Information Sciences 483 (2019) 294–312

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

iMCRec: A multi-criteria framework for personalized point-of-interest recommendations

Yan Lyu[a,*], Chi-Yin Chow[a], Ran Wang[b], Victor C.S. Lee[a]

[a] Department of Computer Science, City University of Hong Kong, Hong Kong, China
[b] College of Mathematics and Statistics, Shenzhen University, China

ARTICLE INFO

Article history:
Received 9 December 2016
Revised 10 May 2018
Accepted 14 January 2019
Available online 17 January 2019

Keywords:
Point-of-interest recommendations
Location recommendations
Personalized recommendations
Multi-criteria decision making
Location-based social media

ABSTRACT

User preferences on geographical positions and categories of points-of-interest (POIs) have been exploited for POI recommendations. However, in reality, the users' visiting behaviors are also affected by the attributes of POIs, which reflect the important features of the POIs. Integrating the geographical, category and attribute criteria for POI recommendations suffers challenges of (1) modeling user preferences on multiple attributes with different values, (2) integrating conflicting multiple criteria for POI recommendations, and (3) learning personalized weights from one's check-in histories with heterogeneous data types. To address these challenges, we propose a new personalized POI recommendation framework, called iMCRec, which recommends POIs by integrating user preferences on geographical, category and attribute criteria with personalized weights. In iMCRec, preference models are first built for individual user's geographical, category, and attribute preferences. Especially, we propose an attribute preference model by considering both preferences on values of each attribute and importance of different attributes. A sophisticated collaborative filtering method is also developed to fuse the opinion of similar users under the three criteria separately. To learn the personalized weights on the three criteria, a weight learning strategy is proposed. We then develop a fast Multi-Criteria Decision Making (MCDM) algorithm, called FastMCDM, to integrate the three conflicting criteria and efficiently generate top-*N* POIs as recommendations. Finally, we evaluate the performance of our iMCRec through extensive experiments using two real-world datasets collected from Yelp. Experimental results show that iMCRec not only performs better than the state-of-the-art POI recommendation techniques, but it is also more flexible in dealing with the scale problem and more effective in learning personalized weights than other multi-criteria-based techniques.

1. Introduction

The increasing use of GPS-enabled smart phones has popularized a number of location-based social media, such as Yelp and Foursquare, where users can check in a point-of-interest (POI). POI recommendations have been considered as a

* Corresponding author.
E-mail addresses: yanlv2-c@my.cityu.edu.hk (Y. Lyu), chiychow@cityu.edu.hk (C.-Y. Chow), wangran@szu.edu.cn (R. Wang), csvlee@cityu.edu.hk (V.C.S. Lee).

https://doi.org/10.1016/j.ins.2019.01.043
0020-0255/

Table 1
Selected attributes with possible values.

Attribute	Possible values
Accepts Credit Cards	true, false
Alcohol	beer and wine, full bar, none
Attire	casual, dressy, formal
Good for Groups	true, false
Good for Kids	true, false
Noise Level	average, quiet, loud, very loud
Outdoor Seating	true, false
Price Range	1, 2, 3, 4
Wi-Fi	free, no, paid

POIs	Details	Preference Levels
POI A	0.04 km	★★★★
	Italian Restaurant	★★
	No Wi-Fi	★
POI B	1.6 km	★★
	Japanese Restaurant	★★★★
	No Wi-Fi	★
POI C	3.3 km	★
	Chinese Restaurant	★★★
	Free Wi-Fi	★★★

Fig. 1. Conflicts of user preferences.

value-added service that models users' preferences from their historical check-ins and recommends new POIs to them, in order to help them explore surroundings and POIs attract more potential visitors.

Most existing POI recommendation techniques model user preferences based on two criteria. (1) **The geographical criterion**. The geographical position of a POI has a significant influence on the user's visiting behaviors, and thus this criterion is a major factor in location recommendations [7,20,39,42–44]. The geographical distribution of the visited locations is usually used to estimate the probability of a new location to be visited [7,39,42–44]. (2) **The category criterion**. A POI usually belongs to a main category, e.g., restaurant, shopping, and arts. This criterion also significantly affects the user's visiting behaviors. For instance, POIs are recommended to users by analyzing their category transition patterns [22], or by consulting local experts with similar category preferences [5].

Although both the geographical position and category of POIs are two important factors in POI recommendations, the user's visiting or check-in behaviors are also significantly affected by the **attributes of POIs** [24,30]. For example, people will consider the price level of a restaurant before having a meal there, alcoholics like a dining place with a full bar, and Internet addicts prefer restaurants offering free Wi-Fi. To the best of our knowledge, there is no existing work that models users' attribute preferences for POI recommendations. To this end, we are motivated to consider the attributes of POIs as a new criterion, called **attribute criterion**, and incorporate it with geographical and category criteria into POI recommendations.

However, integrating the geographical, category and attribute criteria for POI recommendations has three key challenges:

- **Attribute preference modeling**. A POI usually has multiple attributes and each of the attributes has multiple possible values, as listed in Table 1. Either the geographical or the category preference model can hardly be used for modeling a user's preference on attributes. On the one hand, a user may have different preferences on different values of a certain attribute. Take "Price Range" as an instance, one may prefer POIs in the price range of 2 or lower rather than those in higher price ranges. On the other hand, different attributes are of varying importance to the user. For example, the user may care more about the "Price Range" than other attributes such as "Attire" or "Noise Level". Both preferences on different values of each attribute and importance of different attributes need to be considered in modeling the user's attribute preference.
- **Conflicting criteria**. The user may rank POIs differently according to different preference criteria. As depicted in Fig. 1, there are three candidate POIs: A, B and C. For the geographical criterion, the user prefers POI A because it has the shortest travel distance; however, she also prefers B or C in terms of category or attribute. It is hard to decide which POI is the best one for this user when considering the three criteria simultaneously. Although a few existing POI recommenders have considered both geographical and category criteria [5,22], these systems are not readily extendable to integrate more criteria. New approaches have to be explored to integrate multiple criteria for POI recommendations.
- **Personalized weight on each criterion**. The weight on each criterion varies from user to user. For example, the travel distance is more important than the categories and attributes of POIs for busy people. Food lovers would like to explore POIs in category of "Restaurant" or "Food" regardless of their positions and attributes. A mother would concern more about whether a POI is "Good for Kids" rather than its position and category. It is necessary to assign personal-

ized weights on the criteria for different users according to their unique preferences. However, it is challenging to mine the personalized weights from one's visited POIs. This is because each visited POI consists of three heterogeneous data types in terms of the geographical, category and attribute criteria, i.e., two-dimensional geographical coordinate, enumerated category and attribute-value pairs. Personalized weights have to be learned in a unified way that can handle such heterogeneous data types.

To address these three key challenges, we propose a new POI recommendation framework, called **iMCRec**, to recommend POIs by integrating a user's geographical, category and attribute preferences using Multi-Criteria Decision Making (MCDM) with personalized weights. Given a user, iMCRec first builds her geographical, category and attribute preference models from her visited POIs separately. In particular, a new attribute preference model is proposed to distinguish the importance of different attributes of a POI to the user and the user's preference on different values of each attribute. To recommend POIs, the user's ratings on each unvisited POI are then estimated using the three preference models, respectively. Moreover, the personalized weight learning strategy learns the optimal weight of each criterion for the user from her check-in histories. Finally, an MCDM-based algorithm, called FastMCDM, integrates the ratings from the three criteria with personalized weights and generates top-N POIs as recommendations.

Note that user preferences on POIs may change over time. It is possible for iMCRec to closely follow users' evolving preferences by regularly incorporating their latest POI visiting histories into their preference models and the personalized weight learning process. The main contributions of this paper can be summarized as follows:

- We propose a new recommendation framework, called iMCRec, that models a user's preferences on the geographical, categories, and attribute criteria, respectively, and recommends POIs by integrating the three criteria with personalized weights to the user.
- To model a user's preferences on attributes of POIs, a new attribute preference model is developed, which considers both preferences on different values of each attribute and importance of different attributes.
- To fuse a user's personalized preferences with the opinion of the similar users, we develop a sophisticated collaborative filtering method, which measures the similarities of users under the geographical, category and attribute criteria separately.
- To integrate the geographical, category and attribute criteria for POI recommendations, we propose a fast MCDM algorithm, called FastMCDM, with an alternative filtering step. A learning strategy is also developed to learn the personalized weights on the three criteria.
- Extensive experiments are conducted to evaluate the performance of iMCRec using two real-world datasets collected from Yelp. Experimental results show that iMCRec outperforms the state-of-the-art POI recommendation techniques and other multi-criteria approaches (i.e., simple weighted sum, ensemble ranking and Skyline).

The rest of this paper is organized as follows. Section 2 highlights related work. Section 3 gives motivation examples and outlines the framework of iMCRec. Section 4 discusses how to model a user's geographical, category, and attribute preferences on POIs. Section 5 presents how to integrate the three criteria with personalized weights for POI recommendations. Experiment settings and results are presented in Sections 6 and 7 separately. Finally, Section 8 concludes this paper.

2. Related work

In this section, we highlight related work in POI recommendations.

2.1. POI recommendations

Collaborative filtering (CF) [3,17,19] is the most conventional approach for POI recommendations. It assumes that people with similar visiting histories tend to visit similar POIs in the future, and recommends new POIs by consulting other users with similar preferences to the target user.

The geographical criterion has been considered in POI recommendations, e.g., some studies assume that a user's check-in probability to a location follows the power law distribution [39] or multi-center Gaussian distribution [7]. However, most of these methods use the same distribution for all users, so they neglect personalization. To learn personal distributions for different users, Kernel density estimation is employed, and POIs recommendations are generated based on each user's unique geographical preference [42–44].

The category information of POIs is another important criterion in POI recommendations [5,22,40]. Some existing POI recommenders utilize a category hierarchical tree to model the user's preferences [5,22]. In [5], the category similarities between a new visitor and local experts are calculated, and POIs are then recommended to new visitors by consulting the local experts with similar category preferences. Moreover, the user's preference transitions on different categories are studied [22], which are used to predict the categories of new POIs.

It is important to note that all these methods model the user's preferences based on either geographical or category criterion, or both of them. In reality, the user's choices on POIs are also affected by attribute criteria, so it is crucial for POI recommenders to consider all the three criteria.

2.2. Recommendation using attributes

Attributes of products have been considered as an important context in *non-spatial* product recommendations. Conventional content-based filtering methods of product recommendations use attributes to match user's preferences [11,28,32], or determine the similarity between products [8]. However, POIs are different from non-spatial products because the user has to visit a POI physically, and thus the geographical positions of POIs significantly affect the user's visiting behaviors. It is a timely topic to study how to incorporate the attributes of POIs into a POI recommender system and deal with the conflicts between the attribute criterion and other criteria.

2.3. Multi-criteria approaches for POI recommendations

Multi-criteria-based recommenders have been well developed for *non-spatial* product recommendations [1,21]. In these systems, the attributes of products are considered as multiple criteria [1,26], or the ratings to the products given by users are in multiple dimensions [2,15,21]. However, the techniques in these systems are not applicable for POI recommendations. Different from non-spatial products, the geographical positions of POIs significantly affect the user's visiting behaviors, and the user's geographical preferences may conflict with the non-spatial features of the POIs. Besides, since there is no user-specified rating to each feature of POIs, (i.e., the geographical positions, category, or attribute of POIs), we are motivated to model and estimate the user's personal preferences on each criterion.

