Spatial-Aware Hierarchical Collaborative Deep Learning for POI Recommendation

Hongzhi Yin, Weiqing Wang, Hao Wang, Ling Chen and Xiaofang Zhou

Abstract—Point-of-interest (POI) recommendation has become an important way to help people discover attractive and interesting places, especially when they travel out of town. However, the extreme sparsity of user-POI matrix and cold-start issues severely hinder the performance of collaborative filtering-based methods. Moreover, user preferences may vary dramatically with respect to the geographical regions due to different urban compositions and cultures. To address these challenges, we stand on recent advances in deep learning and propose a Spatial-Aware Hierarchical Collaborative Deep Learning model (SH-CDL). The model jointly performs deep representation learning for POIs from heterogeneous features and hierarchically additive representation learning for spatial-aware personal preferences. To combat data sparsity in spatial-aware user preference modeling, both the collective preferences of the public in a given target region and the personal preferences of the user in adjacent regions are exploited in the form of social regularization and spatial smoothing. To deal with the multimodal heterogeneous features of the POIs, we introduce a late feature fusion strategy into our SH-CDL model. The extensive experimental analysis shows that our proposed model outperforms the state-of-the-art recommendation models, especially in out-of-town and cold-start recommendation scenarios.

Index Terms—POI recommendation, Spatial-Aware User Modeling, Deep Learning, Multimodal Modeling, Cold Start.

1 INTRODUCTION

Recent years have witnessed the fast development and popularity of location-based social networks (LBSNs), such as Yelp, Foursquare and Facebook Places. In LBSNs, users can post their physical locations in the form of a “check-in”, and share their life experiences in the physical world. Using user check-in data to make personalized POI recommendation in LBSNs is crucial for helping users to explore new POIs and regions (e.g., cities), and for facilitating location-based services (e.g., launching personalized mobile advertisements). Personalized POI recommendation is especially important and useful when a user travels to an unfamiliar area, where she has little knowledge about the neighbourhood. In this scenario, the recommendation task is referred to as recommendations for out-of-town users [10]. Correspondingly, the problem of making recommendations to users in their local area is called recommendations for home-town users. In this paper, we aim to offer high quality recommendations for both home-town and out-of-town users by effectively leveraging their check-in data. This problem is highly challenging for three main reasons:

Spatial Dynamic of Users’ Preferences. Users tend to have different preferences when they travel in regions with different urban compositions and cultures, as noted in [34]. For example, a person who never gambles in her home city may be quite interested in visiting a casino when she travels to Macao or Las Vegas. Thus, users’ preferences learned from their check-in history in one region (e.g., home town) may not always be effective for making recommendations in another region.

Data Sparsity. While LBSNs expand rapidly, the number of POIs visited by an individual user is rather small compared to the total number of POIs in an LBSN, which results in a very sparse user-POI matrix. Moreover, the travel locality indicates that most of the users’ check-ins happen in their local regions (e.g., home cities) [10], [44]. This observation aggravates the data sparsity problem further for out-of-town POI recommendations.

Cold Start Problem. Cold start is a critical problem in the domain of POI recommendation, and consists of the cold-start POI problem and the cold-start user problem. POIs that have not received any check-ins are called cold-start POIs. Similarly, users who have not rated any POIs are called cold-start users. Besides new users, out-of-town users may encounter the same problem as cold-start users, even though they have historical check-in data in other regions. This is because, as discussed in the issue of spatial dynamic of user preferences, users’ preferences learned from their check-ins in their home towns can seldom be taken as references to produce recommendations in out-of-town regions, especially when the two regions are far away from each other.

Several methods [2], [10], [34], [35], [44], [45], [52] have recently been developed to make POI recommendations for both home-town and out-of-town users. However, these approaches either do not address all the aforementioned challenges, or they do not address these challenges effectively. For example, most existing user behavior modeling research [11], [48] focuses on the temporal dynamics of user interests in the online world, yet ignores the spatial dynamics of user preferences in the physical world. To handle the spatial dynamics of user preferences, the state-of-the-art POI recommender models, such as LCA-LDA [44], Geo-SAGE [34] and ST-SAGE [35] exploit the general public’s preferences in the target region as a proxy for personal preferences. However, this strategy fails to identify the different preferences of different users to provide personalized recommendations. In our recent work [45], we develop the model, ST-LDA, to leverage the textual content of visited POIs to learn region-dependent personal interests that are modeled as a topic distribution. Although the method addresses the spatial dynamics of user preferences to some extent, it is ineffective in incorporating the collaborative filtering
information that is essential for POI recommendation.

A widely adopted approach to alleviate the issues of data sparsity and cold start, especially in the out-of-town recommendation scenario, is to exploit and integrate the auxiliary information of POIs. Some recent studies [2], [34], [35], [44], [45] have leveraged the content information of POIs to infer user interests for POI recommendations. However, existing methods extract the semantic features from the textual content in an unsupervised and general-purpose way that is independent of the recommendation task, thus they are not sufficiently effective to improve the recommendation. In this paper, we exploit four types of POI features with different modalities in a task-guided and supervised way.

Recently, deep learning models have demonstrated great potential for learning effective representations and delivered the state-of-the-art performance in traditional recommendation tasks. For instance, Wang et al. [33] integrated stacked denoising auto-encoders (SDA) into a matrix factorization model. In their method, denoising auto-encoders are used to extract the items’ latent factors from their content information. Li et al. [17] extended the work [33] to unify the marginalized denoising auto-encoders (mDA) with a matrix factorization model. Kim et al. [15] combined convolutional neural network model (CNN) with matrix factorization. Although these methods use deep learning techniques to learn the item representation, only one type of item feature is considered as auxiliary information. We note that there are multiple types of POI features with diverse input modalities, such as textual content, geographical features and neighborhood features, which may contribute rich information to improve POI recommendation. How to extract a unified POI representation from the heterogeneous features for POI recommendation is more challenging and has not been studied. Besides, these methods assume that personal preferences are stable and ignore their spatial dynamics over geographical regions.

To address these challenges of POI recommendation in an effective and unified way, we propose a novel POI recommendation framework to jointly perform spatial-aware personal preference modeling and POI semantic representation learning from versatile POI auxiliary information. Specifically, to address the spatial dynamics of user preferences, we model spatial-aware user preferences by allowing a user to have different preference vectors for different regions. Given the check-in records generated by an individual user in an out-of-town region are extremely sparse, the preferences of a crowd sharing the same role as the user in that region (i.e., the wisdom of the crowd) [34] are employed to enhance the inference of spatial-aware personal preferences in the form of social regularization. To further alleviate the issue of data sparsity, we exploit geographical correlation to smooth spatial-aware personal preferences over a spatial index structure - spatial pyramid, inspired by [34]. To handle the cold start issue, we adopt Deep Belief Networks (DBN) [13] to learn the POIs’ semantic representations from a set of heterogeneous features, inspired by the better performance of DBN than SDA in the classification task and its robustness to noise [9]. To fuse multiple types of POI features and learn a unified semantic representation of individual POIs, we extend the conventional DBN into a multi-modal DBN (MDBN) by introducing a late feature fusion strategy. Finally, to integrate the learned spatial-aware personal preferences and POIs’ semantic representation, we propose a probabilistic matrix factorization model, called Spatial-Aware Hierarchical Collaborative Deep Learning (SH-CDL), to jointly perform matrix factorization and automatic extraction of the POIs’ semantic representation. Our SH-CDL enables the extraction of POIs’ semantic representation to be guided by user feedback information (e.g., check-in) in a supervised way.

The main contributions of this paper are summarized below:

- We propose a spatial-aware user preference modeling method based on latent factor models to overcome the challenge of spatial dynamics of personal preferences. To alleviate data sparsity in inferring spatial-aware personal preferences, we leverage the spatial-aware crowd’s preferences in the form of social regularization and exploit the geographical autocorrelation of personal preferences based on their geographically hierarchical additive representations. Besides, we systematically illustrate how to design a matrix factorization objective function with social regularization and geographically hierarchical smoothing.
- To address the cold start and data sparsity issues, we explore multiple types of POI features and propose a supervised and task-guided method to learn the semantic representation of POIs from their associated heterogeneous features. To address the multi-modality issue, we proposed a late non-linear fusion method.
- We conduct extensive experiments on two datasets. Our experimental results show SH-CDL significantly outperforms the state-of-the-art recommendation methods, especially in out-of-town and cold-start scenarios.

