This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2789445, IEEE Transactions on Knowledge and Data Engineering

Scalable Content-Aware Collaborative Filtering for Location Recommendation

Defu Lian, Yong Ge, Fuzheng Zhang, Nicholas Jing Yuan, Xing Xie, Tao Zhou, and Yong Rui

Abstract—Location recommendation plays an essential role in helping people find attractive places. Though recent research has studied how to recommend locations with social and geographical information, few of them addressed the cold-start problem of new users. Because mobility records are often shared on social networks, semantic information can be leveraged to tackle this challenge. A typical method is to feed them into explicit-feedback-based content-aware collaborative filtering, but they require drawing negative samples for better learning performance, as users' negative preference is not observable in human mobility. However, prior studies have empirically shown sampling-based methods do not perform well. To this end, we propose a scalable Implicit-feedback-based Content-aware Collaborative Filtering (ICCF) framework to incorporate semantic content and to steer clear of negative sampling. We then develop an efficient optimization algorithm, scaling linearly with data size and feature size, and quadratically with the dimension of latent space. We further establish its relationship with graph Laplacian regularized matrix factorization. Finally, we evaluate ICCF with a large-scale LBSN dataset in which users have profiles and textual content. The results show that ICCF outperforms several competing baselines, and that user information is not only effective for improving recommendations but also coping with cold-start scenarios.

Index Terms—Implicit feedback; Content-aware; Location recommendation; Weighted matrix factorization

1 Introduction

As cities develop, the growing number of locations of interest, such as hotels, attractions, and restaurants, offer people more opportunities for amusement than ever before. At the same time, since novelty seeking is regarded as a basic requirement for human activity [2], people really enjoy exploring neighborhoods and visiting locations tailored to their interests. As a result, location recommendation has been exploited to help people discover interesting places [3], [4] and speed up users' familiarization with their surroundings.

The advent of location-based social networks (LBSNs), such as Foursquare, Jiepang, and Yelp, makes it possible to analyze large-scale human mobility data, creating business opportunities for mobile advertising [5]. With the support of massive data, location recommendation has recently become a popular research topic. Prior research has mainly investigated how to leverage spatial patterns [4], [6], temporal effects [7], [8], spatio-temporal influence [9], social influence [10], text-based analysis [11], [12], and implicit characteristics of human mobility [13], [14], [15] to recommend locations. However, some of these methods require each user to have sufficient training data while others assume locations have accumulated ample textual information (e.g., tips), making it challenging to use them to tackle the cold-start problem, specifically, recommending locations for new users. Fortunately, users are often linked to social networks, such as Twitter and Weibo, which probably collect rich semantic content from users. This semantic content is likely to imply user interest, an essential element for capturing users' visiting behavior [16]. Therefore, they can be exploited to address the cold-start challenge and even improve location recommendation. A typical method is to feed them into traditional explicit-feedback content-aware recommendation frameworks, such as LibFM [17], SVDFeature [18], regression-based latent factor model [19] or MatchBox [20]. These frameworks require drawing negative samples from unvisited locations for better learning performance, since a user's negative preference for locations is not observable in human mobility data. However, it has been empirically shown that sampling-based frameworks do not perform as well as an algorithm that treats all unvisited locations as negative yet assigns them a lower preference confidence [13], [15], since the latter one deals with the sparsity issues better.

With this in mind, we propose a novel scalable Implicit-feedback based Content-aware Collaborative Filtering (ICCF) framework. It steers clear of sampling negative locations, by treating all unvisited locations as negative and proposing a sparse and rank-one weighting configuration for modeling preference confidence. This sparse and rank-one weighting configuration not only assigns vastly varying confidence to visited and unvisited locations, but also subsumes three previously developed different weighting schemes for unvisited locations and naturally introduces a novel mixed weighting scheme. ICCF takes a user-location preference matrix, a user-feature matrix (e.g., gender, age and tweets) and a location-feature matrix (e.g., categories, descriptions and neighborhood) as input, and maps each user, each location and their features onto a joint latent

- *Defu Lian and Tao Zhou are with School of Computer Science and Engineering, and Big Data Research Center, University of Electronic Science and Technology of China. E-mail: {dove.ustc,zhutouster}@gmail.com*
- *Yong Ge is with Eller School of Management, The University of Arizona. Email: yongge@email.arizona.edu*
- *Fuzheng Zhang, Xing Xie and Yong Rui are with Microsoft Research, Beijing, China. Email: {fuzzhang,xingx,yongrui}@microsoft.com*
- *Nicholas Jing Yuan is with Microsoft Corporation. Email: nichy@microsoft.com*

This article is an expanded version of [1], which appeared in Proceedings of 2015 IEEE International Conference on Data Mining (ICDM).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2789445, IEEE Transactions on Knowledge and Data Engineering

space, such that the dot product between two objects defines a preference score. For example, the dot product between a user's latent factor and a category's (e.g., restaurant) latent factor indicates a preference score of the user for the category. Thanks to the availability of user/location features, ICCF not only improves location recommendation, but also addresses the cold-start problems of both new users and new locations. To achieve the mapping procedure, we develop a novel variable substitution technique to split the learning of ICCF into two weighted least square problems with respect to user/location latent factors, and two (sparse) multiple dependent-variable regression problems with respect to feature latent factor matrices. To learn user/location latent factors in weighted least square problems, we propose coordinate descent for optimization, which scales linearly with data size and feature size, and quadratically with the dimension of latent space. Without any adjustment to the algorithm, we can easily determine whether to include user/location bias or not by augmenting user/location latent matrix with either an all-one vector or an all-zero vector. The incorporation of user/location bias can further help to deal with the sparsity issues, according to empirical studies. To learn feature latent matrices in multiple dependent-variable regression problems, we extend conjugate gradient descent to matrix variable cases, which scales linearly with feature size, i.e., the number of non-zero entries in the user/location feature matrices.

Through analysis of ICCF, we establish its close relationship with graph Laplacian regularized matrix factorization [21], and offer a good explanation of the proposed algorithm, such that user (location) features refine the similarity between users (locations) on implicit feedback. Therefore, ICCF not only becomes an alternative solution for similarity constrained matrix factorization algorithms, but also can be incorporated with domain-specific knowledge, such as document similarity between user tweets (e.g., vector space model), and age proximity between users.

We then apply ICCF for location recommendation based on human mobility data of over 18M visit records of 265K users obtained from a location-based social network. In this dataset, locations have two levels of categories and geographical information, while users have profile information (e.g., gender and age) and rich semantic content (e.g., tweets and tags) crawled from a social network. Based on the evaluation results of 5-fold cross validation on mobility data, corresponding to the warm-start case, we observe that ICCF is superior to five competing baselines. This implies the effectiveness of information incorporation and parameter learning as well as sparse and rank-one weighting configurations. In addition, based on this evaluation, we find that user profiles and semantic content can make significant improvements over the counterpart without taking them into account. In addition to the warm-start evaluation, we also perform a cold-start evaluation with a user-based 5-fold cross validation by splitting users into five non-overlapping groups. The results indicate that both user profiles and semantic content are useful for tackling the cold-start problem in location recommendation based on human mobility data, and that user profiles are more effective than semantic content.

This paper is an extension of our previous paper [1], which proposed implicit feedback based content-aware collaborative filtering for location recommendation. In this paper, we further deliver the following contributions:

- We extend implicit feedback based collaborative filtering through a sparse and rank-one weighting scheme, thus it subsumes three existing weighting schemes for modeling negative preference and naturally introduces a novel mixed weighting method. The effectiveness of the proposed sparse and rank-one weighting schemes has been extensively evaluated, showing its significant benefit for improving recommendation, in particular for locations at long tails.
- We propose an efficient coordinate descent optimization algorithm to learn parameters in the sparse and rank-one weighting schemes, which scales linearly with data size and feature size, and quadratically with the dimension of latent space. In addition to theoretical analysis of time complexity, we empirically study convergence and efficiency issues in the proposed optimization algorithm.
- We investigate how to incorporate biases into content-aware matrix factorization without any adjustment to the optimization algorithm, achieved by how to augment latent factors of users and locations. We also empirically study the effects of biases, and observe their significant benefit to recommendation from sparse datasets.
- We elaborate on the procedure for establishing close relationship of the proposed model with a graph Laplacian regularized matrix factorization.

2 Related Work

We propose an efficient content-aware collaborative filtering framework for location recommendations based on human mobility data. Therefore, related work consists of location recommendation and content-aware collaborative filtering.

Location recommendation has been an important topic in location-based services. From the perspective of types of recommended items, some prior research focuses on recommending specific types of locations while others are generalized for any type of locations. For example, Horozov et al. [22] have developed a user-based collaborative filtering system to recommend restaurants to a user. Zheng et al. [23] design a random walk style model for tourism hot spot recommendation. Zheng et al. consider location recommendation and activity recommendation together, so that they can provide location recommendation with respect to different types of activities [3], [24]. Ye et al. study how to jointly exploit geographical influence and collaborative filtering for recommending points of interest (of any category) given large scale mobility records from location-based social networks [4]. Following this, more sophisticated models, such as jointly modeling geographical and social influence, and performing model-based collaborative filtering such as matrix factorization [6], [13], [14], [25], tensor factorization [26], [27], and word embedding techniques [28], have been proposed with the aim of seamless integration. In addition to geographical information, textual content is also associated with many locations, since users often leave

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2789445, IEEE Transactions on Knowledge and Data Engineering

comments about venues on location-based social networks after visiting. Thus, some prior works exploit this content by topic modeling [12] and sentiment analysis [11], [29], incorporating these text modeling techniques with collaborative filtering via collective matrix factorization, preference matrix refinement, regularization or empirical linear combination.

In contrast to these methods, we mainly study the effects of user information instead of location information on recommendation. User information should be more important than location information when addressing the cold start problem since it is available earlier for inferring user interest. Additionally, we propose a general framework for location recommendation based on human mobility data, which can incorporate any features without a deep understanding of the factorization model. Such an objective is difficult to satisfy in prior works since the incorporation of any other feature requires expert knowledge to modify the learning procedure. Furthermore, prior works do not take all the characteristics of implicit feedback into account, and most of them require sampling negatively preferred locations from unvisited ones.

Content-aware collaborative filtering is the integration of content-based recommendation and collaborative filtering. In recent years, several general algorithms, including the regression-based latent factor model [19], LibFM [17], MatchBox [20], and SVDFeature [18], have been proposed. These algorithms are almost equivalent to each other in model representation but different in terms of optimization algorithms. For example, the first two algorithms make use of sampling methods for inferring latent factors while MatchBox leverages approximate deterministic approaches for inference. Among prior research works, some methods have been implemented in open-source frameworks and widely used in many applications, such as music recommendation in KDDCup 2011 and friendship prediction in KDDCup 2012. However, it seems that they do not work well in the Million Song Dataset Challenge [30] due to extreme sparsity (0.01% density). In addition to general algorithms taking different kinds of content, specific algorithms with textual content of items have also been proposed, such as collaborative topic regression [31]. They have been exploited for tasks like news recommendation and scientific article discovery.

