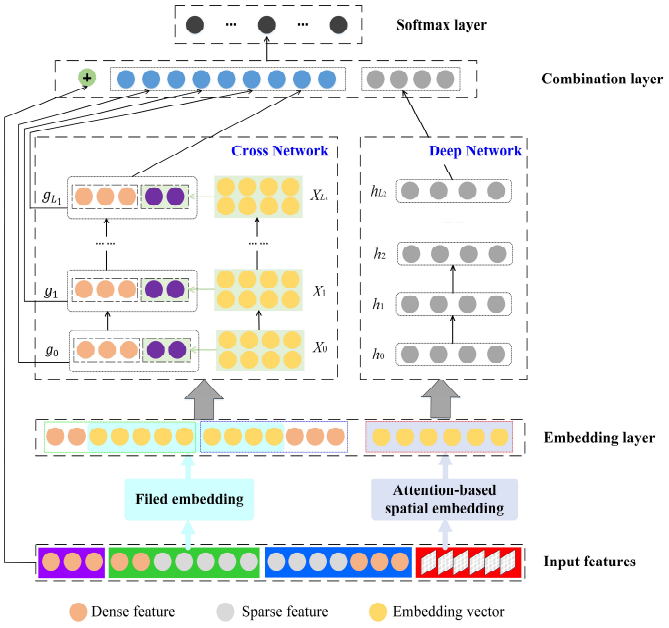


Fig. 1. DeepStore framework.

1) Input Features: Different from the computer vision or natural language understanding that the input data (e.g., images or texts) can be applied directly to DNNs as the raw features, the multisource data in this paper is multidimensional and multicategorical. Therefore, we extract some useful features from multisource data based on the above factor analysis as the input features of DeepStore. Specifically, input features consist of four parts.

1) *Expert features* in purple box extracted from POI data, such as competitiveness and complementarity.
2) *Store-related features* in green box extracted from the store data, such as the opening time of the store.
3) *User-related features* in blue box extracted from the user data, such as the number of people in different age groups.
4) *Spatial features* in red box extracted from the POI data, such as the distribution of POIs of related categories.

2) Embedding Layer: The input features include sparse features and dense features. Considering that most sparse features are high-dimensional, so we employ an embedding layer to learn a low-dimensional, dense real-value feature representation of sparse features. For the categorical feature which is always transformed into a high-dimensional sparse feature via one-hot encoding, we use the *field embedding* to reduce the dimensionality. For spatial features, we employ the *attention-based spatial embedding* to transform them into dense vectors.

3) Cross Network: It exploits a feed-forward neural network to explicitly model the high-order feature interactions based on embedding features of sparse features and dense features, without any other feature engineering besides raw features.

4) Deep Network: It is a fully connected neural network, which can learn implicit high-order interactions. It is noted that cross component and deep component share the same feature embedding.

5) Combination Layer: To model both low- and high-order feature interactions, we apply the combination layer to aggregate the linear competent, the CrossNet and the deep network to make the model stronger. Finally, based on the learned model, we can predict the level of user consumption in the store.

TABLE II
SUMMARY OF THE DEFINITION OF NOTATIONS

Notation	Description
$\mathbf{x}_c = [\mathbf{x}_{c,1} ... \mathbf{x}_{c,m}]$	The input vector of categorical features
$\boldsymbol{\mathcal{X}}_s = [\mathbf{X}_{s,1} ... \mathbf{X}_{s,c}]$	The input tensor of spatial features
$\mathbf{x}_{fm} = [\mathbf{x}_{fm,1} ... \mathbf{x}_{fm,m}]$	The output vector of the field embedding
$\mathbf{x}_{sm} = [\mathbf{x}_{sm,1} ... \mathbf{x}_{sm,c}]$	The output vector of the spatial embedding
$\mathbf{g}_l = [\mathbf{b}_l, \mathbf{a}_l]$	The output of the l-th layer in cross network
$\boldsymbol{h}_l$	The output of the l-th layer in deep network
$\mathbf{W}_{fm,i}$	The embedding matrix of the field embedding
$\mathbf{W}_c, \mathbf{b}_c, \mathbf{W}_s, \mathbf{b}_s, \mathbf{W}_{sm,i}$	The parameters of the spatial embedding
$\mathbf{W}_{l-1}^c, \mathbf{w}_l^c, \mathbf{b}_l^c$	The parameters of cross network
$\mathbf{W}_l^d, \mathbf{b}_l^d$	The parameters of deep network

IV. Proposed Model: DeepStore

In this section, we elaborate the proposed model for predicting the level of user consumption in the store. The DeepStore model starts with an embedding layer which is fed with the set of raw features extracted from multisource data, followed by a CrossNet and a deep network in parallel. Finally, a combination layer is employed to combine the outputs from the linear component, the cross component, and the deep component. The complete DeepStore model is depicted in Fig. 1. A summary of the definition of the main notations used in this paper is given in Table II.

A. Embedding Layer

The input features of our model include sparse features and dense feature, which are extracted from multisource data. Specifically, dense features refer to the vectors of real values, such as the distance, the number POIs, the number of people, etc. Sparse features in this paper contain two types. The first one is the categorical feature (e.g., city and month), which are often encoded as one-hot vectors. The second one is the spatial feature, such as the POI set in different location grids. Generally, sparse features are high-dimensional compared to dense features. Therefore, we apply an embedding layer upon sparse features to learn a low-dimensional and dense feature vectors. More specifically, we use two embedding methods in view of different characteristics of sparse features of two types. For the categorical feature, we use the *field embedding* to reduce the dimensionality. For spatial features, we employ the *attention-based spatial embedding* to transform them into dense vectors.

1) Filed Embedding: Recently, many works have used neural networks to learn advanced representation to replace sparse one-hot vectors. Following widely used methods in

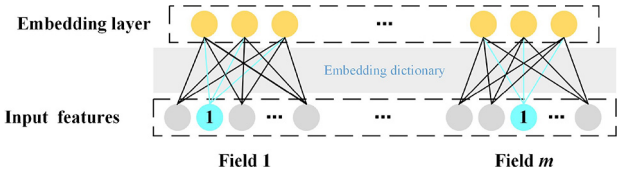


Fig. 2. Filed embedding.

DNNs [20], [21], we adopt field embedding to learn a low-dimensional vectors.