Roadmap of this paper. Section 2 provides the preliminaries of our SH-CDL model. Section 3 details the model structure and inference. We present a hierarchically additive representation for spatial-aware personal preferences in Section 4. We extend the DBN to learn a unified POI representation from heterogeneous features in Section 5. We report our experimental study in Section 6. Section 7 reviews the related works and Section 8 closes the paper with some concluding remarks.

2 Preliminaries

In this section, we introduce the necessary preliminaries.

2.1 Probabilistic Matrix Factorization

POI recommendation is commonly studied on a user-POI matrix $C \in \mathbb{R}^{M \times N}$, where there are $M$ users and $N$ POIs. Each entry $c_{u,v}$ in $C$ records the feedback or the rating score of user $u$ to POI $v$. Hereafter, following the common symbolic notation, we will use upper case bold letters to denote matrices, lower case bold letters to denote column vectors and non-bold letters to represent scalars.

Given a check-in matrix $C$, one possible approach to POI recommendation is the collaborative filtering (CF). The state-of-the-art CF-based methods are probabilistic matrix factorization (PMF) [25], since PMF’s principled probabilistic interpretation enables it to easily incorporate auxiliary information such as POIs’ content and geographical information. PMF assumes that each user $u \in U$ and POI $v \in V$ can be represented as latent factor vectors $p_u$ and $q_v$, respectively. The rating or feedback that $u$ gives $v$ is the inner product of $p_u$ and $q_v$. The training data is usually very sparse, and without regularization the model is crippled by severe over-fitting. Therefore, Gaussian priors are used for both $p_u$ and $q_v$ as regularization. Formally, the probabilistic generative process of the model is listed as follows:

1. Sample $p_u \mid \rho_u \sim N(0, \rho_u^2 I)$
2. Sample $q_v \mid \rho_v \sim N(0, \rho_v^2 I)$
3. Sample $c_{u,v} \mid p_u, q_v, \rho_C \sim N(g(p_u^T q_v), \rho_C^2 I)$
where function \( g(x) \) is the Sigmoid function \( g(x) = 1 / (1 + \exp(-x)) \), which bounds \( P_i^T q_i \) within the range \([0, 1]\).

The negative log-likelihood of the PMF model is:

\[
L_{MF} = \sum_{u \in D} \sum_{v \in V} I_{u,v}(c_{u,v} - g(P_{u}^T q_{v}))^2 + \frac{\lambda_U}{2} \sum_{u \in D} \| p_u \|^2 + \frac{\lambda_V}{2} \sum_{v \in V} \| q_v \|^2, \tag{1}
\]

where \( I_{u,v} \) is a binary value indicating whether \( u \) visits \( v \), i.e., \( I_{u,v} = \| (c_{u,v} > 0) \). \( \lambda_U \) and \( \lambda_V \) are regularization parameters.

### 2.2 RBMs and Their Generalizations

Restricted Boltzmann Machines (RBMs) [14, 30] have been widely used in modeling distributions over binary-valued data. Recent work on Boltzmann machine models and their generalizations to exponential family distributions have allowed these models to be successfully used in many application domains. In particular, the Replicated Softmax model [14, 30] has been shown to be effective in modeling sparse word count vectors, whereas Gaussian RBMs have been used to model real-valued inputs for speech and vision tasks. In this subsection we briefly review these models following [30], as they serve as building blocks for our model.

#### Restricted Boltzmann Machines (RBM)

A RBM is a particular type of Markov random field that has a two-layer architecture. The visible binary stochastic units \( \mathbf{x} \in \{0, 1\}^D \) are connected to hidden binary stochastic units \( \mathbf{h} \in \{0, 1\}^F \), as shown in Figure 1 (a).

The energy of the state \( \{ \mathbf{x}, \mathbf{h} \} \) is:

\[
E(\mathbf{x}, \mathbf{h}; \Psi) = -\mathbf{x}^T \mathbf{W} \mathbf{h} - \mathbf{b}^T \mathbf{h} - \mathbf{x}^T \mathbf{a}^T \mathbf{h} = -\sum_{i=1}^D \sum_{j=1}^F W_{i,j} x_i h_j - \sum_{i=1}^D b_i x_i - \sum_{j=1}^F a_j h_j, \tag{2}
\]

where \( \Psi = \{ W, \mathbf{b}, \mathbf{a} \} \) is the model parameters; \( W_{i,j} \) represents the symmetric interaction term between visible unit \( i \) and hidden unit \( j \); \( b_i \) and \( a_j \) are bias terms. The joint distribution over the visible and hidden units is defined as:

\[
P(\mathbf{x}, \mathbf{h}; \Psi) = \frac{1}{Z(\Psi)} \exp(-E(\mathbf{x}, \mathbf{h}; \Psi)), \tag{3}
\]

where \( Z(\Psi) \) is known as the normalizing constant.

#### Gaussian RBM

Given the visible real-valued units \( \mathbf{x} \in \mathbb{R}^D \) and binary stochastic hidden units \( \mathbf{h} \in \{0, 1\}^F \), the energy of the state \( \{ \mathbf{x}, \mathbf{h} \} \) of the Gaussian RBM is defined as follows:

\[
E(\mathbf{x}, \mathbf{h}; \Psi) = \sum_{i=1}^D \frac{(x_i - h_i)^2}{2\rho_i^2} - \sum_{i=1}^D \sum_{j=1}^F W_{i,j} x_i h_j - \sum_{j=1}^F a_j h_j, \tag{4}
\]

where \( \Psi = \{ W, \mathbf{a}, \mathbf{b}, \mathbf{\rho}^2 \} \) are the model parameters.

#### Replicated Softmax Model

The Replicated Softmax Model is useful for modeling sparse count data, such as word count vectors in a document. Let \( \mathbf{x} \in \mathbb{N}^K \) be a vector of visible units where \( x_i \) is the number of times word \( i \) occurs in the document with the vocabulary of size \( K \). Let \( \mathbf{h} \in \{0, 1\}^F \) be binary stochastic hidden units. The energy of the state \( \{ \mathbf{x}, \mathbf{h} \} \) is defined as follows:

\[
E(\mathbf{x}, \mathbf{h}; \Psi) = -\sum_{i=1}^K \sum_{j=1}^F A_{i,j} x_i h_j - \sum_{i=1}^K b_i x_i - \sum_{j=1}^F a_j h_j, \tag{5}
\]

where \( \Psi = \{ W, \mathbf{a}, \mathbf{b} \} \) are the model parameters and \( \xi = \sum_i x_i \) is the total number of words in a document.

### 2.3 Deep Belief Network

RBMs can be stacked and trained in a greedy manner to form Deep Belief Networks (DBN) [13]. A deep belief network is a probabilistic generative graph model with many layers of hidden nodes at the top and one layer of observations at the bottom. DBN models the joint distribution between observed vector \( \mathbf{x} \) and the \( L \) hidden layers \( \mathbf{h}^m \) as follows:

\[
P(\mathbf{x}, \mathbf{h}^1, \ldots, \mathbf{h}^L; \Psi) = \prod_{m=0}^{L-2} P(\mathbf{h}^{m+1} | \mathbf{h}^m) P(\mathbf{h}^L; \mathbf{L}^{h^L-1}) \tag{6}
\]

where \( h^0 = \mathbf{x} \). \( P(\mathbf{h}^m | \mathbf{h}^{m+1}) \) is a conditional distribution for the visible units conditioned on the hidden units of the RBM at level \( m \), and \( P(\mathbf{h}^L; \mathbf{L}^{h^L-1}) \) is the visible-hidden joint distribution in the top-level RBM. This is illustrated in Figure 1 (b).

### 3 Geographical Deep Learning

In this section, we first formulate the POI recommendation problem, and then present our proposed SH-CDL model.

#### 3.1 Problem Definition

Below we first introduce some closely related concepts, and then define the POI Recommendation problem.