Despite their wide use, these algorithms are mainly designed for explicit feedback with both positively and negatively preferred samples, and optimized only over non-zero entries from user-item rating matrices. The time complexity is only in linear proportion to the number of non-zero entries in the rating matrices. However, due to only positively preferred items being provided in implicit feedback, feeding them together with user/item information into these existing frameworks requires drawing a comparable number of negatively preferred items with the positive ones for the sake of efficiency. This may incur sub-optimal recommendation performance. In contrast, our proposed algorithm targets content-aware collaborative filtering from implicit feedback and successfully addresses the disadvantages by treating the items not preferred by users as negative while assigning them a lower confidence for negative preference, and achieving linear time optimization.

3 Implicit Feedback based Collaborative Filtering

Given mobility data of M users visiting N locations, location recommendation first converts it into a user-location frequency matrix $\mathbf{C} \in \mathbb{N}^{M\times N}$, with each entry $c_{u,i}$ indicating the visit frequency of a user u to a location i. $\mathbf{R} \in \{0,1\}^{M\times N}$ is a preference matrix, for which each entry $r_{u,i}$ is set to 1 if the user u has visited the location i; otherwise it is set to zero. In the following, upper case bold letters denote matrices, lower case bold letters denote column vectors, and non-bold letters represent scalars.

Given the frequency matrix $\mathbf{C}$, although recommendation algorithms such as Bayesian Personalized Ranking based Matrix Factorization (BPRMF) [32], [33], Weighted Approximate-Rank Pairwise (WARP) [26], [27], and Bayesian non-negative matrix factorization [6], [34] have been exploited for recommending locations, they are still not comparable to weighted matrix factorization, a superior One Class Collaborative Filtering (OCCF) algorithm [35], [36], as shown in prior studies [13], [15], [37]. This will be validated in our experiments on two other location-based social network datasets. Weighted matrix factorization, being performed on the preference matrix $\mathbf{R}$, maps both users and locations into a joint latent space of $K \ll \min(M,N)$ dimension, where each user and each location is represented by user latent factor $\mathbf{p}_u$ and location latent factor $\mathbf{q_i}$, respectively, and the preference $r_{u,i}$ of a user u for a location i is estimated as the inner product between their latent factors, that is,

$$\hat{r}_{u,i} = \mathbf{p}'_u \mathbf{q}_i. \tag{1}$$

This representation is easily extended to include user bias b_u and location bias b_i. In particular, $\mathbf{p}'_u\mathbf{q}_i + b_u + b_i = [\mathbf{p}_u; b_u; 1]'[\mathbf{q}_i; 1; b_i]$, where $[;]$ is a row-based concatenation operator so that $[\mathbf{p}_u; b_u; 1]$ is a column vector of length $K+2$. This flexibility provides great convenience for learning biases in content-aware matrix factorization.

To learn user/location latent factor together with their biases, weighted matrix factorization optimizes the following objective function [36],

$$\mathscr{L}_{WMF} = \frac{1}{2}\sum_{u,i} w_{u,i}(r_{u,i} - [\mathbf{p}_u; b_u; 1]'[\mathbf{q}_i; 1; b_i])^2 + \frac{\lambda_1}{2}\sum_u \|[\mathbf{p}_u; b_u]\|^2 + \frac{\lambda_2}{2}\sum_i \|[\mathbf{q}_i; b_i]\|^2, \tag{2}$$

where $w_{u,i}$ is an entry of a weighting matrix $\mathbf{W} \in \mathbb{R}^{M\times N}$, determining the confidence of preference. λ_1 and λ_2 control the extent of regularization, to prevent over-fitting.

3.1 Sparse and One-Rank Weighting Configuration

In a mobility dataset, a user's visit to a location only implies her positive preference, thus her visited locations are considered positive examples and the visit frequency to locations determines the confidence level of positive preference. However, since their negative preference for unvisited locations has not explicitly been observed, all unvisited locations

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2789445, IEEE Transactions on Knowledge and Data Engineering

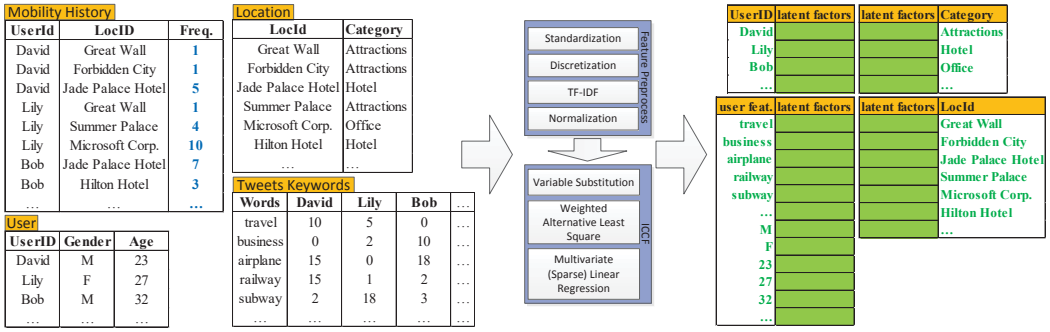


Fig. 1: The framework of ICCF.

are considered "pseudo" negative and the confidence in the negative attitude of unvisited locations is significantly less than the positive attitude of visited locations. Weighted matrix factorization, according to Eq (2), captures these two characteristics by assuming that the confidence level for positive preference increases with frequency, and treating unvisited locations as "pseudo" negative samples but assigning a significant lower confidence to them. However, because this objective function sums over all entries of the preference matrix, the weighting matrix should be carefully designed.

One efficient way to do so, as suggested in [36], is to assign the confidence level of the preference for negative locations to the same value, e.g., 1, as follows:

$$w_{u,i} = \begin{cases} \alpha(c_{u,i}) + 1 & \text{if } c_{u,i} > 0 \\ 1 & \text{otherwise}, \end{cases} \tag{3}$$

where $\alpha(c_{u,i})$ is a monotonically increasing function with respect to $c_{u,i}$ so that positive confidence increases with visit frequency. In this setting, the weighting matrix has a special structure, being an addition of a sparse matrix $\tilde{\mathbf{W}}$ such that $[\tilde{\mathbf{W}}]_{u,i} = \alpha(c_{u,i})$, and a rank-one matrix $\mathbf{1}_M \mathbf{1}'_N$, where $\mathbf{1}_M$ is an all-one vector of length M such that $\mathbf{1}_M \mathbf{1}'_N$ is a rank-one matrix of $M \times N$ whose only singular value is $\sqrt{MN}$. However, this assumption that the confidence for all unvisited locations with respect to a given user is the same is too strong. For example, since popular locations attract users' attention more easily, it is more likely for unvisited but popular locations to be negatively preferred. Therefore, we relax this assumption to consider a more generic sparse and rank-one weighting configuration with the guarantee of reserving efficiency. In particular, the weighting matrix is represented as follows:

$$\mathbf{W} = \tilde{\mathbf{W}} + \mathbf{a}\mathbf{d}', \tag{4}$$

where $\mathbf{a} \in \mathbb{R}_+^M$ and $\mathbf{d} \in \mathbb{R}_+^N$. If each element d_i of $\mathbf{d}$ represents the popularity of the location i and each element a_u of $\mathbf{a}$ indicates the corresponding user's activeness, weighting coefficients for popular but unvisited locations by an active user are higher. In other words, these popular locations are more likely to be negatively preferred with respect to active users. When $\mathbf{a} = \mathbf{0}$ or $\mathbf{d} = \mathbf{0}$, we can recover the basic matrix factorization algorithm [38]. More importantly, such a setting for the weighting matrix subsumes three negative sampling schemes proposed in [35] based on different assignments to $\mathbf{a}$ and $\mathbf{d}$. In particular, the setting $\mathbf{a} = \mathbf{1}_M$ and $\mathbf{d} = \mathbf{1}_N$ corresponds to a uniform weighting scheme; the setting $\mathbf{a} = \mathbf{1}_M$ and $d_i \propto M - \sum_u r_{u,i}$ shares common with an item-oriented weighting scheme, which assumes that if an item (location) has fewer positive examples, the missing entries for this item (location) are negative with a higher probability; the setting $a_u \propto \sum_i r_{u,i}$ and $\mathbf{d} = \mathbf{1}_N$ is similar to a user-oriented weighting scheme, which posits that if a user has more positive examples, it is more likely that she does not like the other items. A similar algorithm to ours but only providing the popularity-based weighting scheme has been proposed in [39]. Their assumption, assuming that popular items are probable to be truly negative, is contrary to the item-oriented weighting in [35]. These two cases will be compared in the experiments in order to seek a better weighting strategy. In addition to providing both user-oriented and item-oriented weighting schemes, our framework also provides a mixed user-item-orienting weighting scheme. Below, we dub this model (with the sparse and rank-one weighting configuration) Implicit-feedback based Collaborative Filtering (ICF) algorithm.

3.2 Coordinate Descent for Parameter Learning

The weighting matrix is configured as sparse and rank-one for the sake of efficient optimization of Eq (2). Below we elaborate the efficient optimization procedure. Using simple algebra, we find that it is possible to compute the gradient of $\mathscr{L}_{WMF}$ with respect to each $\mathbf{p}_u$ and $\mathbf{q}_i$, or the derivative of $\mathscr{L}_{WMF}$ with respect to each entry $p_{u,k}$ of $\mathbf{p}_u$ and each entry $q_{i,k}$ of $\mathbf{q}_i$. Based on the former gradient, an alternating least square algorithm can be developed; based on the latter derivative, a coordinate descent algorithm [39] can be adapted. In the following, we introduce how coordinate descent can be adaptive to this case. The approach for leveraging alternating least square is explained in the Appendix.