Let$m$denote the number of categorical feature fields, then the categorical features are encoded as high-dimensional sparse features via field-aware one-hot encoding, denoted as $\mathbf{x}_c = [\mathbf{x}_{c,1}, \mathbf{x}_{c,2}, \ldots, \mathbf{x}_{c,i}, \ldots, \mathbf{x}_{c,m}]$, where $\mathbf{x}_{c,i}$ is the binary vector in the ith category. To reduce the dimensionality, we employ the field embedding to transform these binary features into dense vectors of real values. Let $\mathbf{x}_{fm} = [\mathbf{x}_{fm,1}, \mathbf{x}_{fm,2}, \ldots, \mathbf{x}_{fm,i}, \ldots, \mathbf{x}_{fm,m}]$ denote the output of the field embedding, and each field is mapped to a D dimension vector via a dictionary look-up operation

$$\mathbf{x}_{fm,i} = \mathbf{W}_{fm,i}\mathbf{x}_{c,i} \tag{9}$$

where $\mathbf{x}_{fm,i} \in R^D$ denotes the embedding of the ith field, and $\mathbf{W}_{fm,i}$ is the corresponding embedding matrix that will be optimized together with other parameters in the network. The field embedding is illustrated in Fig. 2.

2) Attention-Based Spatial Embedding: Different from the categorical features, spatial features (e.g., the POI set in different location grids) have the spatial proximity relationship. In addition, spatial features of different girds and multiple categories have different influences on customers. Therefore, Inspired by studies on the usage of attention mechanisms in computer vision [46], [47], we propose the attention-based spatial embedding to learn the dense vectors of spatial features, as shown in Fig. 3.

a) Input spatial features: As mentioned in Section III-B2, in order to represent the distribution of geospatial locations of different POIs around the given place, we divide the surrounding area of the place into a set of location grids, and each of them has a size of $k \times k$ (e.g., $k = 500$ m). Therefore, the number of POIs in different location grids can be regarded as the image pixel data. For example, an image $w \times h$ pixels means that there are $w \times h$ grids in the surrounding area of the place. It is noted that the center of the area is the place, so w and h should be both even numbers. In addition, to characterize the influence of POIs of different categories, we consider C related categories in our POI data. Similarly, each category is considered as a channel of the image. In general, let $\mathbf{X}_s \in R^{W \times H \times C}$ denote input tensor of spatial features, where C is the number of POI category, and w and h is the number of grids along the weight and height of the area, respectively.

Generally, spatial features have a different impact on customer behavior in different situations. For example, POIs closed to users may have an obvious influence on them compared to other remote POIs, and POIs of different category could attract different types of users. Therefore, we apply

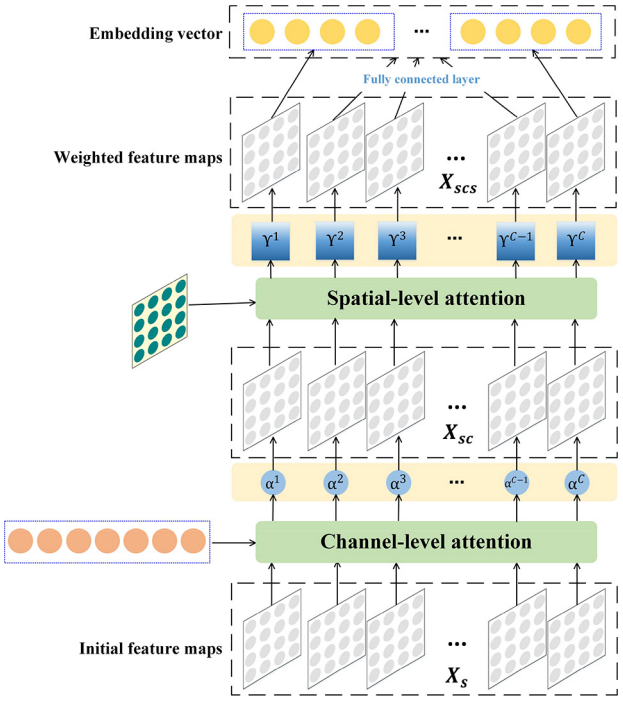


Fig. 3. Attention-based spatial embedding.

the attention mechanism in spatial embedding. Specifically, rather than representing the input raw feature into a static vector, the attention mechanism is able to learn the feature to evolve from the context in different situations, resulting in richer and longer representation for raw features. In general, we incorporate spatial-level attention and channel-level attention to learning the importance of spatial features in spatial embedding.

b) Channel-level attention: Different users may be attracted by POIs of different categories. Hence, we apply the channel-level attention to pay more attention to the useful POIs in view of different users. Given the input feature map $\mathbf{X}_s \in R^{W \times H \times C}$, we first use $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_i, \ldots, \mathbf{v}_C]$ to represent the original feature $\mathbf{X}_s$, where $\mathbf{v}_i \in R^{W \times H}$ is the ith channel of the feature map. Note that we only focus on the influence on different channels in view of different user features $\mathbf{f}_u$, therefore, we ignore the spatial distribution of the feature map. Specifically, we apply mean pooling to each channel to obtain the channel feature v, where scalar v_i is the mean of $\mathbf{v}_i$, which represents the ith channel features:

$$\mathbf{v} = [v_1, v_2, \ldots, v_i, \ldots, v_c]\mathbf{v} \in R^C. \tag{10}$$

The channel-level attention first uses a single-layer neural network to obtain the attention scores, then followed by a softmax function to generate the attention distributions $\boldsymbol{\alpha}$ over the grids. Formally, the channel-level attention network is defined as

$$\mathbf{A} = \sigma((\mathbf{W}_{c1} \otimes \mathbf{v} + \mathbf{b}_{c1}) \oplus \mathbf{W}_{cu}\mathbf{f}_u) \tag{11}$$

$$\mathbf{W}_{c1} \in R^k, \mathbf{v} \in R^C, \mathbf{b}_{c1} \in R^k, \mathbf{W}_{cu} \in R^{k \times p}, \mathbf{f}_u \in R^p$$

$$\boldsymbol{\alpha} = \text{soft max}(\mathbf{W}_{c2}\mathbf{A} + b_{c2})$$

$$\mathbf{W}_{c2} \in R^k, \mathbf{A} \in R^{k \times C}, b_{c2} \in R^1. \tag{12}$$

c) Spatial-level attention: In general, POIs closed to the given place (e.g., the store and community) may contribute more to customer behavior than other remote POIs. Therefore, feeding the global spatial features into the model may lead to suboptimal results due to other irrelevant regions. Rather than considering each grid of the area equally, the spatial attention could learn useful features from the important grids in view of the distance between the grid and the place. Let $\mathbf{D} \in R^{W\times H}$ denote the distance matrix between grids and the center, as the auxiliary features of the spatial-level attention.