**Definition 1. (POI)** A POI is defined as a uniquely identified geographical site (e.g., a restaurant or a cinema).

In our work, we exploit four types of features for a POI: content features, geo-location features, popularity features and neighborhood features (with the details discussed in Section 5.1). Given a POI \( v \), we use \( \mathbf{x}_v^c \) to denote the content features of \( v \) (e.g., its category, description and comments), \( \mathbf{x}_v^g \) to denote its geo-location features (e.g., its geographical position), \( \mathbf{x}_v^p \) to denote the popularity features of \( v \) (e.g., its check-in number), and \( \mathbf{x}_v^n \) to represent the neighborhood related features. \( \mathbf{x}_v \) is used to represent the whole feature vector of POI \( v \), and \( \mathbf{X} \) denotes the feature matrix formed by all \( \mathbf{x}_v \).

**Definition 2. (User Home Location)** Following the recent work of [16], given a user \( u \), we define the user’s home location as the place where the user lives, denoted as \( l_u \).

Due to privacy concerns, users’ home locations are not always available. For a user \( u \) whose home location is not explicitly given, we adopt the method developed by [29] to infer her home location based on her check-in history.

**Definition 3. (Check-in)** A check-in is a triplet \((u, v, s)\) that means user \( u \) visits POI \( v \) with the role of \( s \). If \( s = 1 \), the user is recognized as a local and the check-in activity occurs in \( u \)’s home town. If \( s = 0 \), \( u \) is a tourist when visiting \( v \).

**Definition 4. (Check-in Cuboid)** A check-in cuboid \( C \) is a \( M \times N \times 2 \) cuboid, where \( M \) is the number of users, \( N \) is the number of POIs, and 2 is the number of roles. A cell indexed by \((u, v, s)\) (i.e., \( c_{u,v,s} \)) stores the feedback/rating of \( u \) to \( v \) with the role \( s \). Without loss of generality, \( c_{u,v,s} \) is normalized
to the range $[0,1]$ using the function $f(x) = x/C_{\text{max}}$ where $C_{\text{max}}$ is the maximum possible value.

Given a check-in Cuboid $C$ and a POI feature matrix $X$, we aim to provide POI recommendation for both home-town and out-of-town users. Following [45], we formulate our problem below.

**Problem 1. (POI Recommendation)** Given a querying user $u_q$ with her target region $r_q$, we first compute the user role $s_q$ according to the distance between her target region and her home location. Then, a query $q = (u_q, r_q, s_q)$ is formed. Our goal is to find top-$k$ POIs from $\mathcal{V}$ that match the preferences of query $q$. The problem becomes an out-of-town recommendation if the distance is greater than a predefined threshold $\zeta$. Otherwise, the problem is a home-town recommendation.

Following the related studies [10], [26], we set $\zeta = 100 km$.

### 3.2 Model Structure of SH-CDL

The graphical representation of SH-CDL is shown in Figure 2, where different colors represent different components.

To capture the spatial dynamic of personal preferences, we assume user’s preferences are region-dependent in SH-CDL and use $p_{u,r}$ to denote $u$’s preferences when traveling to region $r$. To overcome the cold-start problem, SH-CDL incorporates versatile auxiliary information of POIs (i.e., POI features). Given a POI $v$ at region $r$, we use $x_v$ to denote its raw feature vector directly extracted from the auxiliary information, and $DBN(x_v; \Psi)$ to denote its hidden feature vector modeled from raw features by DBN [13]. We use $q_v$ to denote its latent factor vector, and assume that the dimension of $q_v$ is the same as that of the hidden feature vector. Similar to the probabilistic matrix factorization (PMF), the probabilistic generative process of SH-CDL for each check-in record $c_{u,v,s}$ is formulated as follows:

\[
\begin{align*}
    c_{u,v,s} &\sim \mathcal{N}(g(p^u_{u,v}, p^r_{r,v}, \theta_{u,r,v}, \hat{q}_v^u, I)), \\
    p_{u,r} &\sim \mathcal{N}(\theta_{u,r,v}, \hat{p}^u_{r,v}I), \\
    \theta_{u,r} &\sim \mathcal{N}(\hat{\theta}_{u,r,v}, \hat{p}^r_{r,v}I), \\
    q_v &\sim \mathcal{N}(DBN(x_v; \Psi), \hat{\rho}^r_{r,v}I)
\end{align*}
\]

where we assume that $c_{u,v,s}$ follows a normal distribution to account for the noise in the user ratings/feedbacks. In SH-CDL, $\theta_{u,r}$ denotes the collective preferences of the general public visiting region $r$ with role $s$ (i.e., the wisdom of the crowd). Since the user’s check-in records generated in a single region $r$ are sparse, especially when $r$ is an out-of-town region, it is very challenging to infer the user’s preferences in the region without over-fitting. Therefore, we exploit the wisdom of the crowd. Following [33], [45], SH-CDL recognizes two roles of an individual user in a geographical region: a local or a tourist. Given a region, check-in records from local users will be mined to learn native preferences $(s = 1)$. Similarly, check-in records from tourists will be used to learn tourist preferences $(s = 0)$. Users with the same role at a region are more likely to have similar preferences. Taking the collective preferences $\theta_{s,r}$ as the Gaussian prior for region-aware personal preferences $p_{u,r}$ alleviates the issue of data sparsity to some extent. In practice, the collective preference $\theta_{s,r}$ plays the role of social regularization (i.e., it serves as a regularization term in the objective function in Equation 8). $\rho_U$ indicates the variance of personal preferences. The smaller the $\rho_U$, the more similar the personal preferences $p_{u,r}$ are to the collective preferences $\theta_{s,r}$ at region $r$, and the more strongly $p_{u,r}$ is regularized.

Automatic learning of hidden feature vector from POI auxiliary information $x_v$ is achieved by Deep Belief Network (DBN) [13], considering its excellent performance in recent applications and its robustness to noise [9], [30], [37]. DBNs are graphical models which learn to extract a deep hierarchical representation of the input features, and can be treated as a very flexible deterministic function that maps $x_v$ to $DBN(x_v; \Psi)$. To overcome the cold-start problem, our SH-CDL unifies the probabilistic matrix factorization (PMF) and the deep representation learning by linking the latent factor vector $q_v$ and the hidden feature vector $DBN(x_v; \Psi)$. Rather than simply defining the two vectors to be equal as in [31], [37], we assume that $q_v$, follows a normal distribution with the mean $DBN(x_v; \Psi)$, to learn a more robust mapping from the POI features, since users’ check-in behaviors may be influenced by other unknown factors beyond our collected POI features.

Based on the probabilistic generative process of SH-CDL, through a simple Bayesian inference, the posterior distribution of the personal preferences, the collective preferences of crowds and the POIs’ latent factor vectors is formulated in Equation (7), where $P, \Theta, Q$ are two hyper-matrices and one matrix formed by $p_{u,r}, \theta_{s,r}$ and $q_v$, respectively. Both $I_{u,v,s}$ and $I_{u,r,s}$ are indicator functions and defined as: $I_{u,v,s} = \mathbb{1}(c_{u,v,s} > 0)$, and $I_{u,r,s} = \mathbb{1}(u \in r) \odot s$ where $\odot$ is the XOR operation. If $u$’s home location $I_u$ is within region $r$, we assume that $u$ is local at $r$. $I_{u,v,s}$ indicates whether $u$ has visited a POI $v$ with the role $s$.

### 3.3 Model Inference of SH-CDL

We adopt the Maximum Likelihood Estimation to infer the model. The negative log-likelihood of Equation (7) is shown in Equation (8), where irrelevant constants are omitted. The hyper-parameters $\lambda_U$, $\lambda_R$ and $\lambda_V$ are the ratios $\hat{p}^u_{r,v}/\hat{p}^r_{r,v}$ and $\hat{p}^u_{r,v}/\hat{p}^u_{r,v}$, respectively, and a larger value indicates a stronger regularization. $r_v$ denotes the region where POI $v$ is located. The model parameters consist of two parts: the weight parameters $\Psi$ in DBN and the latent vectors $p_{u,r}, \theta_{s,r}$ and $q_v$. Below, we introduce how to estimate them using the gradient descent method.