Coordinate descent takes turns optimizing $\mathscr{L}_{WMF}$ with respect to one coordinate $p_{u,k}$ of the latent factor $[\mathbf{p}_u; b_u]$ with user bias or one coordinate $q_{i,k}$ of the latent factor $[\mathbf{q}_i; b_i]$ with location bias given other fixed. Since $\mathscr{L}_{WMF}$ is quadratic with respect to $p_{u,k}$ given other fixed, setting the derivative of $\mathscr{L}_{WMF}$ with respect to $p_{u,k}$ to zero, we obtain the closed form for updating $p_{u,k}$,

$$p^*_{u,k} = \frac{\sum_i w_{u,i}(e_{u,i} + p_{u,k}q_{i,k})q_{i,k}}{\sum_i w_{u,i}q^2_{i,k} + \lambda_1} \tag{5}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2789445, IEEE Transactions on Knowledge and Data Engineering

where $e_{u,i} = r_{u,i} - b_i - [\mathbf{p}_u; b_u]'[\mathbf{q}_i; 1]$ is defined as an error for this user-location pair. For notational convenience, we stack all $\mathbf{q}_i$ by rows into a matrix $\mathbf{Q}$ and augment it with $\mathbf{1}_N$, i.e., $\mathbf{Q}^1 = [\mathbf{Q}, \mathbf{1}_N]$. Defining $\mathbf{Q}^s = (\mathbf{Q}^1)'\text{diag}(\mathbf{d})\mathbf{Q}^1$ and leveraging the special structure of the weighting matrix, we can rewrite the above solution as

$$p^*_{u,k} = \frac{a_u \sum_i d_i(r_{u,i} - b_i)q_{i,k} + a_u(p_{u,k}q^s_{k,k} - [\mathbf{p}_u; b_u]'\mathbf{q}^s_k)}{\sum_i \tilde{w}_{u,i}q^2_{i,k} + a_u q^s_{k,k} + \lambda_1} + \frac{\sum_i \tilde{w}_{u,i}(e_{u,i}q_{i,k} + p_{u,k}q^2_{i,k})}{\sum_i \tilde{w}_{u,i}q^2_{i,k} + a_u q^s_{k,k} + \lambda_1}, \quad (6)$$

where the first numerator of the first term equals $a_u \sum_i d_i r_{u,i} q_{i,k} - a_u \sum_i d_i b_i q_{i,k}$, whose second term should also be precomputed and corresponds to the k^{th} entry of the vector $a_u(\mathbf{Q}^1)'(\mathbf{d} \circ \mathbf{b}_I)$. $\circ$ is the Hadamard (element-wise) product operator between matrices/vectors. Note that after learning $p_{u,k}$, we need to update the error for all her visited locations, $e^*_{u,i} = e_{u,i} + (p_{u,k} - p^*_{u,k})q_{i,k}$. Therefore, by pre-computing $\mathbf{Q}^s$ with the cost of $\mathcal{O}(NK^2)$, the time complexity of updating $p_{u,k}$ is $\mathcal{O}(K + \|\mathbf{r}_u\|_0)$ so that updating all users' latent vectors in one iteration costs $\mathcal{O}(MK^2 + \|\mathbf{R}\|_0 K)$, where the ℓ_0 norm $\|\mathbf{R}\|_0$ of matrix $\mathbf{R}$ equals the number of its non-zero entries.

Remark: According to Eq (5), if $q_{i,k} = 0\ \forall i \in \{1, \cdots, N\}$, then $p^*_{u,k} = 0$. Therefore, if $\mathbf{Q}$ is augmented as $\mathbf{Q}^0 = [\mathbf{Q}, \mathbf{0}_N]$, $b_u = 0, \forall u \in \{1, \cdots, M\}$; otherwise, if $\mathbf{Q}$ is augmented as $\mathbf{Q}^1 = [\mathbf{Q}, \mathbf{1}_N]$, $p_{u,K+1}$ corresponds to user bias b_u. In other words, whether user bias is considered or not just depends on how $\mathbf{Q}$ is augmented, without any adjustments to the optimization algorithm.

Similarly, the update rule of $q_{i,k}$ can be derived as

$$q^*_{i,k} = \frac{d_i \sum_u a_u(r_{u,i} - b_u)p_{u,k} + d_i(q_{i,k}p^s_{k,k} - [\mathbf{q}_i; b_i]'\mathbf{p}^s_k)}{\sum_u \tilde{w}_{u,i}p^2_{u,k} + d_i p^s_{k,k} + \lambda_2} + \frac{\sum_u \tilde{w}_{u,i}(e_{u,i}p_{u,k} + q_{i,k}p^2_{u,k})}{\sum_u \tilde{w}_{u,i}p^2_{u,k} + d_i p^s_{k,k} + \lambda_2}, \quad (7)$$

where $\mathbf{P}^s = (\mathbf{P}^1)'\text{diag}(\mathbf{a})\mathbf{P}^1$ and $e_{u,i} = r_{u,i} - b_u - [\mathbf{q}_i; b_i]'[\mathbf{p}_u; 1]$ by denoting $\mathbf{P}^1 = [\mathbf{P}, \mathbf{1}_M]$. Following similar analysis, the time complexity of updating all locations' latent factors is $\mathcal{O}(NK^2 + \|\mathbf{R}\|_0 K)$. Together with updating $\mathbf{P}$, the total time complexity of each iteration is only linear with K and the number of non-zero entries in the preference matrix when $\|\mathbf{R}\|_0 > \max(M, N) \times K$. This time complexity in each iteration is K times smaller than alternating least square (see the analysis in the Appendix). However, coordinate descent may require slightly more iterations than alternating least square because the update $p_{u,k}$ will affect the update of other coordinates of $\mathbf{p}_u$. We will compare their convergence in the experiment section.

4 Implicit Feedback based Content-aware Collaborative Filtering

ICF, in spite of the sparse and rank-one configuration, will fail in the case of the cold-start problem, specifically, recommending locations for new users. A general solution is to integrate collaborative filtering with content-based filtering [40]. From this research viewpoint, some popular content-aware collaborative filtering frameworks, such as LibFM, MatchBox, and SVDFeature, have recently been proposed, but they are designed based on explicit feedback with both positively and negatively preferred samples. Since only positively preferred samples are provided in implicit feedback datasets while it is impractical to treat all unvisited locations as negative, feeding mobility data together with user and location information into these explicit feedback frameworks requires drawing pseudo-negative samples from unvisited locations. The need to draw negative samples and the lack of different confidence levels cannot allow them to achieve the comparable top-k recommendation performance to ICF when not taking user/location information into consideration.

Without any extension from ICF to incorporate user/location information so far, in this section, we propose an Implicit-feedback based Content-aware Collaborative Filtering (ICCF) model for top-k location recommendation based on mobility data, to incorporate semantic content and steer clear of drawing negative samples. The overall framework is illustrated in Fig. 1, where users have features, such as profiles and textual content, provided in social networks like Twitter and Facebook, and locations have features, like category hierarchy and geographical information. After performing tokenization on textual content and discretizing continuous features (e.g., ages), all user features are encapsulated into a sparse user-feature matrix $\mathbf{X} \in \mathbb{R}^{M \times F}$, where F is the number of user features. Similarly, location features are also encapsulated into a sparse location-feature matrix $\mathbf{Y} \in \mathbb{R}^{N \times L}$, where L is the number of location features. Each entry $x_{u,f}$ in matrix $\mathbf{X}$ is the value of the f^{th} feature of user u and $y_{i,l}$ is the value of the l^{th} feature of location i. However, before feeding them into ICCF, these two feature matrices may need further preprocessing by applying tf-idf transformation and normalization (or standardization). After feeding them into ICCF, users and locations, as well as their features, are mapped into a joint latent space (represented by the rightmost part of Fig. 1). For optimizing ICCF, although it is intuitive to perform gradient descent directly, it is more appealing and effective to make use of variable substitution to decompose the learning of ICCF into two weighted alternative least square problems and two multivariate (sparse) linear regression problems, from the perspectives of extendibility, explainability, and convergence rate.

4.1 Prediction and Loss function

As mentioned above, ICCF takes a user-location preference matrix $\mathbf{R}$, a user-feature matrix $\mathbf{X}$, and a location-feature matrix $\mathbf{Y}$ as inputs. Based on these, ICCF first generates the weighting matrix $\mathbf{W}$ and the preference matrix $\mathbf{R}$ according to Eq. (3). It then follows MatchBox [20] and SVDFeature [18] to define the prediction preference of a user u for a location i as $\hat{r}_{u,i} = (\mathbf{p}_u + \mathbf{U}'\mathbf{x}_u)'(\mathbf{q}_i + \mathbf{V}'\mathbf{y}_i)$ when not considering biases, where each row of latent matrices $\mathbf{U} \in \mathbb{R}^{F \times K}$ and $\mathbf{V} \in \mathbb{R}^{L \times K}$ represents latent factors of user features and location features. Consequently, not only users and locations, but also their features are mapped into a joint latent space, where the inner product between them

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2789445, IEEE Transactions on Knowledge and Data Engineering

indicates one's preference for another. For example, the dot product $\mathbf{p}'_u\mathbf{v}_r$ between the latent factor of a user u and the latent factor of a location's feature r="restaurant" indicates the prediction preference of the user u for restaurants. If the ids of both users and locations are also considered as features and encapsulated into $\{\tilde{\mathbf{x}}_u\}$ and $\{\tilde{\mathbf{y}}_i\}$, the prediction preference is simplified as $\hat{r}_{u,i} = \tilde{\mathbf{x}}'_u\tilde{\mathbf{U}}\tilde{\mathbf{V}}'\tilde{\mathbf{y}}_i$, where $\tilde{\mathbf{U}} \in \mathbb{R}^{(M+F)\times K}$ is obtained by concatenating $\{\mathbf{p}_u\}$ and $\mathbf{U}$ by rows ($\tilde{\mathbf{V}}$ shares a similar meaning). LibFM [17], going further, encapsulates $\{\tilde{\mathbf{x}}_u\}$ and $\{\tilde{\mathbf{y}}_i\}$ into unified feature vectors and allows the interaction between users/locations and their features. However, since indexes of users and locations are implied in the unified feature vectors, it is difficult to distinguish the preference confidence for one user-location pair from another without referring to their unified feature vectors. Thus, such a representation is inappropriate in this case. Based on the prediction function, an objective loss function, taking into account the varying confidence of preference with visit frequency, is then formulated as:

$$\mathscr{L} = \frac{1}{2}\sum_{u,i} w_{u,i}(r_{u,i} - \tilde{\mathbf{x}}'_u\tilde{\mathbf{U}}\tilde{\mathbf{V}}'\tilde{\mathbf{y}}_i)^2 + \frac{\lambda_1}{2}\|\tilde{\mathbf{U}}\|_F^2 + \frac{\lambda_2}{2}\|\tilde{\mathbf{V}}\|_F^2. \quad (8)$$

Compared with prior content-aware frameworks, major differences lie in the introduction of the weighting matrix, incurring the loss function summing over all entries of the preference matrix, and the necessity of developing a novel efficient optimization algorithm. This is because the objective function of existing frameworks only depends on a small number of samples from the user-item matrix so that their excellent optimization algorithms almost cannot be exploited directly for the sake of efficiency. In other words, it is inefficient to perform naive alternating least square over each row of $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$. Below we try this method and elaborate the coupling difficulty of this approach between the optimization of latent factors of different features (i.e., different rows). Actually, this coupling difficulty results from the vastly varying confidences and user/location overlapping between different features. Taking user overlapping for example, two user features, "male" and "young", have a non-empty intersection of users with each other. In order to address this challenge, we propose a variable substitution technique for decoupling, splitting the overall optimization into two stages, the first of which learns the summed preference of users/locations over features and themselves (i.e., $\tilde{\mathbf{p}}_u \triangleq \tilde{\mathbf{U}}'\tilde{\mathbf{x}}_u$ and $\tilde{\mathbf{q}}_i \triangleq \tilde{\mathbf{V}}'\tilde{\mathbf{y}}_i$), by means of weighted alternating least square, and the second of which learns latent matrices $\mathbf{U}$ and $\mathbf{V}$ of user features and location features by means of multivariate (sparse) linear regression. The self-preference of a user or a location (i.e., $\mathbf{p}_u$ and $\mathbf{q}_i$) is then obtained with a subtraction operation.