Given the output $\mathbf{X}_{sc} \in R^{W\times H\times C}$ of the channel-level attention, we reshape $\mathbf{X}_{sc}$ to $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_i, \ldots, \mathbf{h}_m]$ by flattening the width and height of the original feature $\mathbf{X}_{sc}$, where $\mathbf{h}_i \in R^C$ and $m = W \times H$. Known the auxiliary feature $\mathbf{D}$, we also reshape it, denoted as $\mathbf{f_d} = [d_1, d_2, \ldots, d_i, \ldots, d_m]$, $d_i \in R^1$. Similar to the definition of the channel-level attention, the spatial-level attention network is defined as

$$\begin{aligned} &\mathbf{B} = \sigma((\mathbf{W}_{s1}\mathbf{H} + \mathbf{b}_{s1}) \oplus \mathbf{W}_{sd}\mathbf{f}_d) \\ &\mathbf{W}_{s1} \in R^{k\times C}, \mathbf{H} \in R^{C\times m}, \mathbf{b}_{s1} \in R^k, \mathbf{W}_{sd} \in R^{k\times m}, \mathbf{f}_d \in R^m \end{aligned} \tag{13}$$

$$\begin{aligned} &\boldsymbol{\gamma} = \text{soft max}(\mathbf{W}_{s2}\mathbf{B} + b_{s2}) \\ &\mathbf{W}_{s2} \in R^k, \mathbf{B} \in R^{k\times m}, b_{s2} \in R^1. \end{aligned} \tag{14}$$

d) Output of attention-based spatial embedding: Known the initial feature map $\mathbf{X}_s$, we can obtain the weighted feature map $\mathbf{X}_{scs}$ via the channel-level attention and spatial-level attention, calculated by (15). Finally, in order to retain the features of POIs of different categories, we use the fully connected network for each channel to obtain the final embedding vector of spatial features instead of the convolutional network, which is defined as (16). Note that the spatial embedding of the ith category of POI, $\mathbf{x}_{sm,i} \in R^D$ and field embedding of the jth field, $\mathbf{x}_{fm,j} \in R^D$ are of the same length D, and $\mathbf{x}_{sm} = [\mathbf{x}_{sm,1}, \mathbf{x}_{sm,2}, \ldots, \mathbf{x}_{sm,i}, \ldots, \mathbf{x}_{sm,c}]$

$$\mathbf{X}_{scs} = f(\mathbf{X}_s, \boldsymbol{\alpha}, \boldsymbol{\gamma}) \tag{15}$$

$$\mathbf{x}_{sm,i} = f\left(\mathbf{W}_{sm,i}\mathbf{X}_{scs,i} + b_{sm,i}\right). \tag{16}$$

B. Cross Network

CrossNet in our model explicitly learns the high-order feature interactions based on dense features and embedding of sparse features via a feed-forward neural network, without any other feature engineering. Before presenting the CrossNet in our model, we first introduce two interaction methods among features: 1) bite-wise interactions and 2) vector-wise interactions.

1) Bite-Wise Interactions: In the deep learning-based recommendation, the multifield categorical features are usually transformed into the high-dimensional and sparse features via field-aware one-hot encoding. To reduce the dimensionality, the embedding layer is widely used to transform these sparse features into dense vectors of real values, as mentioned above. Generally, the embedding features are directly fed into the feed-forward neural network to learn high-order feature interactions at the bit-wise level, such as the DNNs model. Specifically, feature interactions at the bite-wise level mean that the interaction occurs on an element rather than a whole feature vector. That is to say, even the elements within the same field embedding vector will influence each other.

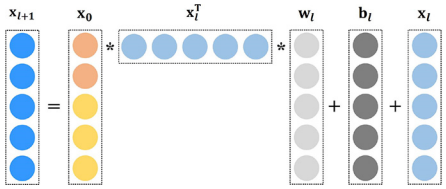


Fig. 4. Bite-wise interactions.

Different from the traditional DNNs model which implicitly learns the high-order feature interactions, DCN can model the high-order feature interactions explicitly at the bit-wise level. In particular, DCN contains the CrossNet, whose hidden layers are calculated by the following cross operation:

$$\begin{aligned} &\mathbf{x}_{l+1} = \mathbf{x}_0\mathbf{x}_l^T\mathbf{w}_l + \mathbf{b}_l + \mathbf{x}_l = f_c(\mathbf{x}_0, \mathbf{x}_l, \mathbf{w}_l, \mathbf{b}_l) + \mathbf{x}_l \\ &\mathbf{x}_l, \mathbf{x}_{l+1} \in R^d, \mathbf{w}_l, \mathbf{b}_l \in R^d \end{aligned} \tag{17}$$

where $\mathbf{x}_l$ and $\mathbf{x}_{l+1}$ are the outputs from the lth and $(l+1)$th cross layers, respectively. w_l and b_l are the weight and bias parameters. The process is shown in Fig. 4. Assuming that the first two elements in $\mathbf{x}_0$ are the real values of dense features, others are elements of an embedding feature. We can see that the elements within the embedding feature will influence each other; however, it will not happen to the dense feature since it only has one element.

2) Vector-Wise Interactions: Traditional FM is the typical framework which models feature interactions at the vector-wise level, because the embedding vector is regarded as a unit for vector-wise interactions, formulated as (18). However, traditional FM hardly models the high-order feature interactions due to high computational complexity

$$\begin{aligned} &y_{FM} = \langle \mathbf{w}, \mathbf{x} \rangle + \sum_{j_1=1}^{d}\sum_{j_2=j_1+1}^{d} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_{j_1}x_{j_2} \\ &\mathbf{w} \in R^d, \mathbf{v}_i, \mathbf{v}_j \in R^k. \end{aligned} \tag{18}$$

On the basis of DCN and FM, xDeepFM is proposed which can jointly learn explicit and implicit high-order feature interactions effectively. Especially, xDeepFM includes a CIN that learns high-order feature interactions at the vector-wise level, and the vector-wise interaction is illustrated in Fig. 5. Each layer in CIN has the following formula:

$$\mathbf{X}_{l+1}^k = \sum_{i=1}^{h_l}\sum_{j=1}^{m} \mathbf{W}_l^{ij}\left(\mathbf{X}_{l-1}^i o \mathbf{X}_0^j\right) \tag{19}$$

where $\mathbf{W}_l \in R^{h_l\times m}$ is the parameter matrix for the hth feature vector, and o denotes the Hadamard product. Different with CrossNet in DCN, the output of field embedding is transformed as a matrix $\mathbf{X}_0 \in R^{m\times D}$, which is fed into the CIN to achieve feature interactions at the vector-wise level, since an embedding vector is regarded as a unit for vector-wise interactions.