Since $\Psi$ consists of a large amount of parameters, directly optimizing $L_{SH-CDL}$ using gradient descent could easily lead to overfitting. Following the DBN training process established in [13], we first pre-train the DBN as stacked layers of Restricted Boltzmann Machines in an unsupervised fashion, i.e., the principle of greedy layer-wise unsupervised training is applied to DBNs with RBMs as the building blocks for each layer. We then optimize $L_{SH-CDL}$ using gradient descent, and the gradient descent part of DBN is implemented as back-propagation. The whole DBN training process is shown as follows:

1) Train the first layer of DBN as an RBM that models the raw input $\tilde{h}^0 = \tilde{x}_v$ as its visible layer, using CD1 algorithm [13].
The computation complexities for updating gradients in Equation (7) follows (a special case of the chain-rule of derivation): entries in the check-in cuboid \( C \) gradients against variables. Because of the sparsity of cuboid in addition to the DBN, the main computation in the model where \( g \) is the number of parameters in the DBN. We implemented our SH-CDL using Theano 1, because it supports convenient GPU programming and automatic symbolic differentiation. GPUs with large video memory enable the training of our SH-CDL to be efficient.

### 3.5 POI Recommendation Using SH-CDL

Once we have trained the model SH-CDL, given a querying user \( u_q \) with the target region \( r_q \), we first compute the indicator \( s_q \) (i.e., the role of user \( u_q \)). Thus, a query \( q = (u_q, r_q, s_q) \) is formed. Then, we compute the ranking score of each unvisited POI \( v \) in region \( r_q \) w.r.t. the query \( q \), as in Equation (13), and then select the top-\( k \) with highest ranking scores as recommendations.

\[
S(q, v) = g(p_{u_q, r_q, v}^T)
\]

As the latent factor vector \( q_v \) is approximate to the hidden feature vector \( DBN(x_v, \Psi) \), the ranking score for the cold-start POIs (e.g., new POIs) can be approximately computed as follows:

\[
S(q, v) = g(\theta_{s_q, r_q}^T DBN(x_v, \Psi)).
\]

For cold-start users, SH-CDL can make accurate recommendation according to the collective preferences of the crowd with the same role \( s_q \) at target region \( r_q \), as follows:

\[
S(q, v) = g(\theta_{s_q, r_q}^T).
\]

### 4 HIERARCHICAL USER PREFERENCE MODELING

To capture the spatial dynamics of personal preferences, we assume that personal preferences are spatial-aware in our SH-CDL model. However, the user’s check-in records at a single region \( r \) are sparse, especially when \( r \) is an out-of-town region. To alleviate the data sparsity, we exploit the geographical autocorrelation of personal preferences. Given close regions have similar urban compositions and cultures, personal preferences in these regions should be similar. Thus, we can “borrow” the user’s check-in records from nearby regions to smooth her preferences at \( r \).

To encode the spatial autocorrelation and enable the learned personal preferences to be smoothed over spatial regions, we adopt a hierarchical spatial index structure - spatial pyramid, to partition and index the entire geographic area, inspired by [34]. The spatial pyramid is constructed by partitioning POI locations into spatial regions of varying sizes at different levels. More specifically, the spatial pyramid decomposes the space into \( h \) levels. The first/top level has only one grid cell. For a given level \( j \geq 1 \), the space is partitioned into \( 4^{j-1} \) equal area grid cells. Thus, the space is divided recursively into finer cells at deeper levels. An example of a spatial pyramid with \( h = 4 \) is illustrated in Figure 3.

Based on the spatial pyramid structure, we propose a novel path-based region representation approach. That is, each region \( r \) can be represented by a path from the root cell to its corresponding

1. http://deeplearning.net/software/theano/
cell. We use a vector to describe the path. For example, given a region \( r_h \) which corresponds to a cell at level \( h \), the region can be represented by the vector \((r_1, r_2, \ldots, r_j, \ldots, r_h)\) where cell \( r_j \) is the father node of cell \( r_{j+1} \). Based on the path-based region representation, we propose a hierarchically additive framework [34] to represent personal preferences, as follows:

\[
p_{u, r_h} = \sum_{j=1}^{h} \phi_{u, r_j}
\]  

Similarly, the collective preferences of the general public with role \( s \) at region \( r \) can be represented as follows:

\[
\theta_{s, r_h} = \sum_{j=1}^{h} \theta_{s, r_j}
\]

where region \( r_h \) is assumed to correspond to a cell at level \( h \); if \( r_j \) is an ancestor node of \( r_h \), that means region \( r_h \) belongs to region \( r_j \), denoted as \( r_h \in r_j \). \( \phi_{u, r_j} \) and \( \theta_{s, r_j} \) represent the amortized personal preferences and collective preferences at level \( j \), respectively. Since \( p_{u, r_h} \) and \( \theta_{s, r_h} \) are respectively represented as the sum of \( \phi_{u, r_j} \) and \( \theta_{s, r_j} \) at each level \( j \), we need to calculate the gradients of \( \phi_{u, r_j} \) and \( \theta_{s, r_j} \). Thus, we extend Equations (10), (11) to infer the gradients of \( \phi_{u, r_j} \) and \( \theta_{s, r_j} \) at each level \( j \), as shown in Equations (18), (19) where \( p_{u, r_h} \) and \( \theta_{s, r_h} \) are computed as in Equations (16), (17).

Based on the hierarchically additive representation for spatial-aware personal preferences, if there are few or no check-in records for a user \( u \) at region \( r \), we can still infer \( u \)'s personal preferences guided by her check-in data generated at \( r \)'s ancestor regions.

5 Exploiting Heterogeneous POI Features

In this section, we first present the four types of heterogeneous features extracted for POIs, and then develop a multimodal learning technique to fuse these features.

5.1 Feature Extraction

In this subsection, we introduce four types of features we have extracted for POIs.

Content Features. Given a POI \( v \), we first extract textual content features \( x^c_v \) from POIs’ categories, description and comments. The formation of the feature vector and its size are discussed in the experiments in Section 6.

Geo-Location Features. Given a POI, its geo-location refers to its geographical position in terms of latitude and longitude.

Popularity Features. As analyzed in [46], the probability of a user visiting a POI is largely affected by local word-of-mouth about the POI (i.e., the popularity of the POI), especially when users travel in unfamiliar regions. In our SH-CDL model, we exploit three kinds of popularity features: overall popularity, role-aware popularity and temporal popularity. Given a POI, the overall popularity features refer to the total number of check-ins and the total number of unique visitors. According to user roles, the overall popularity can be refined into its popularity among local people and tourists, respectively. As different users have different working schedules and life styles, the compatibility between time-varying popularity of POIs and regular availability of users can influence user decision-making [39]. Thus, we further consider the temporal distributions of its check-ins and its unique users over the course of a week (i.e., Monday to Sunday) and in a 24-hour scale as well as on weekdays and weekends, respectively. In total, there are 66 temporal popularity features that consist of \( 7 \times 2 \) weekly popularity features, \( 24 \times 2 \) daily popularity features and \( 2 \times 2 \) weekday-weekend popularity features. In a short, we generate 2 overall popularity features, 4 role-aware popularity features and 66 temporal popularity features for each POI.

Neighborhood Features. Given a POI \( v \), the neighborhood features refer to the ones that describe the environment around \( v \). The geographical influence [40] suggests that people tend to explore the POIs near the ones they favor. More specifically, we measure the density, heterogeneity, competitiveness and popularity of the surrounding area, the transition density and transition quality, and the transportation by analyzing the set \( \{v' \in V : \text{dist}(v, v') < d\} \) of POIs that lie in a disk of radius \( d \) around POI \( v \). The function \( \text{dist} \) denotes the geographical distance between two POIs, and \( V \) is the whole set of POIs.

1) Density: The density feature refers to the number of neighbors around the POI within a circle of radius \( d \). Formally, it is defined as follows:

\[
F_{n1}(v) = \left| \{v' \in V : \text{dist}(v, v') < d\} \right|
\]

Given that the radius \( d \) is the same for all POIs, the density of neighboring POIs depends only on the number of POIs in the neighborhood. We use the notation \( N(v, d) \) to denote the number of \( v \)'s neighbors within a radius \( d \). Intuitively, a denser area could imply a higher likelihood of an opportunistic visit to a target POI.