4.2 Optimization

We first analyze the gradient of the objective function with respect to $\tilde{\mathbf{u}}_l$, the l^{th} row of $\tilde{\mathbf{U}}$, yielding,

$$\frac{\partial\mathscr{L}}{\partial\tilde{\mathbf{u}}_l} = \sum_{u,i} w_{u,i}\tilde{x}_{u,l}(\tilde{\mathbf{x}}'_u\tilde{\mathbf{U}}\tilde{\mathbf{q}}_i - r_{u,i})\tilde{\mathbf{q}}'_i + \lambda_1\tilde{\mathbf{u}}_l,$$

Setting the gradient to zero, we derive the analytic solution for $\tilde{\mathbf{u}}_l$ as follows:

$$(\tilde{\mathbf{u}}_l^{t+1})'(\sum_{u,i}\tilde{x}^2_{u,l}w_{u,i}\tilde{\mathbf{q}}_i\tilde{\mathbf{q}}'_i + \lambda_1\mathbf{I}_K) = \sum_{u,i} w_{u,i}r_{u,i}\tilde{x}_{u,l}\tilde{\mathbf{q}}'_i - \sum_{u,i}\tilde{x}_{u,l}\tilde{\mathbf{x}}'_u\tilde{\mathbf{U}}w_{u,i}\tilde{\mathbf{q}}_i\tilde{\mathbf{q}}_i{}' + (\tilde{\mathbf{u}}_l^t)'\sum_{u,i}x^2_{u,l}w_{u,i}\tilde{\mathbf{q}}_i\tilde{\mathbf{q}}'_i$$

where $\mathbf{I}_K$ is an identity matrix of size $K \times K$. Although this yields the analytic solution, its time complexity is dominated by the evaluation of $\sum_{u,i}\tilde{x}_{u,l}\tilde{\mathbf{x}}'_u\tilde{\mathbf{U}}w_{u,i}\tilde{\mathbf{q}}_i\tilde{\mathbf{q}}_i{}'$, costing at least $\mathcal{O}(\|\mathbf{R}\|_0K^2)$ for this feature. Since there are usually a large number of features considered, its calculation is far from efficient in practice. Additionally, precomputing this term for feature vectors of all rows (i.e., for all l) all at once and then updating them dynamically with the change in corresponding latent factor will still suffer from this coupling difficulty. In other words, updating latent factor of one feature will change latent factors corresponding to the overlapped features, where the overlapping between any two features is determined by the number of common users having these two features.

By analyzing the coupling between the updates of different features, we find that the coupling occurs for two reasons: varying confidence with visit frequency and user/location overlapping between features. Analyzing the procedure with respect to $\tilde{\mathbf{v}}_m$, the m^{th} row of $\tilde{\mathbf{V}}$, will yield a similar coupling difficulty. This coupling can be broken by splitting them into two stages, the first of which only depends on the former and the second of which only depends on the latter. This is achieved by a variable substitution technique to first learn the summed preference of users/locations, according to the relationship between $\tilde{\mathbf{x}}_u$ and $\mathbf{x}_u$ and the relationship between $\tilde{\mathbf{y}}_i$ and $\mathbf{y}_i$,

$$\mathbf{p}_u = \tilde{\mathbf{p}}_u - \mathbf{U}'\mathbf{x}_u \text{ and } \mathbf{q}_i = \tilde{\mathbf{q}}_i - \mathbf{V}'\mathbf{y}_i.$$

The loss function in Eq (8) is then converted into the following:

$$\mathscr{L} = \frac{1}{2}\sum_{u,i} w_{u,i}(r_{u,i} - \tilde{\mathbf{p}}'_u\tilde{\mathbf{q}}_i)^2 + \frac{\lambda_1}{2}\sum_u\|\tilde{\mathbf{p}}_u - \mathbf{U}'\mathbf{x}_u\|^2 + \frac{\lambda_2}{2}\sum_i\|\tilde{\mathbf{q}}_i - \mathbf{V}'\mathbf{y}_i\|^2 + \frac{\lambda_1}{2}\|\mathbf{U}\|_F^2 + \frac{\lambda_2}{2}\|\mathbf{V}\|_F^2. \quad (9)$$

It is easy to see that this new loss function is quadratic with respect to any one of variables $\{\tilde{\mathbf{p}}_u\}, \{\tilde{\mathbf{q}}_i\}$, $\mathbf{U}, \mathbf{V}$, and thus successfully decomposes the original coupling difficulty. Alternating optimization over these variables can now be exploited. This framework can easily be extended to include biases of users, locations, and their features using the approaches in Section 3. Hence, the following updating rules for parameters will take biases into account. At this moment, the last dimension $\mathbf{u}_{K+1}$ of $\mathbf{U}$ and the last dimension $\mathbf{v}_{K+1}$ of $\mathbf{V}$ stores regression coefficients of user bias and location bias, respectively.

Starting from $\{\tilde{\mathbf{p}}_u, \tilde{b}_u\}$, we apply coordinate descent for parameter learning. In particular, setting the derivative with respect to each coordinate of $[\tilde{\mathbf{p}}_u; \tilde{b}_u]$ to zero, we obtain the updated rules for $\tilde{p}_{u,k}$ ($\tilde{p}_{u,K+1} = \tilde{b}_u$) as follows,

$$\tilde{p}^*_{u,k} = \frac{\sum_i w_{u,i}(e_{u,i} + \tilde{p}_{u,k}\tilde{q}_{i,k})\tilde{q}_{i,k} + \lambda_1\mathbf{u}'_k\mathbf{x}_i}{\sum_i w_{u,i}\tilde{q}^2_{i,k} + \lambda_1} \quad (10)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2789445, IEEE Transactions on Knowledge and Data Engineering

where $\tilde{q}_{i,K+1} = 1, \forall i \in \{1, \cdots, N\}$ and $\mathbf{u}_k$ is the k^{th} column of the matrix $\mathbf{U}$. Based on the fast computation of Eq (6), Eq (10) can be computed quickly by adding $\mathbf{u}'_k\mathbf{x}_i$ into the numerator of Eq (6). When initializing $\mathbf{U}$ to a zero matrix, whether user bias is taken into account or not is still determined by how to augment $\tilde{\mathbf{Q}}$, as introduced in the parameter learning of Section 3.2.

Similarly, each coordinate of latent factor $[\tilde{\mathbf{q}}_i; \tilde{b}_i]$ can be updated as follows:

$$\tilde{q}^*_{i,k} = \frac{\sum_u w_{u,i}(e_{u,i} + \tilde{q}_{i,k}\tilde{p}_{u,k})\tilde{p}_{u,k} + \lambda_2 \mathbf{v}'_k\mathbf{y}_i}{\sum_u w_{u,i}\tilde{p}^2_{i,k} + \lambda_2} \tag{11}$$

where $\tilde{p}_{u,K+1} = 1, \forall u \in \{1, \cdots, M\}$ and $\mathbf{v}_k$ is the k^{th} column of the matrix $\mathbf{V}$. Based on the rapid computation of Eq (7), we can quickly compute Eq (11) by adding $\mathbf{v}'_k\mathbf{y}_i$ into the numerator of Eq (7). When initializing $\mathbf{V}$ to a zero matrix, whether location bias is taken into account or not is still determined by how to augment $\tilde{\mathbf{P}}$.

After updating $\{\tilde{\mathbf{p}}_u, \tilde{b}_u\}$ and $\{\tilde{\mathbf{q}}_i, \tilde{b}_i\}$, we continue updating $\mathbf{U}$ and $\mathbf{V}$. Since the objective functions with respect to $\mathbf{U}$ and $\mathbf{V}$ are almost similar, we only consider the former one. Taking all terms depending on $\mathbf{U}$, it is a multiple dependent-variable regression problem, but the regularized term is of equal importance to the regression error term. To control the importance of regularization, we multiply it with a new coefficient. The objective function with respect to $\mathbf{U}$ is then formulated as,

$$\mathscr{F}(\mathbf{U}) = \frac{1}{2}\|[\tilde{\mathbf{P}}, \tilde{\mathbf{b}}_U] - \mathbf{X}\mathbf{U}\|^2_F + \frac{\gamma}{2}\|\mathbf{U}\|^2_F. \tag{12}$$

The optimal $\mathbf{U}$ of this objective function is the solution of the following system of linear equations:

$$(\mathbf{X}'\mathbf{X} + \gamma\mathbf{I}_F)\mathbf{U} = \mathbf{X}'[\tilde{\mathbf{P}}, \tilde{\mathbf{b}}_U]. \tag{13}$$

If the coefficient matrix $(\mathbf{X}'\mathbf{X} + \gamma\mathbf{I}_F)$ is small in size, we can leverage matrix inversion to solve

$$\mathbf{U} = (\mathbf{X}'\mathbf{X} + \gamma\mathbf{I}_F)^{-1}\mathbf{X}'[\tilde{\mathbf{P}}, \tilde{\mathbf{b}}_U]. \tag{14}$$

When $\tilde{\mathbf{Q}}$ is augmented as $\tilde{\mathbf{Q}}^0 = [\tilde{\mathbf{Q}}, \mathbf{0}_N]$ in Eq (10), the optimal $\mathbf{b}_U$ is $\mathbf{0}_N$ if $\mathbf{u}_{K+1}$ is $\mathbf{0}_F$; and $\mathbf{u}_{K+1}$ is optimal at $\mathbf{0}_F$ if $\mathbf{b}_U = \mathbf{0}_N$. In other words, determining whether user/location bias is considered or not by how to augment $\mathbf{Q}/\mathbf{P}$ is also applicable in content-aware collaborative filtering algorithms, as long as we initialize $\mathbf{u}_{K+1}/\mathbf{v}_{K+1}$ to zeros.