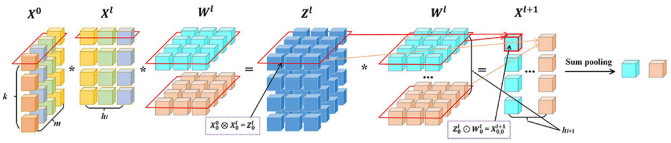


Fig. 5. Vector-wise interactions.

3) Hybrid Interactions: For the deep learning-based recommendation, features explicitly interact at the vector-wise level is more appropriate than the bit-wise level, because most input features are sparse, of huge dimension. Therefore, we can define feature interactions at the vector-wise level based on the embedding features. However, there is no need to map a real value of the dense feature to a multiple dimension vector via the embedding method, which could increase the complexity of computation.

There are many both dense features and sparse features in our problem, in order to reduce the computational complexity, we propose the framework of hybrid interactions, which can make sparse features interact at the vector level, and dense features interact at the bite level, as shown in Fig. 6. Two types of features are fed into the CrossNet: 1) the dense features from input features (indicated by the brown circles), denoted as a vector $\mathbf{b}_o$, and 2) the embedding vectors from the embedding layer (indicated by the yellow circles), denoted as a matrix $\mathbf{X}_o$. Note the we first concatenate the output of field embedding $\mathbf{x}_{fm}$ and attention-based spatial embedding $\mathbf{x}_{sm}$, then formulate it as a matrix $\mathbf{X}_o \in R^{(m+c)\times D}$, where D is the dimension of the embedding, and m, c is the number of categorical feature fields and POI category, respectively.

For each layer of the CrossNet, it first learns the feature interactions at vector level based on $\mathbf{X}_o$, which is defined in (20). The detailed computation procedure is illustrated in Fig. 5

$$\mathbf{X}_{l+1} = f_d(\mathbf{X}_o, \mathbf{X}_l, \mathbf{W}_l). \tag{20}$$

Considering that the output of sparse feature interactions at the vector level is a matrix $\mathbf{X}_l$, and the output of dense feature interaction at the bite level is a vector $\mathbf{b}_l$. Therefore, we first apply sum pooling on $\mathbf{X}_l$, and then concatenate it and dense features as the input of feature interactions at the bite level. Finally, the output of the hidden layer $\mathbf{b}_l$ is calculated in (23)

$$\begin{aligned} \mathbf{a}_l &= \text{sum pooling } (\mathbf{X}_l) \\ &= \text{sum pooling } \left(f_d\left(\mathbf{X}_0, \mathbf{X}_{l-1}, \mathbf{W}^c_{l-1}\right)\right) \quad (21) \\ \mathbf{g}_l &= [\mathbf{b}_l, \mathbf{a}_l] \quad (22) \\ \mathbf{b}_{l+1} &= f_c\left(\mathbf{g}_0, \mathbf{g}_l, \mathbf{w}^c_l, \mathbf{b}^c_l\right) + \mathbf{g}_l. \quad (23) \end{aligned}$$

C. Deep Network

The deep network is a fully connected feed-forward neural network, which is used to learn high-order feature interactions implicitly. The forward process is

$$\mathbf{h}_1 = \delta\left(\mathbf{W}^d_1 \mathbf{x}_e + \mathbf{b}^d_1\right) \tag{24}$$

$$\boldsymbol{h}_l = \delta\left(\mathbf{W}^d_l \mathbf{h}_{l-1} + \mathbf{b}^d_l\right) \tag{25}$$

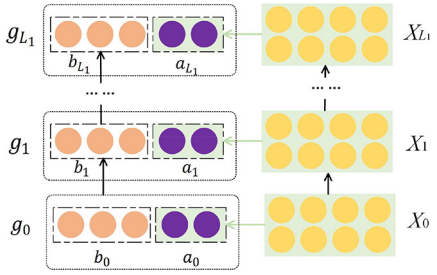


Fig. 6. Hybrid interactions.

where $\mathbf{x}_e = [\mathbf{b}_o, \mathbf{x}_{fm}, \mathbf{x}_{sm}]$ is the output of the embedding layer, $\mathbf{b}_o$ is the dense features, $\mathbf{x}_{fm}$ and $\mathbf{x}_{sm}$ are outputs of the field embedding and attention-based spatial embedding. $\boldsymbol{h}_l \in R^{n_l}$ and $\boldsymbol{h}_{l-1} \in R^{n_{l-1}}$ are the lth and $(l-1)$th hidden layer, respectively, and $\mathbf{W}_l \in R^{n_l \times n_{l-1}}$ and $\mathbf{b}_l \in R^{n_l}$ are parameters of the deep network. δ is the activation function.

D. Combination Layer

Inspired by the Wide&Deep and xDeepFM model, we apply the combination layer to combine the outputs from three components: linear component, CrossNet, and deep network. On the one hand, the CrossNet and deep network can complement each other, since they learn the high-order feature interactions explicitly and implicitly, respectively. On the other hand, linear component could learn linear relationships from raw features without the embedding method.

Therefore, the last hidden layer is the combination layer taking the following function:

$$y'_j = \delta\left(w^T_{\text{linear},j}\mathbf{l} + w^T_{\text{cross},j}\mathbf{g} + w^T_{\text{dnn},j}\mathbf{h} + b_j\right) \tag{26}$$

where $\mathbf{l}$ is the linear combination of raw features with different weights. $\mathbf{h}$ is the output of the deep network. In order to learn both low- and high-order feature interactions, we concatenate the outputs of each hidden layer in the CrossNet, $\mathbf{g} = [\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{L_1}]$.

The output of our proposed model is an s-way softmax which predict the probability distribution over s different levels of user consumption

$$\widehat{y_j} = \frac{\exp\left(y'_j\right)}{\sum_{j=1}^{s} \exp\left(y'_j\right)}. \tag{27}$$

We define the loss function as the cross-entropy between labels $y_{i,j}$ and the predicted results $\hat{y}_{i,j}$ by (28), where N is the total number of training instances.

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{s} y_{i,j} \log \hat{y}_{i,j}. \tag{28}$$

Finally, the optimization process is to minimize the following objective function in (29), where Θ is the set of parameters in the model and λ is the regularization term

$$\mathrm{L}(\Theta) = \mathcal{L} + \lambda\Theta^2. \tag{29}$$

V. Experiments

In this section, we first give a summary of the experimental purposes to evaluate the performance of our proposed model, DeepStore. Then, we introduce the experiment settings in our experiment. Next, we compare our proposed model and the other state-of-the-art models empirically. Finally, we make a discussion of the deep insights and limitations of this paper.