2) Heterogeneity: To incorporate the influence of the spatial heterogeneity of the neighborhood on the users’ check-in behaviors, we apply an entropy measure from information theory to the frequency of the POI types (i.e., categories) in the area. We denote the number of POI neighbors of type \( \gamma \) with \( N_v(v, d) \). The neighborhood entropy is formally defined as follows:

\[
F_{n2}(v) = - \sum_{\gamma \in \Gamma} \frac{N_v(v, d)}{N(v, d)} \times \log \frac{N_v(v, d)}{N(v, d)}
\]

where \( \Gamma \) is the set of all POI types.

3) Competitiveness: We devise a feature to describe the competitiveness of the surrounding area. Given the type of POI \( v \) (e.g., Chinese Restaurant), we measure the proportion of neighboring POIs of the same type \( \gamma_v \) as follows:

\[
F_{n3}(v) = \frac{N_{\gamma_v}(v, d)}{N(v, d)}
\]

Competition can have either positive or negative effects. One would expect that, for instance, a bar in an area populated with nightlife spots would be more attractive to users as there is already an ecosystem of related services. However, being surrounded by competitors may also mean that users have more choices, and the probability of visiting \( v \) might be reduced.

4) Area Popularity: To assess the influence of the overall popularity of the area, we measure the total number of check-ins, empirically observed among the neighboring POIs in the area:

Fig. 3. The Spatial Pyramid with Four Layers.
\[
\frac{\partial L_{SH-CDL}}{\partial \phi_{u,r_j}} = 2 \sum_{r_h \in r_j} \sum_{v \in V_{x \in \mathcal{U}}} \sum_{u,v \in V_{x \in \mathcal{U}}} I_{u,v,s} g'(p_{u,r_h} q_v) g(q_{p_{u,r_h} q_v} - c_{u,v,s}) + \lambda_U \sum_{r_h \in r_j} \sum_{u \in \mathcal{U}} I_{u,r_h,s} (q_{p_{u,r_h} - \theta_{r_h}}),
\]
\[
\frac{\partial L_{SH-CDL}}{\partial \theta_{x,r_j}} = \lambda_U \sum_{r_h \in r_j} \sum_{u \in \mathcal{U}} I_{u,r_h,s} (q_{p_{u,r_h} - \theta_{r_h}}) + \lambda_B \phi_{x,r_j},
\]

\(F_n(v) = |\{(u, v', s) \in C : \text{dist}(v, v') < d\}|.
\]

5) Transition Density: Assuming that increased mobility from neighboring POIs to \(v\) can increase the probability of a user visiting \(v\), we measure the density of transitions between \(v\) and its neighboring POIs as follows:

\(F_n(v) = |\{(v', v) \in T : \text{dist}(v', v) < d\}|
\]

where \(T_v\) is the set of consecutive check-in transitions from other POIs \(v' \in V\) to \(v\).

6) Potential Transition Density: Another aspect of POI attractiveness comes from the potential number of local users that the POI might attract from the area. We thus measure the probability of transitions from POIs of all other types to POIs of type \(\gamma_v\). The resulting probabilities allow us to take into account the nearby POIs as potential sources of users to the POI \(v\). More formally:

\(F_n(v) = \sum_{\{v' \in V : \text{dist}(v', v) < d\}} \kappa_{\gamma_{v'} \rightarrow \gamma_v} \times N_{v'}
\]

where \(N_{v'}\) is the number of check-ins at POI \(v'\), and \(\kappa_{\gamma_{v'} \rightarrow \gamma_v}\) is a probability of transitions between categories \(\gamma_v\) and \(\gamma_{v'}\), defined as the average percentage of all check-ins to POI \(v'\) that are followed by POIs with type \(\gamma_v\):

\(\kappa_{\gamma_{v'} \rightarrow \gamma_v} = \frac{E[\{(m, n) \in T : m = v' \wedge \gamma_m = \gamma_v\}]}{N_{v'}}
\]

where \(T\) is the total set of consecutive check-in transitions between POIs and as a tuple, \((m, n) \in T\), the POIs \(m \in V\) and \(n \in V\) involved in two consecutive check-ins.

7) Transportation: The transportation features refer to the numbers of bus stops and subway stations in the surrounding area, which are defined as follows:

\(F_n(\gamma_{bus}) = |\{v' \in V : \text{dist}(v', v) < d \wedge \gamma_{v'} = \text{bus stop}\}|
\)

\(F_n(\gamma_{subway}) = |\{v' \in V : \text{dist}(v', v) < d \wedge \gamma_{v'} = \text{subway station}\}|
\]

As for the radius \(d\) in the neighborhood features, we have experimented with different values and have selected it to be equal to 200 meters because this yielded the best experimental results, and is also consistent with the optimal neighborhood size in [23].

5.2 Multimodal Learning

The POI features described above are heterogeneous and multimodal. Specifically, the content features are represented as sparse word count vectors, whereas the geo-location, popularity and neighborhood features are real-valued and dense. There is a great deal of structure in the POI features but it is difficult to discover highly non-linear relationships that exist between features across different modalities. Moreover, these features are noisy and may have missing values. Traditional DBNs adopt an early fusion strategy to directly concatenate all types of features as the joint input for the first layer RBM, as shown in the left panel of Figure 4. The joint input is then abstracted by other high-level RBMs. Although the classic early fusion mechanism achieves good performance with homogenous features, it does not apply to multimodal heterogeneous features because the correlations across different modalities are highly non-linear, and it is difficult for the first RBM to represent their relations properly.

In light of this, we extend the traditional DBN to a multimodal DBN (MDBN) and adopt a late fusion strategy, inspired by [30], [51]. As shown in the right panel of Figure 4, the key idea of the MDBN is to first use separate modality-friendly latent variable models to learn latent representations of each data modality independently. Then, the latent representations (instead of the raw features) for different modalities are concatenated to form a multimodal input. Higher-order latent variables can subsequently be used to model the distribution over this input. Similar to the DBN, the parameters of this MDBN can be learned approximately by greedy layer-wise training.

Note that the MDBN can be described as a composition of multiple unimodal pathways. Each pathway is learned separately in a completely unsupervised fashion, which enables the MDBN to be more flexible and powerful than the conventional DBN with the early fusion strategy. Any number of pathways each with any number of layers could potentially be used. The type of lower RBMs in each layer could be different, accounting for different kinds of input distributions, as long as the final hidden representations at the end of each pathway are of the same type. Since input features for our SH-CDL model consists of both dense real-value features and sparse word count features, we use the Gaussian RBM to model the distribution over real-valued features (i.e., its geo-location features, popularity and neighborhood features) while a Replicated Softmax model is employed to model its distribution over the word count vectors (i.e., the content features). As for the rest of the layers, binary RBMs are adopted [12].

Zhang et al. [49] also proposed a late fusion strategy for the multi-modal features, but they assumed that the learnt hidden features of different modalities are in the same semantic space and simply add them together to represent the latent factors of an item. In our fusion approach, the learnt hidden features of different modalities are in different spaces. We first project them to the same semantic space by multiplying a modality-specific matrix and then adopt a non-linear method to fuse them.

6 Experiments

In this section, we first describe the settings for the experiments and then demonstrate the experimental results.

6.1 Datasets

Our experiments are conducted on two large real-life datasets:

Yelp. Yelp’s Challenge Dataset ² contains 552,000 users and 77,000 POIs from 10 cities across 4 countries. There are 2.2

². https://www.yelp.com/dataset_challenge
millions of rating records, and each rating record consists of a user-ID, a POI-ID, a rating score (1-5 star) and a time-stamp. For each POI, we also collected its four types of features.