There are a few other techniques, such as weighted sum, ensemble ranking and skyline queries that deal with multi-criteria problems. (1) As the simplest method, the weighted sum suffers from the scale problem: the ratings in different criteria are usually in different scales, which may make this method less effective. (2) Ensemble ranking [16,18,34] has been utilized to learn appropriate weights for combining multiple rankers in the context of information retrieval applications [18]. Existing methods have the shortcoming of query independent, which cannot be adapted to learn personalized weights for a user. Hoi et al. [18] proposed a semi-supervised ensemble ranking that learns query-dependent weights to combine multiple rankers. However, this method focuses on the score-based ranking approaches rather than order-based approaches, so it still has the scale problem. (3) Skyline queries [9,25,27] deal with multi-criteria problems by selecting the alternatives that are not dominated by any others in all criteria (i.e., dimensions). However, the selected skylines may be good in terms of one criterion, but are bad in other criteria. It is undesirable to recommend such alternatives to users. Also, skyline queries usually assume that all the criteria are of the same importance, which makes it not appropriate to recommend POIs to users who have different weights on each criterion.

MCDM [12,14,23] is a well-known branch of decision making. MCDM evaluates and ranks a set of alternatives based on multiple conflicting criteria [6], and selects the best ones by aggregating them with a trade-off mechanism [37]. The evaluation on alternatives are ordinal, which makes it more flexible to combine multiple conflicting criteria with different weights. With the advantage of being flexible and capable of dealing with conflicting criteria, MCDM is suitable for POI recommendations. However, there are only a few POI recommenders that adopt the MCDM strategy. For example, a gas station recommender for vehicles is proposed in [4], where multiple attributes of gas stations, such as price and brand, are considered as the multiple criteria; and a context-aware recommendation system is proposed for location-based services [13]. Unfortunately, these two recommenders obtain the personalized preferences through questionnaires, which is not efficient for POI recommendations. Our iMCRec learns the user's preferences from their visited POIs based on geographical, category and attribute criteria, and recommends POIs by integrating the three criteria with personalized weights.

3. Motivation examples and framework overview

In this section, we first introduce real-world examples that motivate our work and then present the overview of our iMCRec framework.

3.1. Motivation examples

We use real-world examples to illustrate that (1) each user has unique preference on different attributes of POIs, (2) each user weighs the geographical, category and attribute criteria differently. Fig. 2 depicts the geographical check-in density, category and attribute distributions of visited POIs of three different users, u_1, u_2 and u_3, which are randomly selected from the dataset provided by Yelp. We use the Gaussian probability density model to estimate the geographical check-in density for each user. To show the distributions of user preferences on categories and attributes, we use bar charts to plot the percentage of visited POIs that are with a certain category and have a certain attribute value, respectively. In these bar charts, the black lines connecting the median value of each bar demonstrate the unique distribution of each user's category or attribute preferences.

From the attribute distributions of the three users, we observe each user has unique preference on different attributes of POIs. u_3 prefers POIs with "Full Bar Alcohol" and "Waiter Service" and u_2 slightly likes POIs with "Casual Attire" and "Good for Kids", while u_1 doesn't show any special preference on any attributes. Therefore, it is necessary to build personalized attribute preference model for each user.

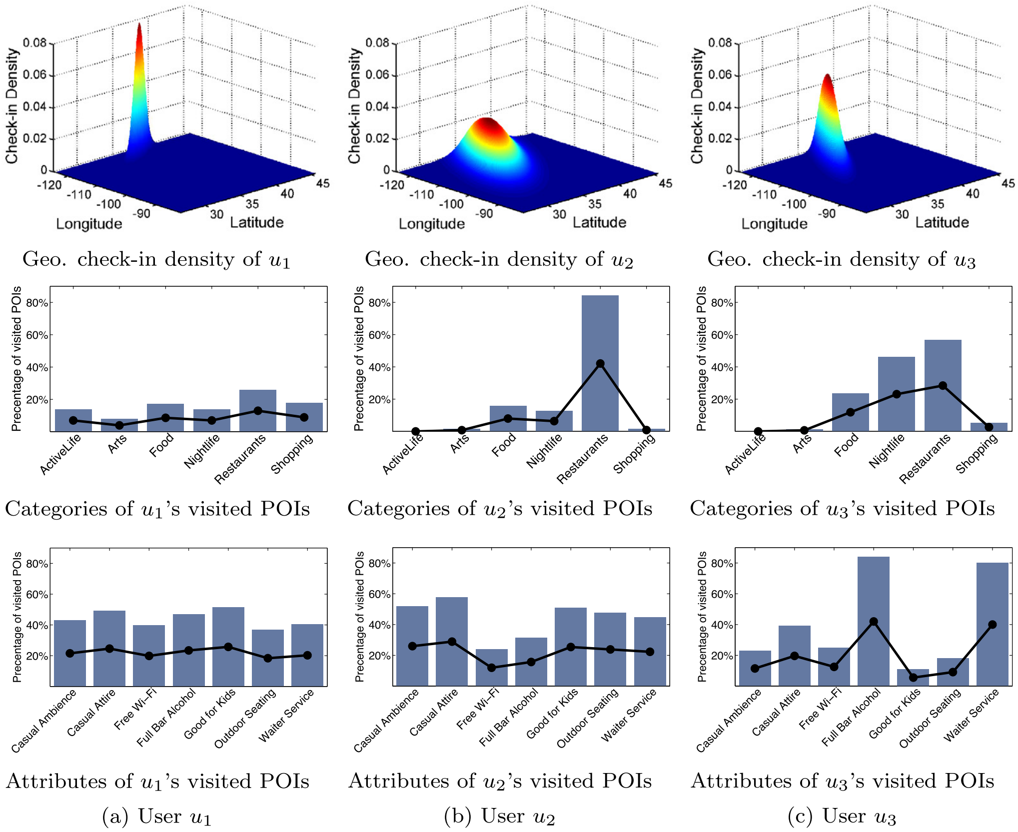


Fig. 2. Preferences on the geographical positions, categories, and attributes of three users' POIs.

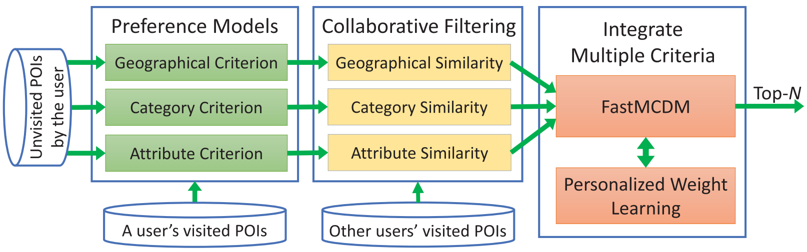


Fig. 3. Framework of iMCRec.

By comparing the geographical, category and attribute distributions between the three users, we can observe that these users weigh differently on the three criteria. u_1 has a concentrated geographical check-in distribution, showing her strong preference on nearby POIs; but the category and attribute distributions of her visited POIs are scattered, i.e., no special preference on any category or attribute. It is obvious that u_1 weighs more on the geographical criterion than the category and attribute criteria. Compared with u_1, u_2 weighs more on the category criterion, (80% of u_2's visited POIs are restaurants), but weighs less on the geographical and attribute criteria (i.e., scattered geographical and attribute distributions); u_3 weighs more on the attribute criterion, (more than 80% of u_3's visited POIs have the attributes of "Full Bar Alcohol" and "Waiter Service"), and weighs relatively less on the geographical and attribute criteria. Therefore, it is necessary to assign personalized weights on the three criteria for each user.

3.2. An overview of iMCRec

The framework of iMCRec is depicted in Fig. 3. Given a user, iMCRec first builds her geographical, category and attribute preference models from her visited POIs separately. The user's ratings on each unvisited POI are then estimated using the three preference models, respectively. The collaborative filtering method measures the similarities of users under the geographical, category and attribute criteria separately, and fuses the user's ratings with similar users under each of the three

Table 2
Key notations.

Symbol	Description
$\boldsymbol{l}$	A POI
$\mathcal{L}$	Set of all POIs in an LBSN
$\mathcal{L}_u$	Set of POIs visited by user u
$\mathcal{L}_u^*$	Set of POIs that have not been visited by user u
$r_u(\boldsymbol{l}_i)$	Visiting frequency of user u on POI $\boldsymbol{l}_i$
c	A category of POI, $c_h^{(j)}$ denotes the hth level category for POI $\boldsymbol{l}_j$
$\mathcal{T}$	Set of attributes of POIs
a	An attribute of POI, $a \in \mathcal{T}$
$\mathcal{V}$	Set of possible values for an attribute
v	A value of an attribute, $v \in \mathcal{V}$, $v_{a_t}^{(j)}$ denotes value of the tth attribute a_t for POI $\boldsymbol{l}_j$
$\mathcal{C}$	Set of criteria
C	A criterion, $C \in \mathcal{C}$
$\psi_k^{\succ}(\boldsymbol{l}_i)$	kth dominated index of $\boldsymbol{l}_i$
$\psi_k^{\prec}(\boldsymbol{l}_i)$	kth dominating index of $\boldsymbol{l}_i$
$\Phi^{\succ}(\boldsymbol{l}_i)$	Dominated index of $\boldsymbol{l}_i$
$\Phi^{\prec}(\boldsymbol{l}_i)$	Dominating index of $\boldsymbol{l}_i$
$D(\boldsymbol{l}_i)$	MCDM-based rating score of $\boldsymbol{l}_i$
$\boldsymbol{w}$	Weight vector, each element denotes a weight for a criterion

criteria. Moreover, the personalized weight learning strategy learns the optimal weight of each criterion for the user. Finally, the MCDM-based algorithm, called FastMCDM, integrates the ratings from the three preference criteria with personalized weights and iteratively generates top-N POIs as recommendations. The key symbols used in this paper are summarized in Table 2.

4. Modeling user preferences on multiple criteria

In this section, we first describe how to model a user's geographical and category preferences by employing the two-dimensional Kernel Density Estimation (KDE) and TF-IDF techniques, respectively. Then a new attribute preference model is proposed to estimate a user's preferences on attributes of POIs. Finally, we develop a sophisticated collaborative filtering method to fuse each preference model with the opinion of similar users.

4.1. Geographical criterion

Since each user has a unique preference on the geographical positions of POIs, we should personalize the geographical preference for each user [42,44]. The two-dimensional KDE has been proved effective in modeling the geographical preference as a personal distribution for each user [43]. It predicts the probability of a user visiting any new location using her personal distribution and hence achieves better performance compared to other state-of-the-art geographical POI recommendation techniques [43]. Therefore, we leverage this technique to model a user's geographical preference.