A. Experimental Purposes

We conduct extensive experiments to answer the following questions.

1) How does our proposed model perform as compared to the state-of-the-art methods?
2) Can the attention mechanism effectively learn the importance of spatial features?
3) How do the key hyper-parameters of our model impact its performance?
4) Are there significant performance differences when choosing different locations in different cities?

B. Experimental Settings

1) Baseline Algorithms: We use seven models as the baselines in our experiments, including logistic regression (LR), gradient boosting decision tree (GBDT), DNN, Wide&Deep, DeepFM, DCN, and xDeepFM. These mode are highly related to our proposed model, and some of them are state-of-the-art models for deep learning-based recommender systems.

2) Evaluation Metrics: We measure the prediction performance of our model and baselines using the two metrics.

1) *Accuracy:* Our problem is a multiclassification problem, so we use the accuracy to measure the performance of the prediction result. It calculates the number of correct predictions divided by the total number of predictions, as formulated in (30), where y_j is the real level of user consumption and $\widehat{y_j}$ is the predicted result. A higher accuracy means the better performance

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} \boldsymbol{I}\left(y_j = \widehat{y_j}\right)}{N}. \tag{30}$$

2) *Error:* Different from the general classification problem, the class in our problem represents the level of user consumption. Specifically, we divide the amount of consumption into s levels based on equidistance partition, denoted as $\{y_1, y_2, \ldots, y_k, \ldots, y_s\}$. Therefore, it also involves the relationship of size between different classes. For example, the difference between y_1 and y_4 is larger than is larger than that between y_1 and y_2. In general, error measures the average absolute difference between the y_j and $\widehat{y_j}$ by (31). A smaller error means the better performance.

$$\text{Error} = \frac{\sum_{i=1}^{N} |y_j - \widehat{y_j}|}{N}. \tag{31}$$

3) Parameter Setting and Training: Our models are learned by optimizing the loss function of (29), which is implemented using Tensorflow. Based on the empirical knowledge and our experiments, we first select communities within 5 km of the store as instances in the experiments. Then, we extract the geographic features of surrounding areas, which lie in a disk of radius 3 km around the store and the community. In addition, we set the location grid size to 500 m $\times$500 m in the surrounding areas to extract spatial features. Inspired by the rating value widely used in most works, we divide the amount of user consumption into five levels based on equidistance partition.

We split the dataset into three portions: 1) 70% for training; 2) 20% for validation; and 3) 10% for testing. However, different from other works which split the dataset randomly in view of all instances, we split the dataset in view of all stores. Because the information of consumer behaviors in a store consists of multiple instances, and an instance represents the users' consumption of each community in the store. In order to understand the consumption behaviors of potential consumers at the store and guide the selection of store location, we need to predict the consumption behavior of users in all nearby communities around the given store. Therefore, we split the dataset in view of all the stores. For example, there are 100 stores in the dataset, training instances include the data of 70 stores, and testing instances include the data of ten stores.

To be fair, we use the same setting when comparing the performance of different models. Specifically, we apply mini-batch stochastic optimization with Adam optimizer, and the learning rate is set to 0.001. Moreover, we use L2 regularization with $\lambda = 0.001$. Finally, to obtain the best prediction results, the hyper-parameters of our model are tuned on the validation set, and the best settings will be shown in the corresponding sections.

TABLE III
Performance Comparison of Different Models

Models		Accuracy	Error
Traditional models	LR	0.5470	0.6006
	GBDT	0.5614	0.5315
Deep models	DNN	0.5699	0.5383
	Wide&Deep	0.5973	0.4998
	DeepFM	0.5788	0.5171
	DCN	0.5761	0.5138
	xDeepFM	0.6034	0.4879
	DeepStore	**0.6080**	**0.4806**

C. Experimental Results

Having depicted the experiment settings and baselines, we present the experimental results regarding the four experiment purposes given in Section V-A.

1) Performance Comparison of Different Models: We want to compare the performance of different models. The result is shown in Table III, where we have the following observations.

1) We can find that deep models (e.g., DNN, Wide&Deep, DeepFM, DCN, xDeepFM, and DeepStore) outperform traditional models (e.g., LR and GBDT), which demonstrates that the deep network has the advantage in capturing nonlinear relations from multisource data.

2) LR is far worse than all the rest models, since it only learns the linear relationship between labels and continuous features. However, therefore are some discrete features in our problem which leads to bad performance of LR.
3) GBDT is a strong tree model which is widely used in many works, since it can deal with various types of data flexibly, including continuous and discrete features. In particular, it has strong robustness to noisy data. The data in our experiments is real data obtained from the store, which has a lot of noisy data. Therefore, GBDT can achieve better performance even compared with deep models.
4) Wide&Deep, DCN, DeepFM, xDeepFM, and DeepStore are significantly better than DNN, which directly rejects that, despite their simplicity, incorporating hybrid components are important for boosting the accuracy of predictive results.
5) Another interesting observation is that Wide&Deep outperforms DeepFM and DCN. The differences among these three models are: Wide&Deep learns the feature interaction from initial extracted features, and DeepFM and DCN models 2 and 4 feature interactions from embedding vectors in this experiment. Because the input data of deep models in our experiments contains some useful features which are extracted based on expert knowledge, so Wide&Deep could achieve better performance by exploiting useful features directly. Although DeepFM and DCN could learn high-order feature interaction, they may fail to acquire initial and useful features.
6) xDeepFM and DeepStore outperform Wide&Deep, DeepFM, and DCN, since they congregate advantages of Wide&Deep, DeepFM, and DCN. More specifically, xDeepFM and DeepStore not only learn relationship from extracted features directly, but also model low-order and high-order feature interactions simultaneously.
7) As we can see, our proposed model, DeepStore, achieves the best performance compared to all baselines, which demonstrates that combining explicit and implicit high-order feature interaction is necessary. Especially, DeepStore outperforms the xDeepFM, which is a state-of-the-art model in the deep learning-based recommendation. The results indicate that interactions over sparse and dense features are necessary, and the different embedding methods in view of different data types are important.
8) Different from xDeepFM, our model has the ability to extract valuable features and model feature interactions from multimodal data, such as learning representative spatial features from geographical data. However, the result of our method is close to the result of xDeepFM. The main reason that DeepStore gains less obvious improvement could be that the data used in this paper suffers from scale and data sparsity issues, which impacts the significant improvement of the experimental performance. Specifically, we design various components (e.g., spatial embedding, CrossNet, etc.) in the DeepStore model to extract valuable features from different types of data and learn complex feature interactions, because the store site recommendation problem is complicated, and customer behaviors are affected by sophisticated factors in real situations. This leads to more parameters to be learned in our model, which may also impacts the performance when the dataset scale is not big.