**Foursquare**. Foursquare is one of the most popular LBSNs. We collected its public check-in data from Sep 2010 to Jan 2011 through Twitter with the same crawling strategy as proposed in [4]. This dataset contains 62,462 POIs and 1,434,668 check-ins generated by 114,508 users who live in the USA. Each check-in is stored as user-ID, POI-ID, time-stamp. For each POI, we crawled its auxiliary information through Foursquare’s API and extracted four types of features. As there is no explicit rating information in this dataset, we assigned a rating of 1 to each observed check-in and used them as the positive samples. To generate negative samples, we used the well-established User-Oriented Sampling method proposed in [27], i.e., for user $u$ who visits $V_u$, we randomly sampled $|V_u|$ POIs from $V \setminus V_u$ and assigned a rating of 0 to each generated negative check-in.

### 6.2 Comparative Approaches

We compare our SH-CDL with the following four deep learning-based recommenders and two state-of-the-art POI recommenders. Table 1 lists the characteristics of these methods. For fair comparison, all deep learning based models take the same input features.

**DL-PMF**. DL-PMF is a two-stage deep learning powered probabilistic matrix factorization model [31]. This model first uses PMF to learn latent factor vectors $p_u$ and $q_v$ for all users and items, and then trains a DBN to map from the item feature vector $x_v$ to the latent factor vector $q_v$.

**HLDBN**. The hierarchical linear model combined with DBN was developed for music recommendation in [37]. It unifies the deep belief network and probabilistic matrix factorization model. They try to minimize the following objective function:

$$L_{HLDBN} = \sum_{u \in U} \sum_{v \in V} (c_{u,v} - p_u^T DBN(x_v, \psi))^2 + \lambda \sum_{u \in U} \|p_u - \mu\|^2$$

where $\mu$ represents the common preferences of all users.

**CDL**. Collaborative Deep Learning [33] unifies the stacked denoising auto-encoders (SDA) and matrix factorization model. In CDL, the denoising auto-encoders is used to learn the hidden feature vectors from the auxiliary information of items.

**DCF**. Deep Collaborative Filtering model was developed in [17], which unifies the marginalized denoising auto-encoders (mDA) and matrix factorization model following CDL.

**ST-LDA**. ST-LDA [45] is the state-of-the-art POI recommender model that can adapt to the spatial dynamics of user interests. It extends the topic model LDA to model region-aware personal interests by a topic distribution.

**ST-GLDA**. Inspired by the success of the recent word2vector techniques, we further extend ST-LDA to generate continuous word embeddings and use a multivariate Gaussian distribution to represent a topic following the Gaussian LDA model [8].

Note that we do not directly compare our SH-CDL with other powerful POI recommendation methods such as GT-BNMF [21], Geo-SAGE [34], Rank-GeoFM [18] and LTSCR [50], because ST-LDA has been validated to outperform these methods in the same experimental settings and datasets in [45].

To further validate the benefits brought by each new technology developed in our SH-CDL, including spatial-aware personal preference modeling (T1), leveraging collective preferences to perform social regularization (T2), exploiting geographical correlation to smooth region-aware personal preferences (T3) and the late feature fusion strategy developed in MDBN (T4), we design four variant versions of SH-CDL. **SH-CDL-T1** is the first variant version of SH-CDL where personal preferences are region-independent, i.e., $p_u$ is stable over different regions. Correspondingly, the social regularization and spatial smoothing techniques are not applicable to SH-CDL-T1. **SH-CDL-T2** does not consider the collective preferences, and simply adopts zero as the Gaussian prior for the personal preference vector $p_u$. **SH-CDL-T3** does not exploit the geographical correlation to smooth personal preferences in the spatial pyramid. **SH-CDL-T4** directly employs the conventional DBN with the early fusion strategy to model the heterogeneous features.

### 6.3 Implementation Notes

Training deep learning models (e.g., DBN, SDA and mDA) on CPUs is extremely slow. GPUs with large memory are indispensable in deep learning experiments. Training the SH-CDL model on our two datasets takes two to three days using a single GPU with 4GB video memory. Hence, we developed a scalable parallel implementation of the SH-CDL by harnessing the powers of GPUs and clusters. Specifically, we deployed our SH-CDL model on an Amazon EC2 Multi-GPU cluster with 10 computing nodes (g2.2xlarge3), each equipped with four high-performance NVIDIA GPUs. Each has 1536 CUDA cores and 4GB video memory. Stochastic gradient descent (SGD) is a popular technique for large-scale optimization problems and has been widely adopted for the inference of deep learning models. However, traditional SGD processes one example per iteration, and this sequential nature makes SGD challenging for distributed inference. To parallelize the inference of our SH-CDL, we employed the efficient minibatch training which processes multiple examples at each iteration.

There are more than 20,000 distinct non-stop words in our datasets. To keep the text representation manageable, the content features are represented using a vocabulary of the 2000 most frequent words. Even so, $x_v^c$ is still a sparse word count vector (i.e., the content features), and we employed a Replicate Softmax Model with 2000 visible and 200 hidden units followed by another layer of 50 hidden units. For the dense real-value features (i.e., the concatenation of the geo-location, popularity and neighborhood

<table>
<thead>
<tr>
<th>Features</th>
<th>Spatial-aware User Preferences</th>
<th>Spatial-Aware Crowd’s Preferences</th>
<th>Supervised Hidden Feature Extraction</th>
<th>Unsupervised Hidden Feature Extraction</th>
<th>Late Feature Fusion</th>
<th>Early Feature Fusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH-CDL</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>DL-PMF</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>HLDBN</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>CDL</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>DCF</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>ST-LDA</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>ST-GLDA</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
</tbody>
</table>

*Table 1: Features of Different Methods.*
features), we employed a Gaussian RBM with 80 visible units followed by 30 hidden units. The joint layer for the two pathways contains 80 hidden units for the POI representation. Obviously, there are four layers in the pathway towards content features, while there are only three layers in the pathway towards the real-value features, because the predicative abilities of content features and the other three types of features are not in the same level. The three types of real-value features are closer to the learned latent features in the joint layer, since they are more powerful than the content features in predicting user check-in behaviors, as will be analyzed in Section 6.6. We actually tried different configurations of hyperparameters such as the number of layers and the number of units for each layer by referring to [12]. With the above configuration, SH-CDL achieved the best recommendation performance.

The pre-training was conducted for each pathway of the MDBN with the greedy layer-wise CD1 algorithm for 200 iterations, and then all the parameters were fine-tuned via the supervised gradient descent of the negative log-likelihood $L_{SH-CDL}$. The mini-batch size for both pre-training and fine-tuning was 5000. The learning rate for the Gaussian RBM was $5 \times 10^{-5}$ and other RBMs $10^{-2}$. The fine-tune learning rate at the supervised training stage was 0.5, and the regularization parameters $\lambda_v=100$, $\lambda_t=0.01$. We use the cross-validation method to obtain the above configuration of hyperparameters with which our model achieved the optimal performance.

### 6.4 Evaluation methods

Our SH-CDL is designed to support both home-town and out-of-town recommendation, we evaluate the recommendation effectiveness of our model in both scenarios. Given a collection of check-in records $C_u$ generated by each user $u$, we first divide $C_u$ into home-town check-ins $C_u^{home}$ and out-of-town check-ins $C_u^{out}$. To decide whether a check-in occurred in the home town or out of town, we measure the distance between the user’s home location and the POI (i.e., $|l_u - l_v|$). If the distance is greater than $\zeta$, then we assume the check-in occurred out of town. Otherwise, it occurred in the hometown. The check-in records in both $C_u^{home}$ and $C_u^{out}$ are first ranked according to their check-in timestamps. Then, we use the 80-th percentile as the cut-off point so that check-ins before this point are used for training, and the rest are for testing. In the training dataset, we choose the last 10% of the check-ins as the validation data to tune the model’s hyperparameters such as the numbers of layers and the number of units in each layer. Note that $C_u^{home}$ (or $C_u^{out}$) is also included in the training set for the evaluation of the out-of-town recommendation (or the home-town recommendation).