When the number of features is far larger than the number of users, the update of $\mathbf{U}$ can be converted into a dual problem by using the matrix inversion lemma so that it only requires the inverse of a matrix of size $M \times M$ instead of the inverse of a matrix of size $F \times F$. In particular,

$$\mathbf{U} = \mathbf{X}'(\mathbf{X}\mathbf{X}' + \gamma\mathbf{I}_M)^{-1}[\tilde{\mathbf{P}}, \tilde{\mathbf{b}}_U]. \tag{15}$$

If the matrix to be inverted in the dual solution is still large in size, an alternative solution is to apply conjugate gradient descent, but it should be extended to matrix variable cases. Conjugate gradient descent is more efficient since it is unnecessary to explicitly precompute and store the coefficient matrix. Thus its time complexity depends only on the multiplication between matrices, costing $\mathcal{O}(\|\mathbf{X}\|_0 K \#iter)$, where $\#iter$ is the number of iterations of conjugate gradient descent to reach a given threshold of approximation error.

Complexity Analysis. Based on the updated rules of these variables, we show the overall optimization procedure with the presence of biases in Algorithm 1, where for simplicity we drop the tilde from $\mathbf{P}$ and $\mathbf{Q}$. Note that user-related biases and location-related biases can be easily turned off by replacing $\tilde{\mathbf{P}}^1$ with $\tilde{\mathbf{P}}^0$ and replacing $\tilde{\mathbf{Q}}^1$ with $\tilde{\mathbf{Q}}^0$, respectively. According to previous analysis, coordinate descent for learning parameters only costs $\mathcal{O}\big(\|\mathbf{R}\|_0 K + (M+N)K^2 + (\|\mathbf{X}\|_0 + \|\mathbf{Y}\|_0)K\big)$ in each iteration. Updating $\mathbf{U}$ and $\mathbf{V}$ based on conjugate gradient descent costs $\mathcal{O}\big((\|\mathbf{X}\|_0 + \|\mathbf{Y}\|_0)K\#iter\big)$. To summarize, the total time of each iteration is $\mathcal{O}\big((\|\mathbf{X}\|_0 + \|\mathbf{Y}\|_0)K\#iter + \|\mathbf{R}\|_0 K + (M+N)K^2\big)$. In other words, the time complexity of one iteration is in linear proportion to the number of non-zero entries in the user-location preference matrix, user-feature matrix, and location-feature matrix. Parallel updating $\{\tilde{\mathbf{p}}_u, \tilde{b}_u\}$ among users and parallel updating $\{\tilde{\mathbf{q}}_i, \tilde{b}_i\}$ among locations is possible since there is no dependence between their individual updates, so that, in practice, the time complexity can be greatly reduced given multiple CPUs in a single machine or in a distributed computing environment.

4.3 Explainability

For notation simplicity, we drop user bias and location bias in the following section. If the analytical update formulation of $\mathbf{U}$ is substituted back to the update of $\tilde{\mathbf{p}}_u$, applying matrix inversion lemma will get

$$\tilde{\mathbf{p}}_u = (\tilde{\mathbf{Q}}'\mathbf{W}^u\tilde{\mathbf{Q}} + \lambda_1\mathbf{I}_K)^{-1}(\tilde{\mathbf{Q}}'\mathbf{W}^u\mathbf{r}_u + \lambda_1\tilde{\mathbf{P}}'\mathbf{s}_u),$$

where $\mathbf{s}_u = (\mathbf{X}\mathbf{X}' + \gamma\mathbf{I}_M)^{-1}\mathbf{X}\mathbf{x}_u$ stores the similarity of user u with others, in terms of a function of the dot product between feature vectors. Let $\mathbf{X} = \mathbf{U}^X\mathbf{\Sigma}^X(\mathbf{V}^X)'$ be singular value decomposition of the feature matrix $\mathbf{X}$, subject to $(\mathbf{V}^X)'\mathbf{V}^X = \mathbf{V}^X(\mathbf{V}^X)' = \mathbf{I}_F$, $(\mathbf{U}^X)'\mathbf{U}^X = \mathbf{I}_F$ and $\mathbf{\Sigma}^X$ is a diagonal $F \times F$ matrix, then $\mathbf{s}_u = \mathbf{U}^X\text{diag}(\frac{\sigma_1^2}{\sigma_1^2+\gamma}, \cdots, \frac{\sigma_r^2}{\sigma_r^2+\gamma}, 0, \cdots, 0)\mathbf{u}^X_u$, where r is the rank of matrix $\mathbf{X}$ and $\mathbf{u}^X_u$ is the u^{th} row of matrix $\mathbf{U}^X$. This indicates $\tilde{\mathbf{p}}_u$ is not only dependent on mobility data in terms of the user-location preference matrix, but also the latent factors of similar users. Therefore, it is directly correlated with graph Laplacian regularized matrix factorization [21], which optimizes the following objective function when only considering user similarity:

$$\frac{1}{2}\sum_{u,i} w_{u,i}(r_{u,i} - \mathbf{p}'_u\mathbf{q}_i)^2 + \frac{\lambda}{2}(\|\mathbf{P}\|^2 + \|\mathbf{Q}\|^2) + \frac{\alpha}{2}\text{tr}(\mathbf{P}'(\mathbf{D}-\mathbf{S})\mathbf{P}),$$

where $\mathbf{S}$ is a symmetric similarity matrix between users, and $\mathbf{D}$ is a diagonal matrix, subject to $d_{u,u} = \sum_v s_{u,v}$. Setting the gradient with respect to $\mathbf{p}_u$ to zero, we can acquire the analytic solution for $\mathbf{p}_u$,

$$\mathbf{p}_u = (\mathbf{Q}'\mathbf{W}^u\mathbf{Q} + (\lambda + \alpha d_{u,u})\mathbf{I}_K)^{-1}(\mathbf{Q}'\mathbf{W}^u\mathbf{r}_u + \alpha\mathbf{P}'\mathbf{s}_u), \tag{16}$$

When similarity between users is measured as a cosine similarity based on feature vector, $s_{u,v} = \frac{\mathbf{x}'_u\mathbf{x}_v}{|\mathbf{x}_u||\mathbf{x}_v|} = \frac{1}{|\mathbf{\Sigma}^X\mathbf{u}^X_u||\mathbf{\Sigma}^X\mathbf{u}^X_v|}(\mathbf{u}^X_u)'\text{diag}(\sigma_1^2, \cdots, \sigma_r^2, 0, \cdots, 0)\mathbf{u}^X_v$. In the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2789445, IEEE Transactions on Knowledge and Data Engineering

Algorithm 1: Implicit Feedback based Content-ware Collaborative Filtering

Input : Preference matrix $\mathbf{R}$, latent space dimension K, user feature matrix $\mathbf{X}$, location feature matrix $\mathbf{Y}$
Output: User latent factor $\mathbf{P}$ and location latent factor $\mathbf{Q}$

Initialize $\mathbf{U}$, $\mathbf{V}$ with zero, initialize $\mathbf{P}$, $\mathbf{Q}$ randomly;
repeat
  pre-computing $\mathbf{Q}^s = (\mathbf{Q}^1)'\text{diag}(\mathbf{d})\mathbf{Q}^1$; // $\mathcal{O}(NK^2)$
  for $u \in \{1, \cdots, M\}$ **do**
    $\mathbf{u}^x = \mathbf{U}'\mathbf{x}_u$; // $\mathcal{O}(\|\mathbf{x}_u\|_0 K)$
    for $i \in \{i | r_{u,i} > 0\}$ **do**
      $e_{u,i} \leftarrow r_{u,i} - b_i - [\mathbf{p}_u; b_u]'\mathbf{q}_i^1$; // $\mathcal{O}(K)$
    end
    for $k \in \{1, \cdots, K+1\}$ **do** // $p_{u,K+1} = b_u$
      update $p_{u,k}$ by fast computing Eq (10); // $\mathcal{O}(K)$
      for $i \in \{i | r_{u,i} > 0\}$ **do**
        $e_{u,i} \leftarrow e_{u,i} + (p_{u,k} - p_{u,k}^*)q_{i,k}$; // $\mathcal{O}(1)$
      end
    end
  end
  pre-computing $\mathbf{P}^s = (\mathbf{P}^1)'\text{diag}(\mathbf{a})\mathbf{P}^1$; // $\mathcal{O}(MK^2)$
  for $i \in \{1, \cdots, N\}$ **do**
    $\mathbf{v}^y = \mathbf{V}'\mathbf{y}_i$; // $\mathcal{O}(\|\mathbf{y}_i\|_0 K)$
    for $u \in \{u | r_{u,i} > 0\}$ **do**
      $e_{u,i} \leftarrow r_{u,i} - b_u - [\mathbf{q}_i; b_i]'\mathbf{p}_u^1$; // $\mathcal{O}(K)$
    end
    for $k \in \{1, \cdots, K+1\}$ **do** // $q_{i,K+1} = b_i$
      update $q_{i,k}$ by fast computing Eq (11); // $\mathcal{O}(K)$
      for $u \in \{u | r_{u,i} > 0\}$ **do**
        $e_{u,i} \leftarrow e_{u,i} + p_{u,k}(q_{i,k} - q_{i,k}^*)$; // $\mathcal{O}(1)$
      end
    end
  end
  solve $(\mathbf{X}'\mathbf{X} + \gamma\mathbf{I}_F)\mathbf{U} = \mathbf{X}'[\tilde{\mathbf{P}}, \mathbf{b}_U]$;
  solve $(\mathbf{Y}'\mathbf{Y} + \gamma\mathbf{I}_F)\mathbf{V} = \mathbf{Y}'[\tilde{\mathbf{Q}}, \mathbf{b}_I]$;
until $\mathscr{L}$ *is convergent*;

case of user features with unit ℓ_2-norm, $s_{u,v} = \mathbf{x}_u'\mathbf{x}_v = (\mathbf{u}_u^X)'\text{diag}(\sigma_1^2, \cdots, \sigma_r^2, 0, \cdots, 0)\mathbf{u}_v^X$, so it differs from our proposed model in whether to shrink singular values by the regularization coefficient γ. Thus user similarity implicitly expressed in ICCF resembles this manually designed user similarity.