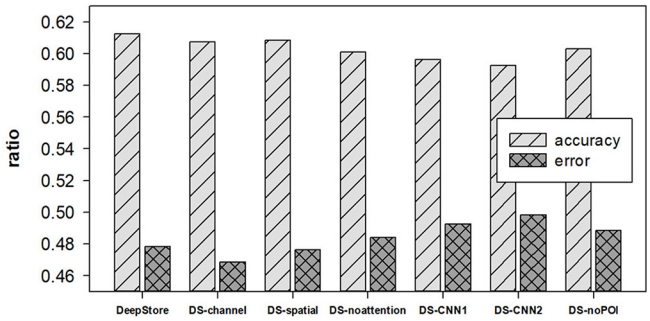


Fig. 7. Performance comparison of DeepStore and its variants.

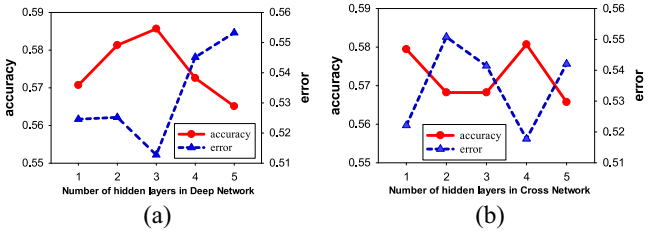
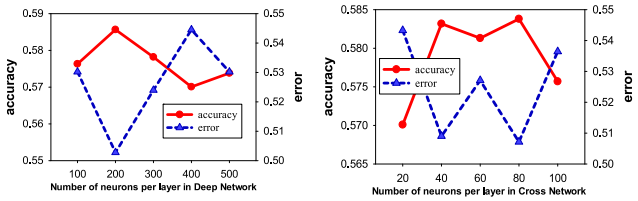


Fig. 8. Impact of number of hidden layers in the deep network and the CrossNet. (a) Deep network. (b) Cross network.

2) Impact of the Attention Mechanism: We want to know whether the attention mechanism effectively learns the importance of spatial features and improve the performance of the model. Fig. 7 shows the performance of DeepStore and its variants.

1) *DeepStore* considers both channel attention and spatial attention in the attention-based spatial embedding.
2) *DS_channel* just considers the channel attention, and *DS_spatial* considers the spatial attention.
3) *DS_noattention* means that the fine-grained POI data is fed into the model without the attention mechanism.
4) *DS_CNN1* means that there is one convolution layer instead of spatial embedding.
5) *DS_CNN2* means that there are two convolution layers instead of spatial embedding.
6) *DS_noPOI* means that there is not fine-grained POI data fed into the model.

First, we can see that DeepStore outperforms its variants, which demonstrates that the attention mechanism is effective in our problem to learn representative features from the fine-grained POI data. Furthermore, DS_channel and DS_spatial achieve better performance than DS_noattention, which indicates that both channel attention and spatial attention are useful in the spatial embedding. From Fig. 7, we can find that DS_CNN1 and DS_CNN2 perform worse than others in accuracy, because spatial features in our problem are different from features in the image. For example, CNN can extract features



(a) (b)

Fig. 9. Impact of number of neurons per layer in the deep network and the CrossNet. (a) Deep network. (b) Cross network.

in different regions based on a filter. However, it is not suitable to extract spatial features in POI data, even POI data in our paper is regarded as the image pixel data. Because the given place stands in the center of the image, the distance between different regions and the given place plays the different role in extracting spatial features. Note that DS_noPOI performs worse than DeepStore, DS_channel, and DS_spatial, but performs better than DS_noattention. This show that designing the efficient framework of the deep network is more important, compared to feeding all raw data into the deep network, which could lead to worse results.

3) Hyper-Parameter Investigation: We study the impact of hyper-parameters on DeepStore, including the number of hidden layers, the number of neurons per layer, dropout, and activation functions.

a) Number of hidden layers: Fig. 8 demonstrates the impact of the number of hidden layers in the deep network and the CrossNet. In Fig. 8(a), we can find that increasing number of hidden layers in the deep network improves the performance of models at the beginning. However, model performance degrades when the depth of the deep network is set greater than 3. This phenomenon is because of overfitting. Similarly, from Fig. 8(b), we can see that DeepStore achieves the better performance when the depth of the CrossNet is set to 4.

b) Number of neurons per layer: As shown in Fig. 9(a), model performance increases when the number of neurons per layer in the deep network is increased from 100 to 200, but it performs worse when we increase the number of neurons from 200 to 400. This is because an over-complicated model is easy to overfit. The increasing the number of neurons does not always bring benefit, as shown in Fig. 9(b), and 80 is a more suitable setting for the number of neurons per layer in the CrossNet.

c) Dropout: Dropout is a regularization technique to compromise the precision and complexity of the neural network. Fig. 10 shows the performance of the model when the dropout is set to different value. We can observe that our model achieves the best performance when the dropout is set to 0.8, because adding reasonable randomness to the model can strengthen the model's robustness.

d) Activation function: We compare the performance of our model when applying different activation functions. As shown in Fig. 11, we can find that Relu is the most suitable one for neurons in our model.

4) Performance on Different Stores: We test our model on different stores in different cities, the result is shown in Table IV.

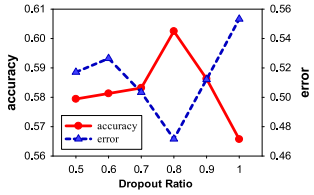


Fig. 10. Impact of differen dropout ratio.

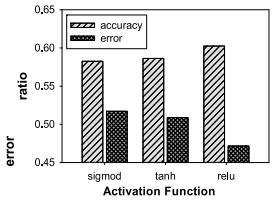


Fig. 11. Impact of different activation function.

TABLE IV
RESULTS ON DIFFERENT STORES

City	Store ID	Accuracy	Error
Beijing	1	0.6163	0.4918
	2	0.6012	0.4993
Shanghai	3	0.6237	0.4510
	4	0.5507	0.5507
Ningbo	5	0.5479	0.5765
Hangzhou	6	0.5409	0.6112

We select six stores in four cities from our dataset as test data. From Table IV, we can see that the result in Beijing and Shanghai is better than the result in Ningbo and Hangzhou. The main reason is that customer behavior differs among different cities even in the same branding stores. In our dataset, the number of stores in Beijing and Shanghai (e.g., 25) is great than that in Ningbo and Hangzhou (e.g., 5), thus the model hardly learns some local knowledge in Ningbo and Hangzhou due to fewer data.