According to the above dividing strategies, we split the dataset $C$ into the training set $C_{training}$ and the test set $C_{test}$. To evaluate the recommendation methods in top-$k$ recommendation, we use the Accuracy@$k$ measurement and evaluation methodology which has been widely adopted by [7], [34], [44]. Specifically, for each non-zero entry $c_{u,v,s} \in C_{test}$:

1. We compute the ranking score for the ground-truth POI $v$ as well as all others within the circle with center $v$ and radius $\zeta$ and unvisited by $u$, instead of all POIs, since only those being geographically close to $v$ are real competitors.
2. We form a ranked list by ordering these POIs according to their scores. Let $p$ denote the position of POI $v$ in the ranking list.
3. We form a top-$k$ recommendation list by picking the $k$ top ranked POIs from the list. If $p \leq k$, we have a hit (i.e., the ground truth $v$ is recommended to the user). Otherwise, we have a miss.

The computation of Accuracy@$k$ is as follows.

$$\text{Accuracy}@k = \frac{\text{#hit}@k}{|C_{test}|}$$

where $\text{#hit}@k$ denotes the number of hits in the test set, and $|C_{test}|$ is the total number of all test cases.

For models that can predict ratings, another evaluation metric - Mean Absolute Error (MAE) is also used to measure the deviation of the predicted ratings from their true values on the Yelp dataset. For each non-zero entry $c_{u,v,s} \in C_{test}$ and a predicted rating $\hat{c}_{u,v,s}$, the absolute difference $|c_{u,v,s} - \hat{c}_{u,v,s}|$ is calculated between them as the error for that pair. Then $MAE$ is computed as the average error over all test cases.

$$MAE = \frac{1}{|C_{test}|} \sum_{c_{u,v,s} \in C_{test}} |c_{u,v,s} - \hat{c}_{u,v,s}|$$

### 6.5 Recommendation Effectiveness

In this section, we report the comparison results between our proposed SH-CDL and other competitor methods with well-tuned parameters in Figures 5 and 6. All differences between our model and the other comparison methods are statistically significant ($p < 0.01$). We only show the performance where $k$ is set to 1, 5, 10, 15, 20, as a greater value of $k$ is usually ignored.

**Out-of-Town Recommendation On Foursquare.** Figure 5(a) presents the recommendation accuracy in the out-of-town recommendation scenario, where the accuracy of SH-CDL is about 0.194 when $k = 10$ (i.e., the model has a probability of 19.4% of placing an appealing POI in the top-10). Clearly, our proposed SH-CDL significantly outperforms other models, and the relative improvements, in terms of Accuracy@10, are 20.13%, 24.61%, 115.06%, 119.75%, 130.42% and 205.54% compared with ST-GLDA, ST-LDA, DCF, CDL, HLDBN and DL-PMF, respectively.

Several observations are made from the results. First, SH-CDL achieves much higher recommendation accuracy than the state-of-the-art models ST-LDA and ST-GLDA. This demonstrates the effectiveness of SH-CDL by extracting and incorporating multiple types of POI features in a supervised and deep manner, while ST-LDA and ST-GLDA only integrate the textual and temporal information in an unsupervised and shallow way. Second, ST-GLDA performs slightly better than ST-LDA, showing the small benefit brought by the word embedding technique. Third, although all SH-CDL, DCF, CDL, HLDBN and DL-PMF are deep learning-based recommender models that integrate the same POI features, our SH-CDL outperforms others significantly due to the new techniques developed in the SH-CDL, including spatial-aware personal preference modeling, leveraging the collective preferences to perform social regularization, exploiting the geographical correlation to smooth personal preferences and multimodal learning from heterogeneous features. SH-CDL also learns a more robust POI representation from the raw features than DCF, CDL and HLDBN, because: 1) The DBN adopted by our SH-CDL is more robust to noise than the denoising auto-encoders [9] used by CDL and DCF; and 2) HLDBN directly takes the learnt latent feature vectors as the latent factors in the matrix factorization, while our SH-CDL introduces the Gaussian noise. Fourth, among all the deep learning-based recommender models, DL-PMF lags far behind others. This is because DL-PMF adopts a pipeline approach to perform matrix factorization and hidden feature extraction in sequence and fails to capture their interaction and interdependence.
Table 2: Rating Prediction on the Yelp Dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Out-of-Town Scenario</th>
<th>Home-Town Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ac@1</td>
<td>Ac@5</td>
</tr>
<tr>
<td>SH-CDL</td>
<td>0.976</td>
<td>1.192</td>
</tr>
<tr>
<td>ST-GLDA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ST-LDA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HLDBN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DL-PMF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
they can be ranked as follows: $F_4 > F_3 > F_1 > F_2$, while in the home-town recommendation scenario they can be ranked as $F_4 > F_2 > F_3 > F_1$.

### 6.7 Parameter Sensitivity Analysis

In this experiment, we study the impact of the spatial pyramid’s height and the regularization parameter for personal preference vectors on the Foursquare dataset.

#### Study of The Impact of Spatial Pyramid Height

The height of the spatial pyramid is an important parameter, since it determines the space partition granularity (i.e., the region size). To study its impact, we test the performance of the SH-CDL model by varying the height of spatial pyramid from 1 to 7. The results are presented in Table 8. When the height is set to 1, it means that there is no space partition, and our SH-CDL degrades to SH-CDL-T1 where personal preferences are assumed to be stable. From the results, we observe that the recommendation accuracy of SH-CDL first increases with increasing height, and then begins to slightly decrease when the height is larger than 5. One possible reason for the early increase is that increasing the height increases the exploitation of spatial dynamic of personal preferences and makes the region-aware personal preferences more precise. Later on, the recommendation accuracy begins to slightly decrease as the height continues to increase, because increasing the height makes personal check-in data in a region cell sparser and also means there are more parameters to be learnt. SH-CDL achieves its best performance when the height of spatial pyramid is 5, which could be a tradeoff between the aforementioned two factors. The size of a cell in level 5 is about $100km \times 100km$. Another observation is that the performance of SH-CDL is more sensitive to the height of the spatial pyramid in the out-of-town recommendation scenario than in the home-town scenario.

#### Study of The Impact of Regularization Parameter $\lambda_U$

$\lambda_U$ is another important hyper-parameter in SH-CDL. The larger the $\lambda_U$, the more similar the personal preferences $p_{u,r}$ are to the collective preferences of the crowd with the same role $\theta_{c,r}$, and the more strongly $p_{u,r}$ is regularized. When $\lambda_U$ becomes extremely small, our SH-CDL would degenerate to SH-CDL-T2 where the collective preferences are not exploited. To study its impact, we test the performance of SH-CDL model by varying the value of $\lambda_U$ from 0.01 to 100, and the results are presented in Table 9. From the results, we observe that the recommendation accuracy of SH-CDL first increases with the increasing $\lambda_U$, and then begins to decrease. Our SH-CDL achieves its best performances with different $\lambda_U$ values in the two different recommendation scenarios: Home-Town $\lambda_U=0.1$ Vs. Out-of-Town $\lambda_U=10$. There are two possible reasons: 1) users tend to follow the public’s collective preferences when they travel in out-of-town regions, while they tend to follow their own preferences in their home town; and 2) users’ generated check-ins in out-of-town regions are too sparse to enable the inference of their preferences for these regions without social regularization, while most of users have enough check-in data in their home towns.

#### 6.8 Test for Cold-Start Recommendation

This experiment studies the effectiveness of the different recommender models for the cold start problem. We first test their performance in recommending cold-start POIs.