Geographical preference model. Given a user u, let $\mathcal{L}_u = \{\boldsymbol{l}_1, \boldsymbol{l}_2, \dots, \boldsymbol{l}_m\}$ be the set of m POIs that have been visited by u, and each POI $\boldsymbol{l}_i = (lat_i, lon_i)^{\mathrm{T}}$ is a two-dimensional vector with the latitude (lat_i) and longitude (lon_i). The kernel density estimator is given by:

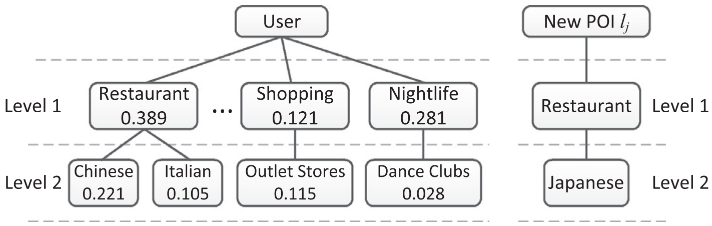
$$\hat{f}(\boldsymbol{x}) = \frac{1}{m\sigma^2}\sum_{i=1}^{m} K\left(\frac{\boldsymbol{x}-\boldsymbol{l}_i}{\sigma}\right), \tag{1}$$

where $\hat{f}(\boldsymbol{x})$ is the estimated check-in probability density for location $\boldsymbol{x}$, $K(\cdot)$ is the kernel function and we adopt the standard two-dimensional normal kernel function [33] (i.e., $K(\boldsymbol{x}) = \frac{1}{2\pi}\exp(-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x})$), and σ is a smooth bandwidth (i.e., $\sigma = m^{-\frac{1}{6}}\sqrt{\frac{1}{2}\hat{\boldsymbol{\sigma}}^{\mathrm{T}}\hat{\boldsymbol{\sigma}}}$, where $\hat{\boldsymbol{\sigma}}$ is the marginal standard deviation of $\mathcal{L}_u$ [33]).

The estimator of Eq. (1) models the check-in probability density on the two-dimensional geographical coordinates of check-ins for each user, and a higher value of $\hat{f}(\boldsymbol{x})$ indicates that a user has a higher probability of visiting location $\boldsymbol{x}$. Thus, the developed estimator can estimate a user's geographical preference on a new POI based on its geographical coordinates.

Geographical preference estimation. The geographical preference of the user on POI $\boldsymbol{l}_j = (lat_j, lon_j)$, denoted by $GeoRating(\boldsymbol{l}_j)$, is calculated as:

$$GeoRating(\boldsymbol{l}_j) = \frac{1}{2\pi m\sigma^2}\sum_{i=1}^{m}\exp\left(-\frac{(\boldsymbol{l}_j-\boldsymbol{l}_i)^{\mathrm{T}}(\boldsymbol{l}_j-\boldsymbol{l}_i)}{2\sigma^2}\right). \tag{2}$$



(a) Example of a hierarchical TF-IDF tree. Each number represents the TF-IDF value for a category.

(b) Category hierarchy of a new POI.

Fig. 4. Hierarchical representation of POI categories.

4.2. Category criterion

TF-IDF has been used for modeling user preferences on POI categories [5], motivated by the following two facts: (1) the frequency of a POI category visited by a user can reflect her preference on this category; and (2) a high visiting frequency to POIs in a certain category could also be caused by its popularity among all POIs. In this section, we utilize TF-IDF to model category preference and propose a new method to estimate the user's category rating on a new POI by considering all its categories at different levels.

Category preference model. Categories of a user's visited POIs are organized as a hierarchical TF-IDF tree. Fig. 4(a) depicts an example of a user's two-level TF-IDF tree, where each node represents a TF-IDF value for the category or subcategory of a POI that has been visited by the user. Let c be a category in the hierarchical TF-IDF tree. The TF-IDF value of the category is calculated as:

$$tf\text{-}idf(c) = \frac{n_c}{n} \cdot \log \frac{|\mathcal{L}|}{|\mathcal{L}_c|}, \tag{3}$$

where n_c is the user's visiting frequency to POIs with category c, n is the number of the user's visit records, $|\mathcal{L}|$ is the number of POIs, and $|\mathcal{L}_c|$ is the number of POIs belonging to category c. Note that many POIs do not belong to any subcategories (e.g., some POIs in the "Nightlife" category do not belong to any subcategories), so the visiting frequency of a category could be larger than the sum of that of its subcategories.

Category preference estimation. Note that a POI has multiple categories in different levels. To estimate the user's category rating on a new POI by considering all its categories in all levels, we calculate the weighted sum of the user's preferences on each category at each level. Given a new POI $\boldsymbol{l}_j$ with its categories in each level, $\{c_1^{(j)}, c_2^{(j)}, \ldots, c_H^{(j)}\}$, where H is the number of category levels of $\boldsymbol{l}_j$, the category rating of the user on POI $\boldsymbol{l}_j$, denoted by $CateRating(\boldsymbol{l}_j)$, is calculated by a weighted sum of the user's preferences on the category of $\boldsymbol{l}_j$ at each level:

$$CateRating(\boldsymbol{l}_j) = \sum\nolimits_{h\in\{1,2,\ldots,H\}} \beta \cdot tf\text{-}idf(c_h^{(j)}), \tag{4}$$

where $\beta = \frac{1}{2^{H-h}}$ and it gives a smaller weight to a category at a lower level (e.g., the weight of Restaurant should be smaller than that of Japanese in Fig. 4 (b)).

Our category preference estimation method (i.e., Eq. (4)) has two properties. (1) It considers all the categories in different levels of a new POI. As an example depicted in Fig. 4(b), given a new POI with category "Japanese", if we only consider the subcategory "Japanese" in the category hierarchy, we would predict that the user has no interest in the new POI. However, if we consider its parent category "Restaurant" with a TF-IDF value of 0.389, the user may like the new POI to some extent. (2) It gives a smaller weight to a category at a lower level, because a user's preference on a parent category cannot completely represent her preference on its specific subcategories and more specific subcategories provide more information about a user's preference.

4.3. Attribute criterion

A POI usually has multiple attributes and each attribute has multiple values, as shown in Table 1. Note that POIs in different categories may have different attributes. For the sake of explanation, we focus on the most popular attributes that all the POIs have.

A user may have different preferences on different values of a certain attribute. Take "Price Range" as an instance, one may prefer POIs in the price range of 2 or lower rather than those in higher price ranges. Moreover, the user may also weigh differently on different attributes. For example, the user may care more about the "Price Range" of POIs than other attributes such as "Attire" or "Noise Level". Therefore, to learn a user's preference on attribute criteria, we propose a new attribute preference model that not only estimates a user's preference on different values of each attribute, but also distinguishes the importance of different attributes to the user.

Attribute preference model. To learn a user's preference on values of an attribute, we focus on the user's visiting frequency to POIs with different values of the attribute. Intuitively, a high visiting frequency to POIs a certain value of the attribute demonstrate the user's high preference on this value, however, this high frequency could also be caused by its popularity among all POIs, so we adopt TF-IDF to measure the preference on an attribute value. Given an attribute a and its possible values $\mathcal{V} = \{v_1, v_2, \ldots, v_{|\mathcal{V}|}\}$, the preference on value v_s, $v_s \in \mathcal{V}$, is calculated as:

$$tf(a = v_s) = \frac{n(a = v_s)}{n} \text{ and } idf(a = v_s) = \log \frac{|\mathcal{L}|}{|\mathcal{L}_{a=v_s}|},$$

where $n(a = v_s)$ is the number of visited POIs and $|\mathcal{L}_{a=v_s}|$ is the total number of POIs with v_s for attribute a.

To distinguish the importance between different attributes for the user, we focus on the diversity of each attribute's possible values. Intuitively, if the user thinks some attribute is of great importance to her, she should have a strong preference on a certain value of the attribute. Namely, this value has a much higher visiting frequency than other values, and the distribution of visiting frequencies to each value of this attribute are very skewed. On the contrary, if the user cares little for the attribute, she should have no specific preference on any values of the attribute. Namely, this attribute has a wide diversity of values. Therefore, we use *entropy* to describe the diversity of the values of attribute a in a user's visited POIs as:

$$E(a) = -\frac{1}{\log |\mathcal{V}|} \sum_{s=1}^{|\mathcal{V}|} tf(a = v_s) \cdot \log tf(a = v_s), \tag{5}$$

where $0 \le E(a) \le 1$. Obviously, a smaller entropy value indicates that the values of an attribute are less diverse, i.e., a user shows a stronger specific preference on the attribute. Hence, the importance of a is defined as:

$$EImportance(a) = 1 - E(a). \tag{6}$$

Attribute preference estimation. Given a new POI $\boldsymbol{l}_j$ with an attribute set $\mathcal{T} = \{a_1, a_2, \ldots, a_{|\mathcal{T}|}\}$ with values $\{v_{a_1}^{(j)}, v_{a_2}^{(j)}, \ldots, v_{a_{|\mathcal{T}|}}^{(j)}\}$, the attribute preference of a user on $\boldsymbol{l}_j$ is modeled as:

$$AttriRating(\boldsymbol{l}_j) = \sum_{t=1}^{|\mathcal{T}|} tf(a_t = v_{a_t}^{(j)}) \cdot idf(a_t = v_{a_t}^{(j)}) \cdot EImportance(a_t). \tag{7}$$

4.4. Fusing preferences with other users' opinions

In POI recommender systems, it is important to consider other users' opinions on POIs. The standard user-based CF model recommends new POIs by consulting other users with similar preferences to the target user. The user similarity is usually measured by cosine similarity, i.e., given a set of users U, the similarity between users u and u' is calculated by:

$$CosSim(u, u') = \frac{\sum_{\boldsymbol{l}_i \in \mathcal{L}} r_u(\boldsymbol{l}_i) \cdot r_{u'}(\boldsymbol{l}_i)}{\sqrt{\sum_{\boldsymbol{l}_i \in \mathcal{L}} r_u^2(\boldsymbol{l}_i)} \sqrt{\sum_{\boldsymbol{l}_i \in \mathcal{L}} r_{u'}^2(\boldsymbol{l}_i)}}, \tag{8}$$

where $r_u(\boldsymbol{l}_i)$ and $r_{u'}(\boldsymbol{l}_i)$ are the frequency of users u and u' visiting POI $\boldsymbol{l}_i$, respectively. Note that the above cosine similarity only considers users' visiting frequency to POIs, but ignores users' personal geographical, category and attribute preferences. Therefore, we propose a new sophisticated CF-based method, which strengthens the similarity measurement under each of the three criteria.

For the geographical criterion, users who live in close distance tend to visit the same POIs in neighborhood; and hence, the geographical similarity between users u and u' is defined as:

$$GSim(u, u') = 1 - \frac{distance(u, u')}{\max_{u'' \in U, u \neq u''} distance(u, u'')}, \tag{9}$$

where $distance(u, u')$ is the geographical distance between the home locations of u and u'. We estimate the home location for each user using the method proposed in [29].

For the category criterion, since users who have similar preferences on categories would have a higher probability of visiting the same POIs (e.g., shopping lovers want to shop at a newly opened shopping center and artistic people like to visit famous museums or galleries), we define the category similarity between users u and u' as:

$$CSim(u, u') = \frac{\sum_{c \in \boldsymbol{c_1}} tf\text{-}idf_u(c) \cdot tf\text{-}idf_{u'}(c)}{\sum_{c \in \boldsymbol{c_1}} tf\text{-}idf_u(c) \cdot \sum_{c \in \boldsymbol{c_1}} tf\text{-}idf_{u'}(c)}, \tag{10}$$

where $tf\text{-}idf_u(c)$ is the u's preference on category c calculated by Eq. (3) and $\boldsymbol{c_1}$ is a set of categories in the first level of the hierarchical TF-IDF tree.