D. Discussion

We next discuss the research findings from this paper and potential future directions to improve this paper.

1) Transferring Knowledge Among Different Cities: In the experiment, we find that the performance of our proposed model differs in different cities, mainly because of distinct city characteristics. In addition, deep models rely on a large amount of data to learn knowledge. However, generally, we do not have enough data to train the model for different types of stores, respectively. Therefore, we are planning to combine deep learning and transfer learning to transfer valuable knowledge from the city with rich data to the city with fewer data, and improve the model' ability of robustness and generalization.

2) Time Series Modeling: This paper studies the consumption of user in the store in a short period of time, which is static. However, customer behavior will change over time, which also influences the location selection and development of the business. In our future work, we intend to model the

dynamic customer behavior based on the sequence model, such as RNN and long short-term memory (LSTM).

3) Improving Results: This paper is a pilot study for fine-grained store site recommendation, where the problem definition is more difficult and the real-world commercial dataset is not easy to be obtained. First, it is quite difficult to predict the amount of user consumption, because customer behaviors have a certain degree of randomness which is affected by sophisticated factors in real situations. Therefore, the results in our experiments seem not as good as those of traditional coarse-grained problem definitions. Second, the real-world commercial data for fine-grained store site recommendation is quite difficult to be obtained and the commercial data used in this paper is based on one type of new retail [48], which has grown in recent two years and the existing stores are limited in number and data quantity (see Table I). In other words, the data used in this paper suffers from data sparsity, scale, and noisy issues, which also affects the experimental performance. In order to improve the results, on the one hand, we intend to have more collaboration with commercial companies to gain data from maturing commercial entities, and on the other hand, we intend to optimize the prediction model to adapt to sparse data.

4) Extension and Usage of the DeepStore to Other Applications: DeepStore is a deep model which can learn nonlinear relations and high-order features interaction from multisource data. Although our model is proposed to predict the level of user's consumption, it is also applicable to other application, such as POI recommendation and CTR prediction. Different applications have different data characteristics, so we intend to extend our model and apply it to different application areas.

VI. Conclusion

In this paper, we propose a novel network named DeepStore, which aims to learn nonlinear relations and high-order feature interactions from multisource data. Particularly, DeepStore can automatically learn high-order feature interactions in both explicit and implicit fashions, which is of great significance to reduce manual feature engineering work. Finally, we conduct comprehensive experiments on the real-world dataset to compare the performance of DeepStore and state-of-the-art models from different perspectives. The results demonstrate the effectiveness of our approach. As for the future work, we intend to combine deep learning and transfer learning to transfer valuable knowledge from the city with rich data to the city with fewer data, and model the dynamic customer behavior based on the sequence model.

References

[1] L. Wang, B. Guo, and Q. Yang, "Smart city development with urban transfer learning," *IEEE Computer*, vol. 51, no. 12, pp. 32–41, Dec. 2018.

[2] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 2, pp. 112–121, Apr. 2014.

[3] B. Guo *et al.*, "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm," *ACM Comput. Surveys*, vol. 48, no. 1, pp. 1–31, 2015.

[4] X. Yang, X. Wang, Y. Wu, L. P. Qian, W. Lu, and H. Zhou, "Small-cell assisted secure traffic offloading for narrowband Internet of Thing (NB-IoT) systems," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1516–1526, Jun. 2018.

[5] B. Guo, Y. Liu, Y. Ouyang, V. W. Zheng, D. Zhang, and Z. Yu, "Harnessing the power of the general public for crowdsourced business intelligence: A survey," *IEEE Access*, vol. 7, pp. 26606–26630, 2019.

[6] B. Thau. (2015). *How Big Data Helps Chains Like Starbucks Pick Store Locations—An (Unsung) Key to Retail Success*. [Online]. Available: https://www.forbes.com/sites/barbarathau/2014/04/24/how-big-data-helps-retailers-like-starbucks-pick-store-locations-an-unsung-key-to-retail-success/

[7] P. Jensen, "Network-based predictions of retail store commercial categories and optimal locations," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, no. 3, 2006, Art. no. 035101.

[8] D. Karamshuk, A. Noulas, S. Scellato, V. Nicosia, and C. Mascolo, "Geo-spotting: Mining online location-based services for optimal retail store placement," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2013, pp. 793–801.

[9] J. Li, B. Guo, Z. Wang, M. Li, and Z. Yu, "Where to place the next outlet? Harnessing cross-space urban data for multi-scale chain store recommendation," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. Adjunct*, 2016, pp. 149–152.

[10] Y. Li, Y. Zheng, S. Ji, W. Wang, L. H. U, and Z. Gong, "Location selection for ambulance stations: A data-driven approach," in *Proc. 23rd SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2015, p. 85.

[11] B. Cuo, J. Li, V. W. Zheng, Z. Wang, and Z. Yu, "Citytransfer: Transferring inter- and intra-city knowledge for chain store site recommendation based on multi-source urban data," in *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 4, 2018, p. 135.

[12] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[14] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 160–167.

[15] J. Lian, F. Zhang, X. Xie, and G. Sun, "Towards better representation learning for personalized news recommendation: A multi-channel deep fusion approach," in *Proc. IJCAI*, 2018, pp. 3805–3811.

[16] W. Zhang, T. Du, and J. Wang, "Deep learning over multi-field categorical data," in *Proc. Eur. Conf. Inf. Retrieval*, 2016, pp. 45–57.

[17] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 355–364.

[18] H.-T. Cheng *et al.*, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10.

[19] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorizationmachine based neural network for CTR prediction," *arXiv preprint arXiv:1703.04247*, Mar. 2017.

[20] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proc. ADKDD*, 2017, p. 12.

[21] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "xDeepFM: Combining explicit and implicit feature interactions for recommender systems," in *Proc. SIGKDD*, 2018, pp. 1754–1763.

[22] H. Yin, B. Cui, Y. Sun, Z. Hu, and L. Chen, "LCARS: A spatial item recommender system," *ACM Trans. Inf. Syst. (TOIS)*, vol. 32, no. 3, p. 11, 2014.

[23] H. Yin, X. Zhou, B. Cui, H. Wang, K. Zheng, and Q. V. H. Nguyen, "Adapting to user interest drift for POI recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 10, pp. 2566–2581, Oct. 2016.