By applying the kernel trick, i.e., replacing $\mathbf{XX}'$ with a Gram matrix $\mathbf{K}$, it becomes possible to incorporate domain-specific similarity, such as user age proximity or document similarity between user tweets, into this framework . In this case, the update of $\tilde{\mathbf{p}}_u$ becomes

$$\tilde{\mathbf{p}}_u = (\tilde{\mathbf{Q}}'\mathbf{W}^u\tilde{\mathbf{Q}} + \lambda_1\mathbf{I}_K)^{-1}(\tilde{\mathbf{Q}}'\mathbf{W}^u\mathbf{r}_u + \lambda_1\tilde{\mathbf{P}}'(\mathbf{K} + \gamma\mathbf{I}_M)^{-1}\mathbf{k}_u),$$

where $\mathbf{k}_u$ is a column vector corresponding to the u^{th} row of matrix $\mathbf{K}$. The similarity of a user u with others is thus measured as $(\mathbf{K} + \gamma\mathbf{I}_M)^{-1}\mathbf{k}_u$. However, if $\tilde{\mathbf{p}}_u$ is updated in this way, it does not only require the inverse of a large sparse matrix but also makes the update of latent factors for different users coupling, rendering it difficult to be in parallel. For the sake of reserving both parallelism and the likelihood of incorporating domain-specific similarity, it may be necessary to perform feature mapping on a kernel matrix using a method such as eigenvalue decomposition or Random Kitchen Sinks [41], whose feasibility is guaranteed by the Mercer theorem, which states that kernels can be expressed as an inner product in some Hilbert space [42]. Additionally, in some special cases, the dot product based similarity in this formulation provides general rules or explanations for normalizing features. For example, document similarity is usually represented as a cosine similarity in the vector space, whose feature maps correspond to ℓ_2-norm normalized word vectors. Therefore, after transforming user-word matrix by tf-idf, it is often practical to perform row-based ℓ_2-norm normalization, making it unit ℓ_2-norm.

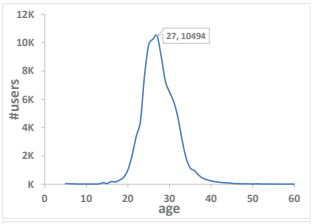
5 Experiments

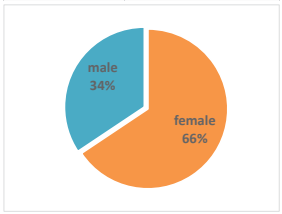
5.1 Dataset and Experimental setup

5.1.1 Dataset

ICCF is evaluated on a large-scale location-based social network dataset crawled from Jiepang, a Chinese location-based social network. We select POIs that are visited by at least ten users and users who have been to at least ten distinct locations. Finally, a total of 265,951 users and 189,850 POIs are then reserved, and the density of these users on these POIs is 3.69×10^{-4}.

duration	2011.3 - 2013.3
#check-in	18,653,706
#locations	189,850
#users	265,981
density	3.69E-04





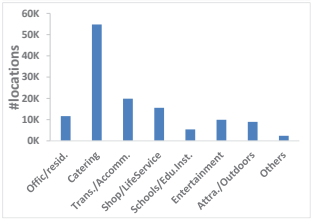


Fig. 2: Statistics information of the dataset.

Since many Jiepang users are linked to Weibo, a Chinese Microblog, we are able to collect rich semantic content, such as tweets and tags, and profile information, including age and gender, from users. This information can be used to improve recommendations in warm-start cases and address the cold-start problem. According to statistics, 62.4% of users have gender information and 35.8% of users have age information. Since users also have profiles on Jiepang, we crawl their profile pages and supplement them into their Weibo profile. After this, 100% of users have gender information but the portion of users with age information increases a little. Additionally, we also crawl each user's tags and tweets, with each user having around 6.7K words and only 5 tags on average. Since the number of total tags describing these users was over 300K, tag information was

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2789445, IEEE Transactions on Knowledge and Data Engineering

sparse. For each tweet, we apply Jieba[1] to perform word segmentation. After that, we remove stop words and use tf-idf to choose the top 8,000 distinct words as vocabulary [31]. Thus each user has around 1.5K words on average. We then perform ℓ_2-normalization on the tf-idf word vector for each user so that each user's word vector is of unit ℓ_2-norm. We also convert the tags of each user into a word vector, followed by the tf-idf transformation, and then perform ℓ_2-normalization. Moreover, each location has two levels of category hierarchy, where the first level contains 8 coarse categories and the second level contains 157 fine categories. 67.2% of the locations have both coarse and fine categories. Each location also has geographical information, which is encoded by influential areas of 250mx250m size within 1km according to [13]. The overall statistics are presented in Fig. 2.

5.1.2 Metrics

We evaluate recommendation algorithms on visited locations in the held-out set. Presenting each user with the top p candidate locations sorted by their preference prediction, we assess recommendation performance by checking how many of these locations actually appear in each user's held-out set. Two widely used metrics, i.e., recall at p and precision at p, in prior work [4], [6], [7], are exploited. The recall at p and the precision at p are defined as:

$$\text{recall@}p = \frac{1}{M}\sum_{u=1}^{M}\frac{|\mathbb{S}_u(p)\cap\mathbb{V}_u|}{|\mathbb{V}_u|}, \text{prec@}p = \frac{1}{M}\sum_{u=1}^{M}\frac{|\mathbb{S}_u(p)\cap\mathbb{V}_u|}{p},$$

where $\mathbb{S}_u(p)$ denotes the top p recommended locations for a user u and $\mathbb{V}_u$ is the set of visited locations in the held-out set.

5.1.3 Framework

We conduct two types of evaluation. One is in-matrix recommendation, referring to making recommendations for users who have mobility history in the system. It can examine improvements by introducing user profiles and textual content. The other is out-matrix recommendation, recommending locations for users who do not have any mobility data.

In the case of **in-matrix recommendation**, we randomly split each user's mobility data into five folds. For each fold, we fit a model to the other four folds (training part) and test the within-fold locations for each user. We form a predictive preference for the held-out set, generating a list of the top p recommended locations, and then calculate the metrics. After evaluating each fold, we report the averaged metrics.

In the case of **out-matrix recommendation**, we randomly split all users into five folds. For each fold, we fit a model to the submatrix formed by out-of-fold users (training part), and then test the recommendations for each user on the within-folds visited locations. Since each user in the test fold lacks training data, collaborative filtering fails in this case.

Based on these two schemes, we first compare ICCF with baselines[2], and then study the effects of user profiles and textual content, followed by the explanation of their effect at performance gain. In addition to recommendation performance, we also study the efficiency and convergence of the proposed algorithm in the end.

1. https://github.com/fxsjy/jieba
2. Please refer to the code via https://github.com/DefuLian/recsys

5.1.4 Parameter Settings

The latent space dimension is set to 150, when not studying the effect of different dimensions. The ϵ in the weight $\alpha(c_{u,i}) = 1 + \log(1 + c_{u,i} \times 10^{\epsilon})$ is set to 30 after being tunned within $\{1, 10, 30, 50, 100\}$ based on another 5-fold cross validation on the training parts. In the configuration of sparse and rank-one weighting scheme, $a_u = \frac{act_u^{\beta_1}}{\max_u act_u^{\beta_1}}$, where $act_u = \frac{\sum_i r_{u,i}}{\sum_{u,i} r_{u,i}}$, and $b_i = \frac{pop_i^{\beta_2}}{\max_i pop_i^{\beta_2}}$, where $pop_i = \frac{\sum_u r_{u,i}}{\sum_{u,i} r_{u,i}}$. β_1 and β_2 are tunned within $\{0.05, 0.1, 0.2, 0.4, 0.8\}$ by another 5-fold cross validation on the training parts, and set to 0.2 and 0.05, respectively. Here, we choose the item-weighting scheme in [39] instead of [35] because location visit frequency is subject to power law distribution [14] so that the frequency of locations at long tails cannot easily be distinguished from each other. λ_1 and λ_2 in Eq (9) are tuned over $\{0.01, 1, 50, 100, 300, 500, 1000, 5000\}$ based on another 5-fold cross validation on the training parts. In the end, when features are taken into account, $\lambda_1 = 500$, $\lambda_2 = 50$; otherwise, $\lambda_1 = 300$, $\lambda_2 = 100$.

5.1.5 Baselines

We compare the proposed algorithm with the following five baselines, whose latent space dimension is set to 150.

1) LibFM [17], using regression functionality because of its superiority to classification [1] and being optimized by MCMC, but varying the number of negative samples within $\{1, 3, 10\}$ for each positive sample, denoted as LibFM-1, LibFM-3, and LibFM-10.
2) GRMF, a graph regularized matrix factorization [3], [43], measuring the similarity between users and between locations as cosine similarity based on their respective content information. Its graph regularized coefficient is tuned over $\{0.01, 0.1, 1, 10, 100\}$.
3) LightFM [44], similar to ICCF, but uses the Weighted Approximate-Rank Pairwise (WARP) [45], the state-of-the-art ranking loss according to [26], [27], [46]. We almost follow the default setting of parameters, except the learning rate, which is tuned over $\{0.005, 0.01, 0.02, 0.04, 0.08, 0.16\}$.
4) GeoMF [13] and IRenMF [15], two state-of-the-art location recommendation algorithms according to [47]. The weighting matrix is set based on Eq (3), whose ϵ is also set to 30. The sparsity regularization coefficient in GeoMF is tuned over $\{0.1, 1, 5, 10, 50, 100\}$. For fair comparison, locations within 2km are considered similar in IRenMF since ICCF takes influential areas within 1km as input. However, the similarity decays exponentially with an increase of distance [15]. Other parameters in both algorithms are set as default values.

5.2 Experiment Results

5.2.1 Comparisons with baselines

The comparison of ICCF with baselines were shown in Fig. 3. First, we observe that ICCF outperforms LibFM by a significant margin (the standard errors are on the order of 10^{-4}). One of the reasons is that ICCF considers all unvisited locations as negative but assigns them a lower confidence for negative preference while LibFM only samples some of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2789445, IEEE Transactions on Knowledge and Data Engineering

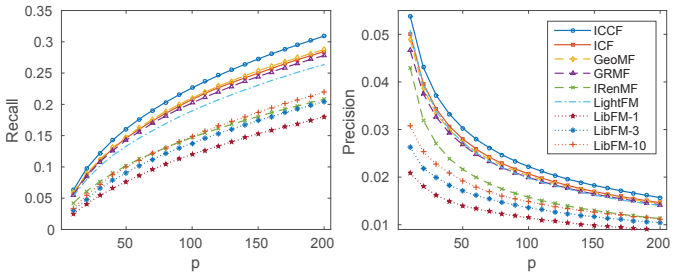


Fig. 3: Comparison with baselines.

them and treats them as equally important as positive ones. Moreover, LibFM improves recommendation performance with the increasing number of negative samples. This confirms the effect of unvisited locations as negative samples on recommendation based on mobility data [13]. However, the increase of negative samples reduces the efficiency of LibFM so that it is impractical to incorporate all unvisited locations. Second, ICCF has superior recommendation performance to LightFM, implying that weighted-regression loss is a good choice for recommendation from implicit feedback. Third, two state-of-the-art location recommendation algorithms, i.e., GeoMF and IRenMF, perform worse than ICCF. Fourth, ICCF is better than GRMF, due to the effectiveness of information incorporation and parameter learning. Actually, the recommendation performance of both GRMF and GeoMF is only close to ICF, indicating the benefit of sparse and rank-one weighting configuration in ICF. Finally, by comparing ICCF with ICF, incorporating user profiles and semantic content improve recommendation performance, indicating they are useful for promoting recommendation performance. However, the recommendation performance is still comparatively low, since location recommendation is a difficult task in the absence of any context. In practice, spatial and temporal context is usually available, it can help filter a lot of candidate locations due to beyond users' geographical range and mismatch with users' temporal preference, and can greatly improve the recommendation performance.