For the attribute criterion, users who have similar preferences on attributes also have a higher probability of visiting the same POI. For example, mothers would gather at some POIs that have the attribute of "Good for Kids" and drinkers would visit some POIs that offer "Full Bar Alcohol". Given a set of attributes $\mathcal{T}$, the attribute similarity between users u and u' can be defined as:

$$ASim(u, u') = \sum_{a \in \mathcal{T}} \frac{\sum_{v \in \mathcal{V}_a} tf\text{-}idf_u(v) \cdot tf\text{-}idf_{u'}(v)}{\sum_{v \in \mathcal{V}_a} tf\text{-}idf_u(v) \cdot \sum_{v \in \mathcal{V}_a} tf\text{-}idf_{u'}(v)}, \tag{11}$$

where $tf\text{-}idf_u(v)$ indicates the preference of user u on the value v of attribute a and $\mathcal{V}_a$ is the set of values of attribute a.

The developed geographical, category and attribute similarities are used to strengthen the similarity measurement between users; and the rating of user u to new POI $\boldsymbol{l}_j$ is estimated by the ratings of other similar users. We next incorporate these similarities into user ratings as:

$$\hat{r}_{geo}(\boldsymbol{l}_j) = \frac{\sum_{u'\neq u} GSim(u,u')\cdot CosSim(u,u')\cdot r_{u'}(\boldsymbol{l}_j)}{\sum_{u'\neq u} GSim(u,u')\cdot CosSim(u,u')},$$

$$\hat{r}_{cate}(\boldsymbol{l}_j) = \frac{\sum_{u'\neq u} CSim(u,u')\cdot CosSim(u,u')\cdot r_{u'}(\boldsymbol{l}_j)}{\sum_{u'\neq u} CSim(u,u')\cdot CosSim(u,u')},$$

$$\hat{r}_{attri}(\boldsymbol{l}_j) = \frac{\sum_{u'\neq u} ASim(u,u')\cdot CosSim(u,u')\cdot r_{u'}(\boldsymbol{l}_j)}{\sum_{u'\neq u} ASim(u,u')\cdot CosSim(u,u')}.$$

Finally, the geographical, category and attribute preference models are fused with the estimated ratings through the product rule for new POI $\boldsymbol{l}_j$ as:

$$\begin{aligned} GeoRating_f(\boldsymbol{l}_j) &= GeoRating(\boldsymbol{l}_j)\cdot \hat{r}_{geo}(\boldsymbol{l}_j),\\ CateRating_f(\boldsymbol{l}_j) &= CateRating(\boldsymbol{l}_j)\cdot \hat{r}_{cate}(\boldsymbol{l}_j),\\ AttriRating_f(\boldsymbol{l}_j) &= AttriRating(\boldsymbol{l}_j)\cdot \hat{r}_{attri}(\boldsymbol{l}_j). \end{aligned} \tag{12}$$

5. Integrating multiple criteria with personalized weights for POI recommendations

This section presents how to integrate the three criteria with personalized weights for POI recommendations. We first introduce MCDM preliminaries, and then present our FastMCDM algorithm. Finally, we present a weight learning strategy to learn personalized weights for our FastMCDM algorithm.

5.1. Preliminaries

MCDM has two key concepts, namely, *preference preorder* and *distance of ordinal relations*.

Preference preorder. Given a set of POIs that have not been visited by a user u, $\mathcal{L}_u^* = \{\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_{|\mathcal{L}_u^*|}\}$, and a set of criteria, $\mathcal{C} = \{C_1, C_2, \ldots, C_{|\mathcal{C}|}\}$, for each criterion $C_k \in \mathcal{C}$, three ordinal relations can be defined as:

- $\boldsymbol{l}_i \succ \boldsymbol{l}_j$: $\boldsymbol{l}_i$ is preferred to $\boldsymbol{l}_j$, i.e., $C_k(\boldsymbol{l}_i) - C_k(\boldsymbol{l}_j) > T(C_k)$;
- $\boldsymbol{l}_i \prec \boldsymbol{l}_j$: $\boldsymbol{l}_j$ is preferred to $\boldsymbol{l}_i$, i.e., $C_k(\boldsymbol{l}_j) - C_k(\boldsymbol{l}_i) > T(C_k)$; and
- $\boldsymbol{l}_i \approx \boldsymbol{l}_j$: $\boldsymbol{l}_i$ is indifferent to $\boldsymbol{l}_j$, i.e., $|C_k(\boldsymbol{l}_i) - C_k(\boldsymbol{l}_j)| \leq T(C_k)$,

where $C_k(\boldsymbol{l}_j)$ represents the fused rating score of $\boldsymbol{l}_j$ with regard to a specific criterion C_k, (e.g., the fused rating score $GeoRating_f(\boldsymbol{l}_j)$, $CateRating_f(\boldsymbol{l}_j)$, or $AttriRating_f(\boldsymbol{l}_j)$ with regard to the geographical, category or attribute criterion in our framework, respectively), and $T(C_k)$ is a threshold of indifference, which is defined as:

$$T(C_k) = \frac{\max_{\boldsymbol{l}_j\in\mathcal{L}_u^*} C_k(\boldsymbol{l}_j) - \min_{\boldsymbol{l}_j\in\mathcal{L}_u^*} C_k(\boldsymbol{l}_j)}{|\mathcal{L}_u^*|}. \tag{13}$$

Based on these three relations, a preference preorder of the alternatives with regard to a criterion is given.

Definition 1. (Preference preorder.) Given a set of alternatives $\mathcal{L}_u^* = \{\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_{|\mathcal{L}_u^*|}\}$ and a criterion C_k on them, the order $\boldsymbol{l}_1^* \succ\approx \boldsymbol{l}_2^* \succ\approx \ldots \succ\approx \boldsymbol{l}_{|\mathcal{L}_u^*|}^*$ is called a preference preorder of $\mathcal{L}_u^*$ with regard to C_k if and only if they satisfy $\boldsymbol{l}_1^* \neq \boldsymbol{l}_2^* \neq \ldots \neq \boldsymbol{l}_{|\mathcal{L}_u^*|}^* \in \mathcal{L}_u^*$ and $C_k(\boldsymbol{l}_1^*) \geq C_k(\boldsymbol{l}_2^*) \geq \ldots \geq C_k(\boldsymbol{l}_{|\mathcal{L}_u^*|}^*)$.

We denote the preference preorder of $\mathcal{L}_u^*$ with regard to C_k as $Preorder(C_k)$. Obviously, $Preorder(C_k)$ indicates the priorities of the alternatives with respect to C_k. For example, $\boldsymbol{l}_1^*$ is the most preferred alternative and $\boldsymbol{l}_{|\mathcal{L}_u^*|}^*$ is the least preferred one in this preorder. In our framework, the preference preorders of $\mathcal{L}_u^*$ with regard to the geographical, category, and attribute criteria are represented by $Preorder(C_{geo})$, $Preorder(C_{cate})$, and $Preorder(C_{attri})$, respectively.

Distance of ordinal relations. Intuitively, in a preference preorder, an alternative with more relations of $\succ$ and fewer relations of $\prec$ should have a higher priority. The distance between relations is employed to measure the priority of an alternative for a criterion [6], which is defined based on the following conditions: (1) Non-negativity: $dist(R, R') \geq 0$ or $dist(R, R') = 0$ iff $R = R'$; (2) Symmetric: $dist(R, R') = dist(R', R)$; and (3) Triangle inequality: $dist(R, R') + dist(R', R'') \geq dist(R, R'')$; where R, R', $R'' \in \{\succ, \prec, \approx\}$. The adopted values of these distances are given in Table 3.

5.2. Fast MCDM algorithm for POI recommendations

We first introduce how to calculate MCDM-based ratings, and then present our fast MCDM algorithm.

Table 3
Distance of ordinal relations [38].

$dist(\cdot,\cdot)$	$\succ$	$\prec$	$\approx$
$\succ$	0	2	1
$\prec$	2	0	1
$\approx$	1	1	0

5.2.1. MCDM-based rating

The priorities of a given POI in $Preorder(C_{geo})$, $Preorder(C_{cate})$, and $Preorder(C_{attri})$ may be significantly different from each other. The same POI may be ranked with a very high priority in one criterion but a very low priority in another one. In this case, it is important to have an effective tradeoff mechanism among different criteria, in order to prioritize the best alternatives listed in a recommendation result.

Let the relation of $\boldsymbol{l}_i$ and $\boldsymbol{l}_j$ for a user with regard to criterion C_k be $R_{ij}^{(k)}$, where $R_{ij}^{(k)} \in \{\succ, \prec, \approx\}$. The kth dominated index of $\boldsymbol{l}_i$ is defined as:

$$\psi_k^{\succ}(\boldsymbol{l}_i) = \sum\nolimits_{j\neq i} dist(\succ, R_{ij}^{(k)}), \tag{14}$$

and the kth dominating index of $\boldsymbol{l}_i$ is defined as:

$$\psi_k^{\prec}(\boldsymbol{l}_i) = \sum\nolimits_{j\neq i} dist(\prec, R_{ij}^{(k)}). \tag{15}$$

By determining the relations between $\boldsymbol{l}_i$ and other alternatives, and summing up their distances to $\succ$ and $\prec$, the kth dominated index and the kth dominating index of $\boldsymbol{l}_i$ measure the degree of it being dominated and dominating others under the criterion C_k.

In our framework, the kth dominated index of $\boldsymbol{l}_i$ with regard to the geographical, category and attribute criteria are represented by $\psi_{geo}^{\succ}(\boldsymbol{l}_i)$, $\psi_{cate}^{\succ}(\boldsymbol{l}_i)$ and $\psi_{attri}^{\succ}(\boldsymbol{l}_i)$, respectively. And the kth dominating index with regard to the three criteria are represented by $\psi_{geo}^{\prec}(\boldsymbol{l}_i)$, $\psi_{cate}^{\prec}(\boldsymbol{l}_i)$ and $\psi_{attri}^{\prec}(\boldsymbol{l}_i)$, respectively.

To further consider a set of criteria, i.e., $\mathcal{C} = \{C_1, C_2, \ldots, C_{|\mathcal{C}|}\}$, the dominated index of $\boldsymbol{l}_i$ is defined as:

$$\Phi^{\succ}(\boldsymbol{l}_i) = \sum\nolimits_{k=1}^{|\mathcal{C}|} w_k \psi_k^{\succ}(\boldsymbol{l}_i), \tag{16}$$

and the dominating index of $\boldsymbol{l}_i$ is defined as

$$\Phi^{\prec}(\boldsymbol{l}_i) = \sum\nolimits_{k=1}^{|\mathcal{C}|} w_k \psi_k^{\prec}(\boldsymbol{l}_i), \tag{17}$$

where w_k is the personalized weight of the criterion C_k. The weights of the geographical, category, and attribute criteria will be determined through a weight learning strategy described in Section 5.3.