We first choose POIs with less than 10 check-ins as cold-start POIs, and then select users with at least one cold-start check-in as test users. For each test user, we first choose her check-ins associated with cold-start POIs as the test set, and the remaining check-ins as the training set. Our aim is to measure whether the marked-off cold-start POIs in the test set can be accurately recommended to the right users in the top-$k$ results. As SH-CDL, ST/GLDA, ST-LDA, DCF, CDL, HLDBN and DL-PMF exploit the auxiliary information of POIs (i.e., POI features), all of them have the potential to recommend cold-start POIs to users. We test their recommendation effectiveness for cold-start POIs on the Foursquare dataset and present the results in Figure 7. From the results, we observe that: 1) our SH-CDL model consistently outperforms the other models in recommending cold-start POIs, and its advantage is more obvious; 2) compared with the results in Figure 5, the recommendation accuracies of all models for cold-start POIs decrease to different degrees, e.g., the degree of decrease of ST-LDA and ST-GLDA is much larger than that of our SH-CDL, as our SH-CDL deeply exploits multiple types of POI features in a supervised way while ST-LDA and ST-GLDA only leverage the textual and temporal information in an unsupervised and shallow manner; and 3) the top-$k$ recommendation performances of all models show a more significant drop in the out-of-town setting than in the home-town scenario.
To test the recommendation performance for cold-start users, we choose users with less than 10 check-ins as cold-start users. For those cold-start users, we mark off their check-ins as the test set, and use the check-in records generated by the other users as the training set. Among all comparison methods, only ST-LDA and ST-GLDA are able to recommend POIs for cold-start users according to the collective preferences of the crowd with the same role in the target region. Thus, we compare our SH-CDL with ST-LDA, ST-GLDA and a strong baseline recommendation algorithm called Top-Pop that recommends the top-k POIs with highest popularity in the target region, and present the results in Figure 8. By comparing the Figures 5 and 8, we observe that, although the performances of ST-GLDA, ST-LDA and SH-CDL decrease to various degrees, for cold-start users, our proposed SH-CDL still achieves the best performance. As all SH-CDL, ST-LDA and ST-GLDA consider the role-aware crowd’s preferences, they are more accurate than Top-Pop. Additionally, ST-LDA and ST-GLDA can only capture the popularity of POIs among the users with the same roles, while our SH-CDL can capture the collective preferences towards the POI features. Thus, the POIs with not so many check-ins but with popular features can still be recommended to the cold-start users by our approach.

6.9 Model Training Efficiency
In this experiment, we evaluate the efficiency of our SH-CDL model training in 4 different computational hardware environments: Single-CPU, Singe-GPU, Multi-GPUs and Multi-GPUs cluster, and also compare the training time of ST-LDA model. We conduct this experiment on an Amazon EC2 Multi-GPUs cluster with maximum of 10 computing nodes, each equipped with 4 high-performance NVIDIA GPUs. The Gibbs EM algorithm is used for the inference of ST-LDA [45], and Gibbs sampling is performed in the E-step. Because the Gibbs sampling procedure does not involve any matrix operation, it cannot take advantage of the power of GPUs. To speed up the training of ST-LDA, we implemented the ST-LDA on the distributed GraphLab system with 10 nodes. Figure 9 shows the training time of our SH-CDL model and ST-LDA model. We can conclude that the modern computational hardware environment (GPUs clusters) can effectively guarantee the scalability of our SH-CDL. Obviously, the training of our SH-CDL based on the GPUs cluster is much more efficient than ST-LDA with parallelization. Additionally, we make the following observations: 1) Single-GPU is 6.3x faster than Single-CPU, showing the advantage of GPU in training deep learning models; 2) as expected, Multi-GPU-4 is nearly 4x faster than the Single-GPU; and 3) as the number of computing nodes increases from 1 to 10, there is only a 7.0x increase in speed for the Multi-GPU-Cluster due to the synchronization cost of minibatch training. In a cluster environment, computing nodes need to communicate with each other for every mini-batch in order to synchronize the shared variables, such as gradients. Given that both bandwidth and latency of networks are at least 10x worse than physical memory, this overhead cannot be ignored.

7 Related Work
In general, our work is closely related to content-aware collaborative filtering and POI recommendation.

7.1 Content-aware Collaborative Filtering
To address the issues surrounding the data sparsity and cold start, the matrix factorization models have been widely extended to content-aware factorization models by incorporating the content features, such as regression-based latent factor model [1], LibFM [28], SVD [3] and ICCF [19]. These models are almost equivalent in model representation, but adopt different optimization algorithms. A latent factor vector is learned for each feature besides users and items. Inspired by the success of word2vec [24], some embedding techniques were developed to embed features, users and items in the same latent space [32], [38]. However, these methods exploited the content features in a shallow way and assumed the interactions between features are linear.

Recently, some deep learning models have been integrated to learn a deep item representation from its associated content information for recommendation [6], [17], [31], [33], [37], [47], [54]. For example, Oord et al. [31], Tim et al. [15] and Zheng et al. [54] applied a convolutional neural network (CNN) to process the content features. Wang et al. [37] utilized deep belief nets (DBN) to process the music content. They assumed that a user has a latent factor vector drawn from a Gaussian prior, and a song has a latent factor vector learned by a deep belief network. Then, the matrix factorization was used to compute the preference score. Wang et al. [33] proposed a hierarchical Bayesian model called collaborative deep learning (CDL) which used the stacked denoising auto-encoders (SDA) to process the item content and extended the collaborative topic regression (CTR) to compute user
preferences. Li et al. [17] extended CDL by replacing the SDA with the marginalized denoising stacked auto-encoders.

Although these existing works have combined deep learning models with collaborative filtering to improve the recommendation, our work in this paper is distinct from theirs in several points. First, compared to our SH-CDL, existing models ignore the spatial dynamic of personal preferences. Second, they do not consider the heterogeneity and multi-modality of item content features. Third, they do not exploit the social and spatial correlations to alleviate the data sparsity issue in learning user preferences.

Our proposed SH-CDL is a generalized deep learning-based framework for POI recommendation and can be easily extended by other deep learning techniques to process the rich POI features. For example, the ReLU architecture [6] can be adopted to process the content features of POIs and deals with the sparse textual features by the word2vector technique. The deep learning matcher in [47] can be modified to process the textual content of POIs and then join with Gaussian restricted Boltzmann machine to obtain a unified POI representation. Similarly, the convolutional neural networks (CNN) in [54] can also be used to replace the replicated softmax model to process the textual features of POIs. The stacked denoising autoencoder in [33] can replace the Gaussian restricted Boltzmann machine to process the dense real-value features.

7.2 POI Recommendation

Many recent studies [5], [20], [40], [43], [55] have shown that there is a strong correlation between user check-in activities and geographical distance as well as social connections, so most of current POI recommendation work mainly focuses on leveraging the geographical and social influences to improve recommendation accuracy. For example, Ye et al. [40] delved into POI recommendation by investigating the geographical influences among locations and proposed a framework that combines user preferences, social influence and geographical influence. The temporal effect of user check-ins in LBSNs has also attracted much attention from researchers. POI recommendation with temporal effect mainly leverage temporal cyclic patterns [11], [48], temporal sequential patterns [22], [36], [53] and temporal matching between POI popularity and user availability [39]. Recently, researchers explored the content information of POIs to alleviate the issues surrounding data sparsity and cold start. Yin et al. [41], [44] exploited both personal interests and local preferences based on the contents of POIs. Zhao et al. [52] integrated both POI-associated contents and user sentiment information into POI recommendation. Jian et al. [19] proposed an implicit-feedback based content-aware collaborative filtering framework to incorporate the semantic content for POI recommendation. To support real-time POI recommendation, our work in this paper is distinct from theirs in several points. First, compared to our SH-CDL, existing models ignore the spatial-aware dynamic user preference modeling. To deal with the multimodal heterogeneous features, we extended the DBN to MDBN by introducing a late feature fusion strategy. Extensive experiments were conducted, and the experimental results showed that our SH-CDL model significantly outperforms the state-of-the-art recommendation methods.

8 CONCLUSIONS

In this paper, we developed a novel POI recommendation model SH-CDL to jointly perform deep representation learning for POIs from heterogeneous features and hierarchically additive representation learning for spatial-aware personal preferences to overcome the challenges of the spatial dynamics of user preferences, cold start and data sparsity. Social regularization and spatial smoothing technologies were developed to overcome data sparsity in the spatial-aware dynamic user preference modeling. To deal with the multimodal heterogeneous features, we extended the DBN to MDBN by introducing a late feature fusion strategy. Extensive experiments were conducted, and the experimental results showed that our SH-CDL model significantly outperforms the state-of-the-art recommendation methods.

ACKNOWLEDGMENT

This work was supported by ARC Discovery Early Career Researcher Award (Grant No. DE160100308), ARC Discovery Project (Grant No. DP140103171) and New Staff Research Grant of The University of Queensland (Grant No.613134). It was also partially supported by National Natural Science Foundation of China (Grant No. 61572335 and 61672501).

REFERENCES