In order to understand the superiority of ICCF, we compare the basis of ICCF, i.e., ICF, with other types of matrix factorization, ranging from Bayesian Personalized Ranking Matrix Factorization [32], Weighted Approximate-Rank Pairwise (WARP) [45], Max Margin Matrix Factorization (MMMF) for implicit feedback datasets [48][3] to Hierarchical Poisson Factorization (HPF) [50]. The comparison is performed on three datasets, including this mobility dataset, a Foursquare mobility dataset [51] and a Gowalla mobility dataset [52]. In these datasets, we only use locations at which at least 10 users have checked in, and only consider users who have checked into at least 10 locations. After preprocessing, there are 269,217 records from 4,163 users on 121,142 locations remaining on the Foursquare dataset, and 1,315,047 records from 64,976 users on 54,691 locations remaining on the Gowalla dataset. The results of the comparison are shown in Fig. 4, where we have fine-tuned the hyper-parameters of these baselines.

The results show that ICF consistently and significantly outperforms other factorization models, although WARP also performs well for the top-k recommendation. BPRMF is not as good as WARP since BPRMF optimizes the area under curve (AUC) while WARP optimizes top-k precision directly. In spite of only substituting hinge-loss for logit-loss, MMMF does not perform as well as BPRMF. Although HPF can model the skewness of visit frequency [6], [14], [50] based on Poisson distribution, it cannot model varying confidence for negative and positive preferences so that it is also not as good as ICF. In a word, ICF is an optimal choice among them for a model-based recommendation algorithm based on mobility datasets, thus confirming the motivation of designing content-aware recommendation algorithms based on ICF.

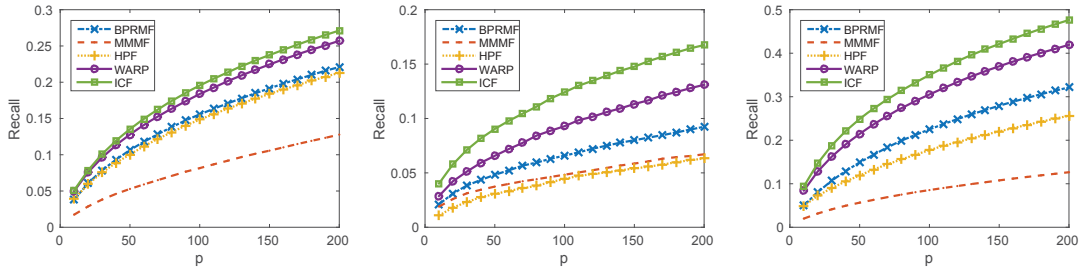
3. We use the implementation of MyMediaLite [49]

5.2.2 *Studying Effect of Bias and Weighting Schemes*

We further study the effects of user bias and location bias, with and without the presence of user/location features (corresponding to ICCF and ICF, respectively), and show the results in Fig. 5. When not taking user/location features into account, the biases improve recommendation performance but the relative improvement gradually decreases with an increase of data size. The relative improvements using 5%, 10%, 20%, 40% and 60% training data are 52.4%, 38.1%, 17.6%, 10.0% and 4.2%, respectively. This may be because in sparser data, bias plays a more important role in location recommendation. When considering user and location features, because of their effect on dealing with sparsity issues, the relative improvement becomes much smaller. The relative improvements using 5%, 10%, 20%, 40% and 60% training data are 15.6%, 7.5%, 5.1%, 1.6% and 1.2%, respectively. It is worth noting the lack of improvement by incorporating user and user-feature bias. This mainly lies in their independence of location ranking.

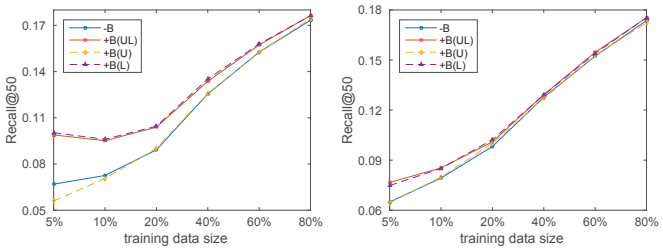
We then studied the effects of different weighting schemes in ICF. The results are shown in Fig. 6. We do not observe consistent improvements in exploiting the item-oriented weighting scheme by comparing "item" with "uniform", which is not in line with the results in [39]. We re-run their published code but still do not observe consistent improvements resulting from the item-oriented scheme (except in the leave-one-out evaluation). Compared with the item-oriented scheme, the user-oriented scheme and the user-item-oriented scheme show significantly superior recommendation performance, in particular the recommendation of long tail locations. More importantly, the relative improvement becomes larger and larger with an increase of latent space dimension from 50 to 300, indicating the effect of the proposed weighting scheme on learning parameters in more complex models. When the dimension is larger than 200, the relative improvements of recall@50, recall@100, recall@150 and recall@200 are at least 4.1%, 4.9%, 5.3% and 5.3%, respectively. The superiority of the user-oriented scheme to the item-oriented scheme is in line with the results in [35]. This observation can be reasonably explained by unvisited locations of active users (visiting many locations) being more potentially negative. According to the t-test of ten independent evaluation, the user-item-oriented scheme is significantly better than the user-oriented scheme for recommending long-tail locations (position $p \geq 136$) when the latent space dimension is large ($K \geq 250$), though the improvements are small. The significance level is 5%.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2789445, IEEE Transactions on Knowledge and Data Engineering



(a) Jiepang (b) Foursquare (c) Gowalla

Fig. 4: Comparison between matrix factorization models on different mobility datasets.



(a) ICF (b) ICCF

Fig. 5: The effect of biases.

5.2.3 Studying Effects of Profiles and Content

After demonstrating the total effects of profiles and textual content, we still cannot comprehend their individual effects. Therefore, we examine them carefully from in-matrix recommendation and out-matrix recommendation perspectives. One thing to note is that we do not observe the impact of tags on improving recommendation due to extreme sparsity and thus do not show the corresponding experiment results.

The results of in-matrix recommendation are shown in Fig. 7(a). Both profiles and content have an effect on improving the performance of recommendation individually compared to ICF. However, when they are integrated together, there is no further significant performance improvement. The possible reasons are two-fold: one is that profiles and textual content are correlated with each other; another is that regularized coefficients of these two types of latent factors may not be optimally tuned. This thus motivates Bayesian treatment for learning ICCF. Compared to the performance of ICF, the small improvement of the integrated one with two types of features implies the limited information gain from these two types of knowledge about users.

Although it has been shown that profiles and textual content are important for improving recommendation performance in the warm-start case, it is more necessary for recommendation in the cold-start case, where there is no or little past mobility data of users. We thus study this case, and show the results in Fig. 7(b). Since collaborative filtering fails in this case, it is not shown in these two figures. From them, we can see that both profiles and textual content of users are effective for recommending locations in cold-start cases, and that profiles are more effective than textual content. Moreover, when integrating them together, it makes further improvement in recommendation performance, indicating they complement each other. Nevertheless, the impact of incorporating content with profile is not as large as expected. One reason is that textual content is not as strongly correlated with recommending novel and attractive locations as profiles.

5.2.4 Explaining Effects of Profiles and Content

In order to thoroughly understand the amount of useful information that profiles and semantic content offer location recommendation, we analyze the learned relationship in terms of the dot product in latent space between locations and user profiles/content. The results are shown in Fig. 7(c)-7(d), and Fig. 8. We observe that male users prefer to show visits to offices, residences, hotels, and educational institutions while female users are more likely to visit shops, entertainment venues, and restaurants. Therefore, males and females have different preference when visiting locations. Based on the relationship between age and visited locations, we find that young users (around 18-26 years) prefer to visit campus-related locations like teaching buildings and universities. This is because most of these users are students, living in and around campus. It is more likely for users older than 26 years to visit restaurants and entertainment venues, since such a visit is more interesting to share with friends. However, the preferences of older users are much weaker than for younger ones. Finally, we measure the relationship between user tweets and locations by their dot product in the latent space and choose the top 100 correlated keywords. We then observe that most words associated with locations are geographical. Taking the locations of attractions and outdoors as an example, they can be correlated with "railway stations", "services zone", "hotels", and so on. Therefore, such a correlation may be not only explicit but also implicit, indicating their effectiveness in promoting recommending performance and dealing with cold-start cases. Unfortunately, most important keywords are geographical, without making full use of other irrelevant keywords to locations, meaning information gain from textual content is not as large as expected. However, only 35% of users make their ages public but there is no dependence between the features of users or locations in the current setting. This thus motivates another direction of research, that is, predicting user age based on semantic content before learning ICCF. Going further, it is possible to convert this process into an iterative one between learning ICCF and predicting ages based on textual content.

5.2.5 Efficiency Study

After evaluating the recommendation performance from the above perspectives, we continue evaluating the efficiency

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2789445, IEEE Transactions on Knowledge and Data Engineering

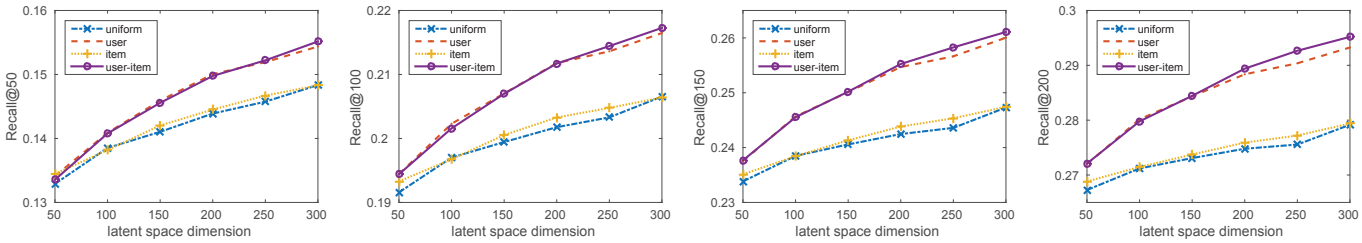
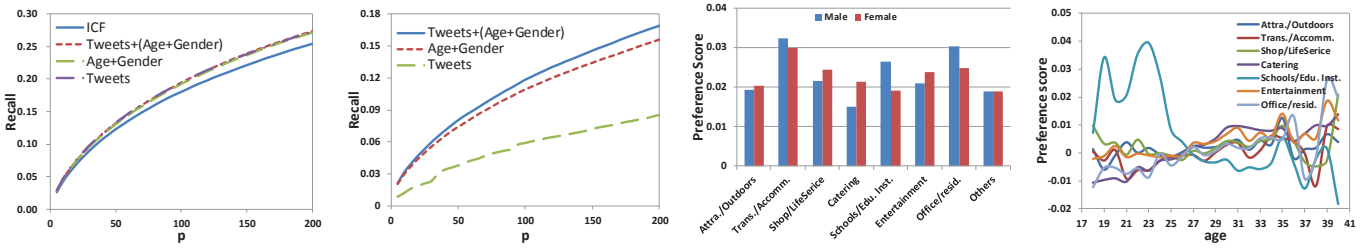


Fig. 6: The study of weighting schemes.