By aggregating the above ordinal evaluation under each criterion (i.e., the kth dominated and dominating indices) with weights, the dominated index and dominating index of $\boldsymbol{l}_i$ indicate the degree of being dominated and the degree of dominating the others with respect to all the criteria, respectively. Since an alternative with a higher dominating index and a lower dominated index should have a higher priority, the MCDM-based rating score is determined as:

$$D(\boldsymbol{l}_i) = \Phi^{\prec}(\boldsymbol{l}_i) - \Phi^{\succ}(\boldsymbol{l}_i). \tag{18}$$

5.2.2. FastMCDM algorithm

The basic MCDM algorithm usually iteratively calculates the MCDM-based rating $D(\boldsymbol{l}_i)$ for each POI alternative, and selects the POI with the highest value as the recommendation [37,38]. However, some alternatives can be identified as the ones that can never be the part of the recommendation before calculating the MCDM-based rating for them. To avoid unnecessary calculations and improve the efficiency, we develop a fast MCDM algorithm, called **FastMCDM**, with an *alternative filtering step*.

As depicted in Algorithm 1, our FastMCDM consists of three main steps. The algorithm first computes the preference preorders for the geographical, category, and attribute criteria, and then computes the kth dominating index for each alternative (Lines 2–6). After that, it prunes alternatives that must not be part of a resultant recommendation list (Lines 8–13). Finally, it finds out top-N alternatives with the highest priority in an iterative manner (Lines 15–21). We will give the detail of these steps.

Step 1: *k*th Dominating Index Calculation. This algorithm takes a set of new POIs $\mathcal{L}_u^*$ that have not been visited by user u as input (also called as a set of alternatives). It first computes the ratings of geographical, category and attribute criteria for each new POI $l_j \in \mathcal{L}_u^*$ by Eq. (12) (i.e., $GeoRating_f(\boldsymbol{l}_j)$, $CateRating_f(\boldsymbol{l}_j)$, and $AttriRating_f(\boldsymbol{l}_j)$, respectively). After determining the threshold of each criterion by Eq. (13), the preference preorders for the three criteria are initialized. Then the kth dominating index for each alternative in each criterion is computed by Eq. (15).

Algorithm 1 Fast MCDM Algorithm (FastMCDM).

Input: A set of visited POIs for user u: $\mathcal{L}_u = \{\boldsymbol{l}_i\}_{i=1}^{|\mathcal{L}_u|}$, a set of new POIs for u: $\mathcal{L}_u^* = \{\boldsymbol{l}_j\}_{j=1}^{|\mathcal{L}_u^*|}$, the number of POIs to be recommended: N, and the weight on each criterion: w_g, w_c, w_a.
Output: A set of top-N recommended POIs: $\mathcal{R}$.

```
//kth dominating index calculation step
for each l_j ∈ L*_u do
    Compute GeoRating_f(l_j), CateRating_f(l_j), and AttriRating_f(l_j) by Eq. (12)
end for
Determine the threshold of each criterion by Eq. (13), and initialize the preference preorders for the three criteria:
    Preorder(C_geo), Preorder(C_cate), and Preorder(C_attri)
Compute ψ_k^≺(·) for each alternative in each criterion by Eq. (15)
//Alternative filtering step
Initialize a set of remaining alternatives: L_r = L*_u
for each l_j ∈ L_r do
    if l_j is an impossible alternative based on Definition 2 then
        L_r = L_r − {l_j}
    end if
end for
//Iterative recommendation step
Set R = ∅
while |R| < N do
    Rearrange Preorder(C_geo), Preorder(C_cate), and Preorder(C_attri) for L_r
    Compute the MCDM-based rating score of each l_j, l_j ∈ L_r, by Eq. (18), i.e., D(l_j) = Φ^≺(l_j) − Φ^≻(l_j)
    Select alternative l* with the largest MCDM-based rating score, i.e., l* = argmax_{l_j ∈ L_r} D(l_j)
    R ← R ∪ {l*}; L_r = L_r − {l*}
end while
```

Step 2: Alternative Filtering. There are could be a large number of candidate POIs, but some of them can never be part of a recommendation result. To avoid unnecessary MCDM-based rating calculations for these alternatives, we propose this alternative filtering step.

Basically, if an alternative $\boldsymbol{l}_j$ is preferred to alternative $\boldsymbol{l}_i$ in terms of each criterion, the overall priority of $\boldsymbol{l}_j$ that combines all the criteria is higher than $\boldsymbol{l}_i$, i.e., $D(\boldsymbol{l}_j) > D(\boldsymbol{l}_i)$ regardless of the weights of all the criteria. Furthermore, if there are more than N alternatives that are all preferred to $\boldsymbol{l}_i$ in terms of all the criteria, $\boldsymbol{l}_i$ must not be included in the top-N POIs with the highest MCDM-based rating scores. Thus, such impossible alternative $\boldsymbol{l}_i$ should be pruned before generating the top-N POIs.

To identify impossible alternatives, we employ the kth dominating index $\psi_k^{\prec}(\boldsymbol{l}_i)$ to measure $\boldsymbol{l}_i$'s priority in the kth criterion, as defined in Definition 2.

Definition 2. (Impossible Alternative.) Given a set of alternatives $\mathcal{L}_u^* = \{\boldsymbol{l}_1, \boldsymbol{l}_2, \dots, \boldsymbol{l}_{|\mathcal{L}_u^*|}\}$, $\boldsymbol{l}_i$ is an impossible alternative if there are at least N alternatives with all kth dominating indices larger than that of $\boldsymbol{l}_i$, i.e., $\boldsymbol{l}_i$ satisfies

$$\left|\{\boldsymbol{l}_j | \psi_k^{\prec}(\boldsymbol{l}_j) > \psi_k^{\prec}(\boldsymbol{l}_i), \forall k \in [1, |\mathcal{C}|], \boldsymbol{l}_j \in \mathcal{L}_u^*\}\right| \geq N.$$

Specifically, with the geographical, category and attribute criteria, $\boldsymbol{l}_i$ is an impossible alternative if it satisfies $|\{\boldsymbol{l}_j | \psi_{geo}^{\prec}(\boldsymbol{l}_j) > \psi_{geo}^{\prec}(\boldsymbol{l}_i), \psi_{cate}^{\prec}(\boldsymbol{l}_j) > \psi_{cate}^{\prec}(\boldsymbol{l}_i), \psi_{attri}^{\prec}(\boldsymbol{l}_j) > \psi_{attri}^{\prec}(\boldsymbol{l}_i), \boldsymbol{l}_j \in \mathcal{L}_u^*\}| \geq N$. Let $\mathcal{L}_p$ be a set of impossible alternatives. After this alternative filtering step, a set of remaining candidate POIs is $\mathcal{L}_r = \mathcal{L}_u^* - \mathcal{L}_p$.

To efficiently identify impossible alternatives, we build a k-d tree for the set of alternatives $\mathcal{L}_u^*$. The k-d tree recursively divides the kth dimension space into halves at each level of the tree. Namely, each non-leaf node splits the alternatives into two groups based on the a split value of kth dimension. The left sub-tree consists of data points whose kth dominating indices are smaller than the split value, while the right subtree consists of data points having larger kth dominating indices. Each leaf node stores an alternative in $\mathcal{L}_u^*$. To search for alternatives whose kth dominating index is larger than $\psi_k^{\prec}(\boldsymbol{l}_i)$ in each k dimension, the searching could mainly focus on the right subtrees of the k-d tree. The searching process is terminated as long as $N+1$ such alternatives is discovered, and in this case, $\boldsymbol{l}_i$ is the impossible alternative. As the searching process never visits the subtrees consisting of data points whose kth dominating indices are smaller than $\psi_k^{\prec}(\boldsymbol{l}_i)$, the search space is greatly reduced.

Step 3: Iterative Recommendation. This step mainly finds out the top-N POIs with the highest MCDM-based rating scores in an iterative manner. In each iteration, the algorithm rearranges the three preference preorders for the remaining alternative set $\mathcal{L}_r$ with respect to the geographical, category, and attribute criteria. It then calculates the MCDM-based rating score for each alternative by Eq. (18), based on the rearranged preorders. Finally, the alternative $\boldsymbol{l}^*$ with the highest MCDM-based rating score in $\mathcal{L}_r$ is selected as a part of a recommendation result (i.e., $\mathcal{R}$), and removed from $\mathcal{L}_r$. This iterative process is repeated until N recommended POIs are generated.

Complexity Analysis. In Step 1, the algorithm first takes $O(n)$ time to calculate ratings of the three criteria for all the alternatives, and $O(n\log n)$ time to arrange the alternatives into three preorders ($n = |\mathcal{L}_u^*|$). The kth dominating index can also be calculated in $O(n)$. In Step 2, building a k-d tree for storing the all the alternatives takes $O(n\log n)$ time [36]. By searching the k-d tree, filtering all the impossible alternatives takes $O(n\log n)$ time. Step 3 runs $O(Nn^*)$ time, where N is an input constant, and n^* is the number of remaining alternatives (i.e., $n^* = |\mathcal{L}_r|$). n^* is significantly smaller than n. In summary, the time complexity of our framework is $O(n\log n)$.

5.3. Personalized weight learning for MCDM

As discussed in Section 3.1, different users weigh differently on each criterion. It is necessary to assign personalized weight for each criterion when recommending POIs to a user.

However, a user's weight information cannot be obtained directly, it is challenging to mine personalized weight on each criterion from one's visited POIs. This is because each visited POI consists of three heterogeneous data types in terms of the geographical, category and attribute criteria, i.e., two-dimensional geographical coordinate, enumerated category and attribute-value pairs. To the best of our knowledge, there is no existing method that can directly handle such three heterogeneous data types in a unified manner to differentiate the user's preference weights on the three criteria.

To this end, we propose a weight learning strategy based on the idea of supervised learning. Intuitively, if a user has a specific preference on a certain criterion, a larger weight on this criterion can help to achieve better recommendation quality. Therefore, the user's optimal weights can be obtained by testing whether they can achieve the best recommendation quality. Specifically, we split a user's historical POIs into two sets based on visiting timestamp. The first set with earlier timestamp is called weight training set, which is used for learning user preferences and generating top-N recommendations using FastMCDM with different combinations of weights on the three criteria. The second set is weight testing set, which is used for checking recommendations quality. The weights that achieve the best quality of recommendations are selected as the personalized weights for the user.