[24] H. Yin, B. Cui, X. Zhou, W. Wang, Z. Huang, and S. Sadiq, "Joint modeling of user check-in behaviors for real-time point-of-interest recommendation," *ACM Trans. Inf. Syst.*, vol. 35, no. 2, p. 11, 2016.

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

[26] S. Feng, G. Cong, B. An, and Y. M. Chee, "POI2Vec: Geographical latent representation for predicting future visitors," in *Proc. AAAI*, 2017, pp. 102–108.

[27] T. Qian, B. Liu, Q. V. H. Nguyen, and H. Yin, "Spatiotemporal representation learning for translation-based POI recommendation," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, p. 18, 2019.

[28] H. Yin, W. Wang, H. Wang, L. Chen, and X. Zhou, "Spatial-aware hierarchical collaborative deep learning for POI recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 11, pp. 2537–2551, Nov. 2017.

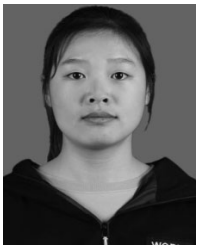
[29] A. Sevtsuk, "Location and agglomeration: The distribution of retail and food businesses in dense urban environments," *J. Plan. Educ. Res.*, vol. 34, no. 4, pp. 374–393, 2014.
[30] Y. Li and L. Liu, "Assessing the impact of retail location on store performance: A comparison of Wal-Mart and Kmart stores in Cincinnati," *Appl. Geography*, vol. 32, no. 2, pp. 591–600, 2012.
[31] N. Roig-Tierno, A. Baviera-Puig, J. Buitrago-Vera, and F. Mas-Verdu, "The retail site location decision process using GIS and the analytical hierarchy process," *Appl. Geography*, vol. 40, pp. 191–198, Jun. 2013.
[32] F. Wang, L. Chen, and W. Pan, "Where to place your next restaurant? Optimal restaurant placement via leveraging user-generated reviews," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manag.*, 2016, pp. 2371–2376.
[33] J. Lin, R. Oentaryo, E.-P. Lim, C. Vu, A. Vu, and A. Kwee, "Where is the goldmine? Finding promising business locations through facebook data analytics," in *Proc. 27th ACM Conf. Hypertext Soc. Media*, 2016, pp. 93–102.
[34] M. Xu, T. Wang, Z. Wu, J. Zhou, J. Li, and H. Wu, "Store location selection via mining search query logs of Baidu maps," *arXiv preprint arXiv:1606.03662*, Jun. 2016.
[35] L. Nie *et al.*, "Enhancing micro-video understanding by harnessing external sounds," in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 1192–1200.
[36] W. Wang, H. Yin, X. Du, W. Hua, Y. Li, and Q. Nguyen, "Online user representation learning across heterogeneous social networks," in *Proc. SIGIR*, 2019.
[37] L. Nie, X. Song, and T.-S. Chua, "Learning from multiple social networks," in *Synthesis Lectures on Information Concepts, Retrieval, and Services*, vol. 8. San Rafael, CA, USA: Morgan & Claypool, 2016, pp. 1–118.
[38] X. Du, H. Yin, L. Chen, Y. Wang, Y. Yang, and X. Zhou, "Personalized video recommendation using rich contents from videos," *IEEE Trans. Knowl. Data Eng.*, to be published.
[39] A. Wu and Y. Han, "Multi-modal circulant fusion for video-to-language and backward," in *Proc. IJCAI*, vol. 3, no. 4, pp. 1029–1035, 2018.
[40] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," *arXiv preprint arXiv:1708.04617*, Aug. 2017.
[41] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 335–344.
[42] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
[43] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3128–3137.
[44] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Machine Learn.*, 2015, pp. 2048–2057.
[45] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 842–850.
[46] L. Chen *et al.*, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 6298–6306.
[47] K. Chen, J. Wang, L.-C. Chen, H. Gao, W. Xu, and R. Nevatia, "ABC-CNN: An attention based convolutional neural network for visual question answering," *arXiv preprint arXiv:1511.05960*, Nov. 2015.
[48] X. Wang and C. T. Ng, "New retail versus traditional retail in e-commerce: Channel establishment, price competition, and consumer recognition," *Annals of Operations Research*. New York, NY, USA: Springer, 2018, pp. 1–17.

Yan Liu received the B.E. degree from Northwestern Polytechnical University, Xi'an, China, where she is currently pursuing the Ph.D. degree at the School of Computer Science.

Her current research interests include recommendation, commercial big data, and mobile computing.

Bin Guo (GS'09–M'09–SM'14) received the Ph.D. degree in computer science from Keio University, Tokyo, Japan, in 2009

He is a Professor from Northwestern Polytechnical University, Xi'an, China. He was a Post-Doctoral Researcher with Télécom SudParis, Évry, France. His current research interests include ubiquitous computing and mobile crowd sensing.



Nuo Li received the bachelor's degree from Northwestern Polytechnical University, Xi'an, China, where she is currently pursuing the Ph.D. degree at the School of Computer Science.

Her current research interests include urban computing and commercial big data.

Jing Zhang received the bachelor's degree from Northwestern Polytechnical University, Xi'an, China, where he is currently pursuing the graduate degree at the School of Computer Science.

His current research interests include mobile crowd sensing and social media mining.

Jingmin Chen, photograph and biography not available at the time of publication.

Daqing Zhang (M'11–SM'16–F'19) received the Ph.D. degrees from the University of Rome "La Sapienza," Rome, Italy, and the University of L'Aquila, Rome, in 1996.

He is currently a Full Professor with Télécom SudParis, Évry, France. His current research interests include context-aware computing, urban computing, and mobile computing.

Yinxiao Liu, photograph and biography not available at the time of publication.

Zhiwen Yu (S'03–M'06–SM'11) received the Ph.D. degree of Engineering in computer science and technology from Northwestern Polytechnical University, Xi'an, China, in 2005.

He is currently a Professor with the School of Computer Science, Northwestern Polytechnical University. His current research interests include ubiquitous computing and HCI.

Sizhe Zhang, photograph and biography not available at the time of publication.

Lina Yao (M'14) received the master's and Ph.D. degrees from the University of Auckland (UoA), Auckland, New Zealand, in 2010 and 2014, respectively.

She is currently a Senior Lecturer with the School of Computer Science and Engineering, University of New South Wales, Sydney, NSW, Australia. Her current research interests include data mining and machine learning, recommender systems, and human activity recognition.