The quality of recommendations is measured by the number of discovered recommendations, namely, the size of intersection between the set of recommendations $\mathcal{R}$ and the POIs in the weight testing set, i.e., $|\mathcal{R}\bigcap \text{Weight Testing Set}|$. All weights are non-negative and their sum is equal to one, i.e.,

$$w_g + w_c + w_a = 1, w_g \geq 0, w_c \geq 0, \text{and } w_a \geq 0, \tag{19}$$

where w_g, w_c, and w_a denote the weight on geographical, category, and attribute criteria, respectively. Let $\mathcal{W}$ denote a set of weights, in which $\boldsymbol{w} = (w_g, w_c, w_a)$ satisfies Eq. (19), $\forall \boldsymbol{w} \in \mathcal{W}$. And let $\mathcal{R}_{\boldsymbol{w}}$ be a set of recommendations that are generated by the FastMCDM Algorithm with the input of $\boldsymbol{w}$ and the weight training set. Then, the optimal weight, denoted as $\boldsymbol{w}_{\text{opt}}$, can be obtained by searching the weight that can achieve the best recommendation quality, i.e.,

$$\boldsymbol{w}_{\text{opt}} = \underset{\boldsymbol{w}\in\mathcal{W}}{\operatorname{argmax}} \left|\mathcal{R}_{\boldsymbol{w}} \bigcap \text{Weight Testing Set}\right|. \tag{20}$$

Learning algorithm. We adopt exhaustive search to find $\boldsymbol{w}_{\text{opt}}$ with a searching step. However, two cases could happen during the searching:

(1) *More than one optimal weight vector achieves the best recommendation quality*. This case takes place due to the reason that the recommendation quality may not be sensitive to a small searching step. For example, given a searching step of 0.01, the weight $\boldsymbol{w}_1 = (0.78, 0.20, 0.02)$ and $\boldsymbol{w}_2 = (0.79, 0.19, 0.02)$ could both achieve the best recommendation quality. In this case, we select a representative optimal weight vector that has the minimum total Euclidean distance to the other optimal weight vectors. Let $\mathcal{W}_{\text{opt}} = \{\boldsymbol{w}_i\}_{i=1}^{|\mathcal{W}_{\text{opt}}|}$ be the set of optimal weight vector generated by Eq. (20), $\boldsymbol{w}_i = (w_g^i, w_c^i, w_a^i)$. The representative optimal weight vector, denoted as $\boldsymbol{w}_{\text{p}}$, can be defined as

$$\boldsymbol{w}_{\text{p}} = \underset{\boldsymbol{w}_i\in\mathcal{W}_{\text{opt}}}{\operatorname{argmin}} \sum\nolimits_{\boldsymbol{w}_j\in\mathcal{W}_{\text{opt}}} distance(\boldsymbol{w}_i, \boldsymbol{w}_j), \tag{21}$$

where $distance(\boldsymbol{w}_i, \boldsymbol{w}_j)$ is the Euclidean distance between $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$.

(2) *No optimal weight vector for some users*. This case happens for some users who cannot get good recommendations that will be discovered in their weight testing sets for any input weight vectors in this learning process. In other words, the intersection between the set of recommendations and the weight testing set is empty for those users, i.e., $|\mathcal{R}_{\boldsymbol{w}}\bigcap \text{Weight Testing Set}| = \emptyset, \forall \boldsymbol{w} \in \mathcal{W},$ so they cannot obtain their optimal weight vectors by Eq. (20). In this case, we simply assign them the most popular optimal weight vector that is shared by the largest number of users.

6. Experiment settings

In this section, we describe our experiment settings for evaluating the performance of iMCRec.

Table 4
Statistics about the experiment datasets.

State	Nevada (NV)	Arizona (AZ)
Number of POIs	19,467	29,498
Number of users	15,851	14,976
Number of check-ins	143,928	441,419
User-POI matrix density	1.3×10^{-3}	9.5×10^{-4}
Average number of check-ins per POI	21.88	14.96
Average number of visited POIs per user	25.71	28.01

6.1. Dataset

All experiments are conducted on a real-world dataset of POIs provided by the “Yelp Dataset Challenge”.[1] As one of the most popular LBSNs, Yelp not only allows users to check in their locations; but also provides the detailed attribute information of POIs, including price range, noise level, etc. Some of the attributes are depicted in Table 1.

Due to privacy concerns, this dataset does not provide any user check-in history, but it contains the users' reviews on POIs that can be treated as the user check-in records. To evaluate the effectiveness of iMCRec, experiments are conducted on the two datasets collected from Nevada (NV) and Arizona states (AZ), USA. Statistics of the two datasets are listed in Table 4.

6.2. Baseline approaches

To demonstrate the effectiveness of iMCRec, we compare our method with the following four types of recommendation techniques, i.e., (1) no-criterion techniques, (2) single criterion based techniques, (3) multi-criteria based techniques and (4) iMCRec without personalized weighting.

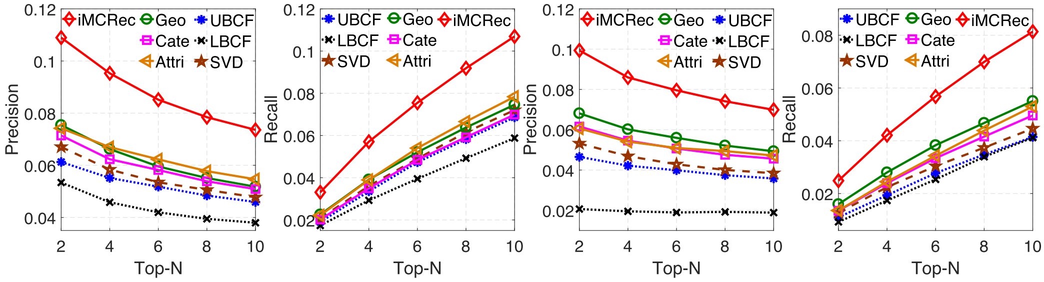
(1) No-criterion techniques
- User-based collaborative filtering (UBCF) [17]: This method recommends POIs that have been frequently visited by similar users. The similarity between users are measured by the cosine similarity between the POI visiting vectors.
- Location-based collaborative filtering (LBCF) [31]: This method recommends POIs that are similar to the user's visited POIs. The similarity between two POIs is measured using cosine similarity between user visiting vectors on the two POIs.
- Singular Value Decomposition (SVD) [10]: This method factorizes the user-POI visiting matrix to a product of two lower rank matrices, and thus maps both users and POIs to a joint latent factor space with lower dimensionality [35].

(2) Single-criterion techniques
- Geographical Criterion (Geo): This method recommends POIs based on the single geographical criterion. It models users' geographical preferences as described in Section 4.1, and recommends POIs that achieve the highest values of $GeoRating_f$ in Eq. (12).
- Category Criterion (Cate): This method recommends POIs based on the single category criterion. It models users' category preferences as described in Section 4.2, and recommends POIs that achieve the highest values of $CateRating_f$ in Eq. (12).
- Attribute Criterion (Attri): This method recommends POIs based on the single attribute criterion. It models users' attribute preferences as described in Section 4.3, and recommends POIs that achieve the highest values of $AttriRating_f$ in Eq. (12).

(3) Multi-criteria based techniques
- Weighted sum (WeightedSum): This method fuses a user's ratings to each POI from the geographical, category and attribute criteria using weighted sum. For a fair comparison, the weight on each criterion is set to be the same as our iMCRec.
- Ensemble ranking (Ensemble) [18]: This method employs support vector machine (SVM) to learn query-dependent weights for combining multiple rankers in web search applications [18]. Here we adapt it to learn personalized weights on the three criteria for each user.
- Skyline (Skyline) [25]: This method uses Pareto dominance [9] to evaluate each POI based on the three criteria and selects the top-*N* representative skylines [25] as recommendations.

(4) Without personalized weighting
- iMCRec w/o PW: This method recommends POIs using our iMCRec framework without assigning personalized weights on the three criteria. Specifically, it assigns the same weights for all users, i.e., $w_g = 0.33$, $w_c = 0.33$ and $w_a = 0.33$ in Eqs. (16) and (17) for each user.

[1] https://www.yelp.com/dataset_challenge.



(a) NV (Precision) (b) NV (Recall) (c) AZ (Precision) (d) AZ (Recall)

Fig. 5. Comparison of non-multi-criteria techniques on top-*N*.

6.3. Performance metrics

In POI recommendation systems, it is important to find how many recommendations that actually visited by the target user in his/her testing dataset. For this purpose, two standard performance metrics, precision and recall, are adopted to measure the quality of recommendation results. The precision and recall are defined as follows:

$$\text{precision} = \frac{|\text{Positive POIs} \bigcap \text{Recommended POIs}|}{|\text{Recommended POIs}|},$$

$$\text{recall} = \frac{|\text{Positive POIs} \bigcap \text{Recommended POIs}|}{|\text{Positive POIs}|},$$

where the positive POIs are the POIs that have been visited by the target user in the testing dataset.

6.4. Parameter settings

The visited POIs of a user are divided into training and testing sets based on the visit timestamp. The training set contains the earlier 70% visited POIs, which is used to model the user's preferences and weights on the three criteria. The testing set, which contains the remaining 30% visited POIs, is considered as the ground truth.

To implement our personalized weight learning strategy, we split the training set into two parts using the commonly used ratio: 70% of the training set with earlier timestamp is used for the weight training and the rest 30% of the training set is used for weight testing. To search the optimal personalized weights, we set the searching step as 0.01 for two main reasons. On the one hand, a large step (e.g., 0.1) may lead to a coarse result. On the other hand, the testing recommendation quality is not sensitive to a too small searching step (e.g., 0.001). For example, given the searching step of 0.001, the weight $\boldsymbol{w}_1 = (0.800, 0.198, 0.002)$ and $\boldsymbol{w}_2 = (0.800, 0.199, 0.001)$ can achieve the same recommendation quality. Further, a smaller searching step will lead to a longer searching time.

7. Experimental results

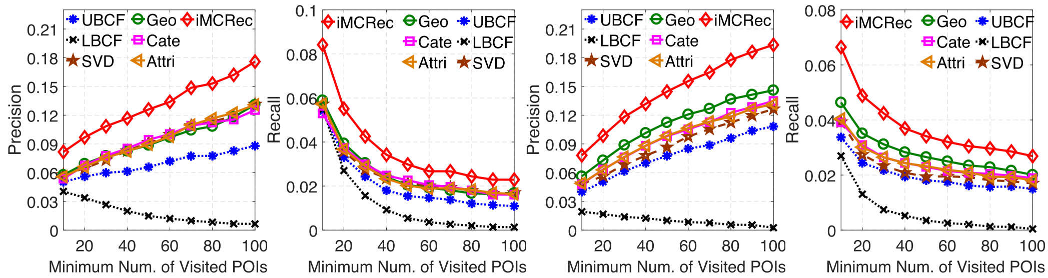
This section presents our extensive experimental results. We first compare our iMCRec against the state-of-the-art recommendation techniques in terms of the recommendation accuracy (Section 7.1), and then evaluate the efficiency of FastMCDM algorithm (Section 7.2).

7.1. Recommendation accuracy

The aim of this experiment is to study the effectiveness of our iMCRec in three-fold: (1) Effectiveness of integrating multiple criteria. We compare iMCRec with the no-criterion techniques: UBCF, LBCF and SVD, and the single criterion based techniques: Geo, Cate and Attri. (2) Effectiveness of FastMCDM. We compare iMCRec with the methods that employ weighted sum (WeightedSum), ensemble ranking (Ensemble) and skyline (Skyline) to integrate the geographical, category and attribute criteria. (3) Effectiveness of personalized weighting. We compare iMCRec with the learned personalized weights and iMCRec without the personalized weights. Further, the data sparsity problem is discussed.

7.1.1. Effectiveness of integrating multiple criteria

This experiment compares the recommendation accuracy of iMCRec with the no-criterion recommendation techniques: UBCF, LBCF and SVD, and the single-criterion methods: Geo, Cate and Attri.



(a) NV (Precision) (b) NV (Recall) (c) AZ (Precision) (d) AZ (Recall)

Fig. 6. Comparison of non-multi-criteria techniques on the minimum number of visited POIs.

Fig. 5 depicts the average precision and recall of the evaluated recommendation methods with respect to various numbers of recommended POIs, *N*, ranging from 2 to 10. As *N* increases, the precision gradually gets lower but the recall steadily becomes higher in both datasets. This is because with a larger number of recommendations, it is more easily to discover POIs that users would like to visit, leading to higher recall; however, the number of qualified recommendations is limited, resulting in lower precision.

Fig. 6 depicts the precision and recall of the evaluated methods when the minimum number of visited POIs increases from 10 to 100, in order to show how these recommendation techniques perform on both the cold-start users (i.e., users with a few visited POIs) and the active users (i.e., users with many visited POIs). For example, a measure at “Minimum Number of Visited POIs =10” is averaged on all users who have at least checked in 10 POIs in the training set. The raise of the number of visited POIs brings two effects: (1) more checking data for modeling user preference, resulting in higher precision; and (2) a larger number of positive POIs in the testing set, which leads to lower recall with the fixed number of recommended POIs. It is worth to note that LBCF has even lower precision with a larger number of visited POIs Figs. 6(a) and (c). This is because more check-in data cannot help LBCF model users' preferences well.

Nevertheless, in both Figs. 5 and 6, iMCRec performs better than the baseline methods in terms of both precision and recall. The comparison details are as follows.

UBCF and LBCF perform worse than the other methods as depicted in Figs. 5 and 6. There are two reasons: (1) In both methods, POIs are recommended only based on other users' opinions, no personalized preferences are considered. (2) The low precision and recall are also caused by the sparsity of our datasets. The average numbers of visited POIs per user are only about 25 and 28 in two datasets, and the average number of check-in records per POI is even smaller, which explains why LBCF performs much worse than UBCF.

SVD alleviates the data sparsity problem by mapping both users and POIs to a joint latent factor space with lower dimensionality. So it performs better than UBCF and LBCF, as shown in Figs. 5 and 6. However, the improvement is considerably limited since none of the personalized preferences is considered.

Geo models geographical preference for each user using two-dimensional kernel density estimation as described in Section 4.1. It improves the recommendation precision and recall in comparison to the no-criterion techniques UBCF, LBCF and SVD. Actually, it achieves the second best precision and recall in AZ dataset, as shown in Figs. 5(c), (d), 6(c) and (d). However, the improvement is also limited because users' preferences have more than one criterion, and the geographical criterion is not enough for modeling users' preferences.

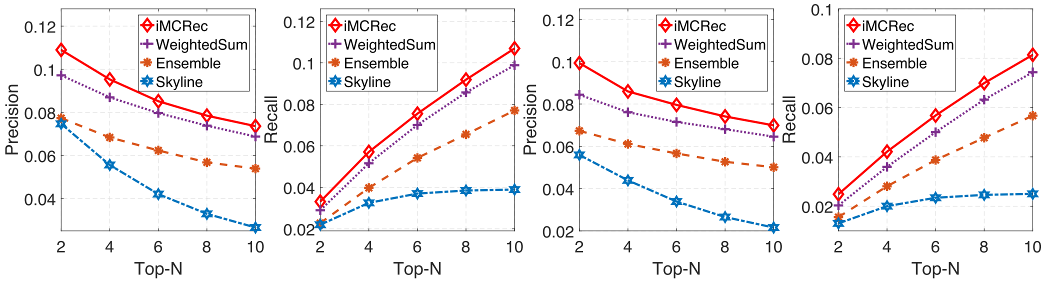
Cate models category preference for each user using the hierarchical TF-IDF tree as described in Section 4.2. Since the category criterion is not enough for modeling users' preferences, it only performs slightly better than the no-criterion techniques, as shown in Figs. 5(a), (c), (d), 6(c) and (d). Furthermore, it may not be able to overcome the sparsity problem, as it performs no better than SVD according to Figs. 5(b), 6(a) and (b).

Attri models attribute preference for each user using the TF-IDF and entropy techniques as described in Section 4.3. According to Figs. 5 and 6, this method also improves the recommendation accuracy comparing to UBCF and LBCF. This demonstrates that users' preferences on POIs are also affected by the attributes of POIs. So it is necessary to consider the attribute criterion for the recommendation.

Our iMCRec models users' preferences based on the three criteria and integrates them with personalized weights using FastMCDM algorithm. It always achieves the best recommendation quality in terms of both precision and recall. These results verify the superiority of integrating multiple criteria for recommendation.

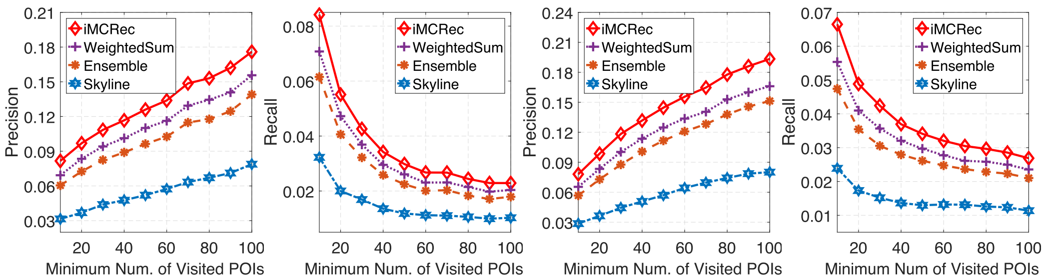
7.1.2. Effectiveness of FastMCDM

To demonstrate the effectiveness of our FastMCDM algorithm in integrating the geographical, category and attribute criteria, we compare iMCRec with the methods that employ weighted sum (WeightedSum), ensemble ranking (Ensemble) and skyline (Skyline) to integrate the three criteria for recommendation.



(a) NV (Precision) (b) NV (Recall) (c) AZ (Precision) (d) AZ (Recall)

Fig. 7. Comparison of Multi-criteria techniques on top-*N*.



(a) NV (Precision) (b) NV (Recall) (c) AZ (Precision) (d) AZ (Recall)

Fig. 8. Comparison of multi-criteria techniques on the minimum number of visited POIs.

Fig. 7 plots the average precision and recall of all the multi-criteria based methods with respect to various numbers *N* of recommended POIs on NV and AZ datasets. As *N* increases, the precision gets lower but recall becomes higher. Fig. 8 plots the average precision and recall of all the multi-criteria based methods with respect to the increase of the minimum number of visited POIs on the two datasets. As the number of the visited POIs gets larger, the precision generally increases but the recall usually decreases. From the two figures, we can observe that our iMCRec outperforms these three methods. The comparison details are as follows.

WeightedSum performs still worse than iMCRec in both Figs. 7 and 8 under the same setting of personalized weights. This is because it suffers from the scale problem: the ratings in different criteria are in different scales since they are estimated by different approaches. The simple summation would weaken the effect of the assigned weights.

Ensemble employs SVM to learn personalized weights on the three criteria for each user. However, it performs even worse than WeightedSum, according to Figs. 7 and 8. This is because this weight learning process depends on the ratings (scores) rather than orders of POIs in the three criteria. It still has the scale problem which makes its weight learning process less effective.

Skyline does not have the scale problem since it uses Pareto dominance to evaluate each alternative [9]. However, it treats each criterion in the same way, i.e., no personalized weight is assigned to each criterion, which makes the performance of this approach the worst, as depicted in Figs. 7 and 8.

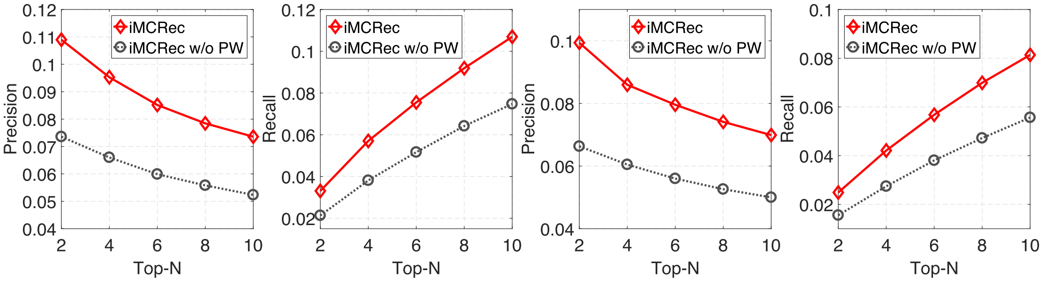
In summary, the relatively high precision and recall achieved by our iMCRec demonstrate its flexibility to the scale problem and its capability of assigning personalized weight to each criterion.

7.1.3. *Effectiveness of personalized weighting*

In this section, we evaluate the effectiveness of our weight learning strategy proposed in Section 5.3 by comparing the recommendation accuracy of our iMCRec with the learned personalized weights (denoted as iMCRec), and iMCRec without the personalized weights (denoted as iMCRec w/o PW).

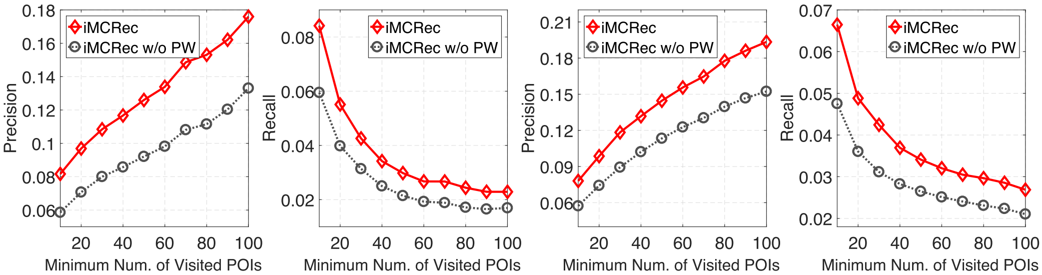
Note that in iMCRec, some of the users cannot obtain their optimal weight vectors during the weight learning process, as they cannot get good recommendations that will be discovered in their weight testing sets for all input weight vectors. In this case, we assign the most popular weight vector to them. The most popular weight vectors obtained in our experiments are ($w_g = 0.10,\ w_c = 0.15,\ w_a = 0.75$) and ($w_g = 0.72,\ w_c = 0.12,\ w_a = 0.16$) in NV and AZ datasets, respectively. In iMCRec w/o PW, the same weights are assigned to the geographical, category, and attribute criteria. Namely, $w_g = 0.33,\ w_c = 0.33$ and $w_a = 0.33$ are set for all the users.

Figs. 9 and 10 depict the average precision and recall of the two methods, with respect to the change of the number of recommended POIs and the increase of the minimum number of visited POIs on the two datasets. By adopting the



(a) NV (Precision) (b) NV (Recall) (c) AZ (Precision) (d) AZ (Recall)

Fig. 9. Comparison of non-personalized weighting techniques on top-*N*.



(a) NV (Precision) (b) NV (Recall) (c) AZ (Precision) (d) AZ (Recall)

Fig. 10. Comparison of non-personalized weighting techniques on the minimum number of visited POIs.

Table 5
Efficiency improvement by alternative filtering (AF).

State	Avg. number of input alternatives		Avg. CPU running time (ms)	
	MCDM	FastMCDM	MCDM	FastMCDM
Nevada	19,389	97.12	32.1	1.2
Arizona	29,410	114.91	48.4	1.7

personalized weights, iMCRec outperforms iMCRec w/o PW that assigns a universal and equal weight for each criterion. This demonstrates the effectiveness of our weight learning strategy.

7.1.4. Discussion on the data sparsity

In general, the accuracy of all recommendation techniques for LBSNs is not high due to the data sparsity problem, i.e., the density of user-POI check-in matrix is pretty low. For example, in [20], the reported best performance of their recommendation approach is 0.03 in terms of precision when recommending 10 POIs, (i.e., N=10), and their user-POI check-in matrix has a sparsity of 99.87%, which is 1.3×10^{-3} in terms of the matrix density. Similarly, in [41], the reported best precision of recommending 10 POIs is lower than 0.025 over its two datasets with density of 6.35×10^{-3} and 9.85×10^{-4}, respectively.

In our datasets, the user-POI check-in matrix densities of the two datasets are also pretty low, 1.3×10^{-3} in Nevada dataset, and 9.5×10^{-4} in Arizona dataset, as shown in Table 4. So the relatively low precision and recall values are common and reasonable in the experiments. Instead, we focus on the relative accuracy improvement of our method compared with the baseline methods. As we have concluded from Figs. 5–10, our iMCRec outperforms all the baseline methods. Furthermore, as shown in Figs. 6, 8 and 10, the recommendation precision achieved by iMCRec improves a lot with a larger number of visited POIs, which means that our method would have better performance given a dataset with higher density.

7.2. Efficiency of the alternative filtering step

In this section, we analyze the efficiency of our FastMCDM algorithm proposed in Section 5.2.2. In FastMCDM, impossible alternatives are pruned before generating the top-*N* recommendations, to avoid unnecessary MCDM-based rating calculations. To evaluate its efficiency, we compare the number of input alternatives as well as CPU running time between FastMCDM algorithm and the basic MCDM algorithm. Both algorithms are implemented with Matlab 9.0. The experiments are conducted on a computer with Intel CPU i7-4470, 16GB memory and Windows 7 system.

Table 5 gives the average number of alternatives per user and averaged CPU running time for both algorithms. The number of recommended POIs is set to 10 (i.e., $N = 10$). This table shows that FastMCDM with the alternative filtering step can greatly reduce the number of input alternatives, and thus, reduce the CPU running time.

8. Conclusion and future work

In this paper, we proposed a new POI recommendation framework called iMCRec by considering user preferences on geographical, category and attribute criteria. In iMCRec, preference models are first built for individual user's geographical, category, and attribute preferences. Especially, we proposed the attribute preference model by considering both preferences on different values of each attribute and importance of different attributes. A sophisticated collaborative filtering method is developed under each of the three criteria. We also proposed a learning strategy to learn the personalized weights on the three criteria. A fast multi-criteria decision making algorithm, called FastMCDM, is prosed to integrate the three criteria with the personalized weights and efficiently generate top-N POIs as recommendations. Finally, experimental results on two real-world datasets from Yelp show that iMCRec outperforms the state-of-the-art POI recommendation techniques and other multi-criteria approaches.

We have two promising research directions for future study: (1) Modeling the temporal change of a user's preferences. For example, the POIs that were visited by a user long time ago would be less indicative on the user's preferences than the recently visited ones. How to incorporate the temporal influence into our preference models would be a promising research direction. (2) Leveraging the reviews of users commenting on POIs. The user reviews on POIs contain valuable information indicating the users' preferences. How to integrate the information from user reviews into our preference model would be another promising research direction.

References

[1] S. Aciar, D. Zhang, S. Simoff, J. Debenham, Informed recommender: basing recommendations on consumer product reviews, IEEE Intell. Syst. 22 (3) (2007) 39–47.
[2] G. Adomavicius, Y. Kwon, New recommendation techniques for multicriteria rating systems, IEEE Intell. Syst. 22 (3) (2007) 48–55.
[3] H.J. Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, Inf. Sci. 178 (1) (2008) 37–51.
[4] R. Bader, E. Neufeld, W. Woerndl, V. Prinz, Context-aware poi recommendations in an automotive scenario using multi-criteria decision making methods, ACM CaRR, 2011.
[5] J. Bao, Y. Zheng, M.F. Mokbel, Location-based and preference-aware recommendation using sparse geo-social networking data, ACM SIGSPATIAL, 2012.
[6] G. Campanella, R.A. Ribeiro, A framework for dynamic multiple-criteria decision making, Decis. Support Syst. 52 (1) (2011) 52–60.
[7] C. Cheng, H. Yang, I. King, M.R. Lyu, Fused matrix factorization with geographical and social influence in location-based social networks, AAAI, 2012.
[8] S.H. Choi, S. Kang, Y.J. Jeon, Personalized recommendation system based on product specification values, Expert Syst. Appl. 31 (3) (2006) 607–616.
[9] J. Chomicki, P. Ciaccia, N. Meneghetti, Skyline queries, front and back, ACM SIGMOD Record 42 (3) (2013) 6–18.
[10] P. Cremonesi, Y. Koren, R. Turrin, Performance of recommender algorithms on top-n recommendation tasks, in: ACM RecSys, 2010, pp. 39–46.
[11] J. Díez, J.J. Del Coz, O. Luaces, A. Bahamonde, Clustering people according to their preference criteria, Expert Syst. Appl. 34 (2) (2008) 1274–1284.
[12] J.S. Dyer, P.C. Fishburn, R.E. Steuer, J. Wallenius, S. Zionts, Multiple criteria decision making, multiattribute utility theory: the next ten years, Manage. Sci. 38 (5) (1992) 645–654.
[13] A. Emrich, A. Chapko, D. Werth, P. Loos, Adaptive, multi-criteria recommendations for location-based services, HICSS, 2013.
[14] J. Figueira, S. Greco, M. Ehrgott, Multiple Criteria Decision analysis: State of the Art Surveys, 78, Springer, New York, 2005.
[15] S. Garcia Esparza, M.P. O'Mahony, B. Smyth, A multi-criteria evaluation of a user generated content based recommender system, ACM RecSys, 2011.
[16] H.B. Hashemi, N. Yazdani, A. Shakery, M.P. Naeini, Application of ensemble models in web ranking, in: Telecommunications (IST), 2010 5th International Symposium on, 2010, pp. 726–731.
[17] J.L. Herlocker, J.A. Konstan, A. Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering, ACM SIGIR, 1999.
[18] S.C. Hoi, R. Jin, Semi-supervised ensemble ranking, in: AAAI, 2008, pp. 634–639.
[19] B. Jeong, J. Lee, H. Cho, Improving memory-based collaborative filtering via similarity updating and prediction modulation, Inf. Sci. 180 (5) (2010) 602–612.
[20] B. Liu, Y. Fu, Z. Yao, H. Xiong, Learning geographical preferences for point-of-interest recommendation, ACM SIGKDD, 2013.
[21] L. Liu, N. Mehandjiev, D.-L. Xu, Multi-criteria service recommendation based on user criteria preferences, in: ACM RecSys, 2011, pp. 77–84.
[22] X. Liu, Y. Liu, K. Aberer, C. Miao, Personalized point-of-interest recommendation by mining users' preference transition, ACM CIKM, 2013.
[23] Y. Liu, Z.-P. Fan, Y. Zhang, A method for stochastic multiple criteria decision making based on dominance degrees, Inf. Sci. 181 (19) (2011) 4139–4153.
[24] P. Luarn, J.-C. Yang, Y.-P. Chiu, Why people check in to social network sites, Int. J. Electron. Commerce 19 (4) (2015) 21–46.
[25] M. Magnani, I. Assent, M. Mortensen, Taking the big picture: representative skylines based on significance and diversity, VLDB J. 23 (5) (2014) 795–815.
[26] N. Manouselis, C. Costopoulou, Experimental analysis of design choices in multiattribute utility collaborative filtering, IJPRAI 21 (2) (2007) 311–331.
[27] M. Morse, J. Patel, W. Grosky, Efficient continuous skyline computation, Inf. Sci. 177 (17) (2007) 3411–3437.
[28] M.J. Pazzani, D. Billsus, Content-based recommendation systems, The Adaptive Web, 2007.
[29] T. Pontes, G. Magno, M. Vasconcelos, A. Gupta, J. Almeida, P. Kumaraguru, V. Almeida, Beware of what you share: Inferring home location in social networks, in: Proceedings of IEEE International Conference on Data Mining Workshops, 2012, pp. 571–578.
[30] D. Ramendra Singh, R. Yavuz, A. Toker, Location sharing on social networks: implications for marketing, Market. Intell. Plann. 32 (5) (2014) 567–585.
[31] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, WWW, 2001.
[32] Y.-Y. Shih, D.-R. Liu, Product recommendation approaches: collaborative filtering via customer lifetime value and customer demands, Expert Syst. Appl. 35 (1) (2008) 350–360.
[33] B.W. Silverman, Density Estimation for Statistics and Data Analysis, Chapman and Hall, 1986.
[34] A. Töscher, M. Jahrer, Collaborative filtering ensemble for ranking, J. Mach. Learn. Res. W&CP 18 (2012) 61–74.
[35] M. Vozalis, K. Margaritis, Using SVD and demographic data for the enhancement of generalized collaborative filtering, Inf. Sci. 177 (15) (2007) 3017–3037.
[36] I. Wald, V. Havran, On building fast kd-trees for ray tracing, and on doing that in O(N log N), in: IEEE Symposium on Interactive Ray Tracing, 2006, pp. 61–69.
[37] R. Wang, S. Kwong, Active learning with multi-criteria decision making systems, Pattern Recognit. 47 (9) (2014) 3106–3119.
[38] X. Xu, J.-M. Martel, B.F. Lamond, A multiple criteria ranking procedure based on distance between partial preorders, Eur. J. Oper. Res. 133 (1) (2001) 69–80.
[39] M. Ye, P. Yin, W.-C. Lee, D.-L. Lee, Exploiting geographical influence for collaborative point-of-interest recommendation, ACM SIGIR, 2011.

[40] J.J.-C. Ying, E.H.-C. Lu, W.-N. Kuo, V.S. Tseng, Urban point-of-interest recommendation by mining user check-in behaviors, ACM SIGKDD, 2012.
[41] Q. Yuan, G. Cong, Z. Ma, A. Sun, N.M. Thalmann, Time-aware point-of-interest recommendation, ACM SIGIR, 2013.
[42] J. Zhang, C.-Y. Chow, iGSLR: personalized geo-social location recommendation: a kernel density estimation approach, ACM SIGSPATIAL, 2013.
[43] J. Zhang, C.-Y. Chow, CoRe: exploiting the personalized influence of two-dimensional geographic coordinates for location recommendations, Inf. Sci. 293 (2015) 163–181.
[44] J. Zhang, C.-Y. Chow, Y. Li, iGeoRec: a personalized and efficient geographical location recommendation framework, Serv. Comput. 8 (5) (2015) 701–714.