(a) Recall-Feature-In (b) Recall-Feature-Out (c) Explanation-Gender (d) Explanation-Age

Fig. 7: The effect of user's profiles.

(a) Attractions & Outdoors (b) Catering (c) School & Edu. Institution (d) Beijing Int'l Airport

Fig. 8: Correlation between user's tweets and locations.

of the proposed algorithm. First, we run the proposed algorithm using different dimensions of latent space and record the corresponding running times. The relationship between dimension K and the running time in seconds is shown in Fig. 9(a). We see that alternating least square costs more time compared to coordinate descent within the same number of iterations, particularly when the dimension of latent space is large. In other words, coordinate descent is more scalable than alternating least square with the increase of latent space dimension. Second, we feed different percentages of training data into the proposed algorithm and record the corresponding running time of the proposed algorithm, with the results shown in Fig. 9(b). From this figure, except for explicitly higher efficiency of coordinate descent, both algorithms are almost linear with data size, being scalable with the increasing size of training data.

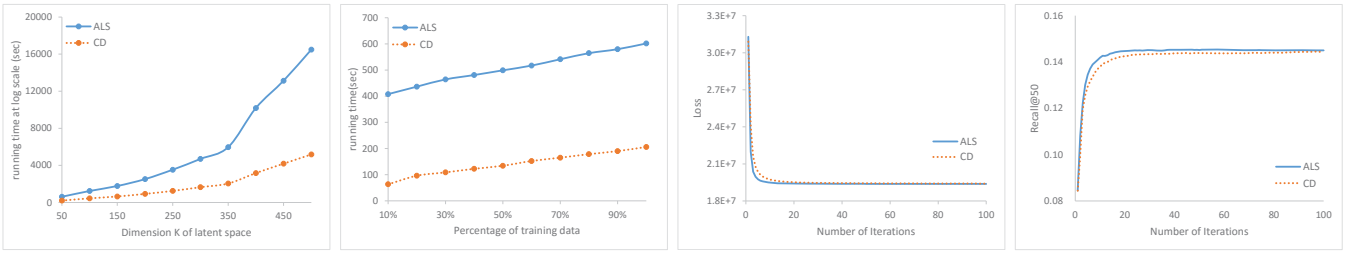
5.2.6 Convergence Study

Although coordinate descent costs less time than alternating least square given the same number of iterations, it may need a larger number of iterations for achieving the same precision in loss value. Thus we study the convergence of both algorithms, and show the results in Fig. 9(c) and (d). From these two figures, we first see that both algorithms are convergent within 50 iterations at most. Second, either the loss value measured by $\mathscr{L}$ or the recommendation performance measured by recall@50 of coordinate descent are more slowly convergent than alternating least square, as deduced in Section 3.2. However, their differences in recall and loss are subtle after 30 iterations. Therefore, coordinate descent is a comparatively optimal choice for learning user latent factor and location latent factor.

6 Conclusions

In this paper, we propose an ICCF framework for content-aware collaborative filtering from implicit feedback datasets, and develop coordinate descent for efficient and effective parameter learning. We establish ICCF's close relationship with graph Laplacian regularized matrix factorization and show that user features actually refine mobility similarity between users. We then apply ICCF for location recommendation on a large-scale LBSN dataset. Our experiment results indicate that ICCF is superior to five competing baselines, including two state-of-the-art location recommendation algorithms and ranking-based factorization machine. By comparing different weighting schemes for negative preference of unvisited locations, we observe that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2789445, IEEE Transactions on Knowledge and Data Engineering



(a) Dimension vs. time (b) Data size vs. time (c) Iteration vs. loss (d) Iteration vs. recall@50

Fig. 9: (a)&(b) efficiency study and (c)&(d) convergence study.

the user-oriented scheme is superior to the item-oriented scheme, and that the sparse and rank-one configuration significantly improves recommendation performance. The evaluation of biases reveals that they play an important role in recommendation from sparse datasets. By studying the effects of user profiles and semantic content, we find that they improve recommendation in warm-start cases and help address the cold-start problems. Finally, we empirically study the issues of efficiency and convergence of the proposed algorithm, and observe that coordinate descent is more slowly convergent than alternating least square, but the differences of their recommendation performances and the differences of their objective values are subtle after dozens of iterations, implying coordinate descent is a better choice for learning parameters.

Acknowledgments

We appreciate the valuable suggestions from anonymous reviewers. This work is supported by the National Natural Science Foundation of China (61502077,61631005), the Fundamental Research Funds for the Central Universities (ZYGX2014Z012, ZYGX2016J087), and Anhui Science and Technology Project of China (1604b0602025).

References

[1] D. Lian, Y. Ge, F. Zhang, N. J. Yuan, X. Xie, T. Zhou, and Y. Rui, "Content-aware collaborative filtering for location recommendation based on human mobility data," in *Proceedings of ICDM'15*. IEEE, 2015, pp. 261–270.

[2] C. R. Cloninger, T. R. Przybeck, and D. M. Svrakic, *The Temperament and Character Inventory (TCI): A guide to its development and use*. center for psychobiology of personality, Washington University St. Louis, MO, 1994.

[3] V. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang, "Collaborative filtering meets mobile recommendation: A user-centered approach," in *Proceedings of AAAI'10*. AAAI Press, 2010.

[4] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proceedings of SIGIR'11*. ACM, 2011, pp. 325–334.

[5] W.-Y. Zhu, W.-C. Peng, L.-J. Chen, K. Zheng, and X. Zhou, "Modeling user mobility for location promotion in location-based social networks," in *Proceedings of KDD'15*. ACM, 2015, pp. 1573–1582.

[6] B. Liu, Y. Fu, Z. Yao, and H. Xiong, "Learning geographical preferences for point-of-interest recommendation," in *Proceedings of KDD'13*. ACM, 2013, pp. 1043–1051.

[7] H. Gao, J. Tang, X. Hu, and H. Liu, "Exploring temporal effects for location recommendation on location-based social networks," in *Proceedings of RecSys'13*. ACM, 2013, pp. 93–100.

[8] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, "Time-aware point-of-interest recommendation," in *Proceedings of SIGIR'13*. ACM, 2013, pp. 363–372.

[9] Q. Yuan, G. Cong, and A. Sun, "Graph-based point-of-interest recommendation with geographical and temporal influences," in *Proceedings of CIKM'14*. ACM, 2014, pp. 659–668.

[10] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "A random walk around the city: New venue recommendation in location-based social networks," in *Proceedings of SocialCom'12*. IEEE, 2012, pp. 144–153.

[11] D. Yang, D. Zhang, Z. Yu, and Z. Wang, "A sentiment-enhanced personalized location recommendation system," in *Proceedings of HT'13*. ACM, 2013, pp. 119–128.

[12] B. Liu and H. Xiong, "Point-of-interest recommendation in location based social networks with topic and location awareness," in *Proceedings of SDM'13*. SIAM, 2013, pp. 396–404.

[13] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation," in *Proceedings of KDD'14*. ACM, 2014, pp. 831–840.

[14] C. Cheng, H. Yang, I. King, and M. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in *Proceedings of AAAI'12*, 2012.

[15] Y. Liu, W. Wei, A. Sun, and C. Miao, "Exploiting geographical neighborhood characteristics for location recommendation," in *Proceedings of CIKM'14*. ACM, 2014, pp. 739–748.

[16] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender systems handbook*. Springer, 2011, pp. 73–105.

[17] S. Rendle, "Factorization machines with libfm," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 3, p. 57, 2012.

[18] T. Chen, W. Zhang, Q. Lu, K. Chen, Z. Zheng, and Y. Yu, "Svdfeature: a toolkit for feature-based collaborative filtering," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 3619–3622, 2012.

[19] D. Agarwal and B.-C. Chen, "Regression-based latent factor models," in *Proceedings of KDD'09*. ACM, 2009, pp. 19–28.

[20] D. H. Stern, R. Herbrich, and T. Graepel, "Matchbox: large scale online bayesian recommendations," in *Proceedings of WWW'09*. ACM, 2009, pp. 111–120.

[21] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1548–1560, 2011.

[22] T. Horozov, N. Narasimhan, and V. Vasudevan, "Using location for personalized poi recommendations in mobile environments," in *Proceedings of SAINT'06*. IEEE Computer Society, 2006.

[23] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W. Ma, "Recommending friends and locations based on individual location history," *ACM Trans. Web*, vol. 5, no. 1, p. 5, 2011.

[24] K. Zheng, S. Shang, N. J. Yuan, and Y. Yang, "Towards efficient search for activity trajectories," in *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*. IEEE, 2013, pp. 230–241.

[25] H. Li, R. Hong, D. Lian, Z. Wu, M. Wang, and Y. Ge, "A relaxed ranking-based factor model for recommender system from implicit feedback," in *Proceedings of IJCAI'16*, 2016, pp. 1683–1689.

[26] W. Zhang and J. Wang, "Location and time aware social collaborative retrieval for new successive point-of-interest recommendation," in *Proceedings of CIKM'15*. ACM, 2015, pp. 1221–1230.

[27] X. Li, G. Cong, X. Li, T.-A. N. Pham, and S. Krishnaswamy, "Rank-geofm: A ranking based geographical factorization method for point of interest recommendation," in *Proceedings of SIGIR'15*. ACM, 2015, pp. 433–442.

[28] X. Liu, Y. Liu, and X. Li, "Exploring the context of locations for personalized location recommendations," in *Proceedings of IJCAI'16*. AAAI, 2016.

[29] H. Gao, J. Tang, X. Hu, and H. Liu, "Content-aware point of interest recommendation on location-based social networks," in *Proceedings of AAAI'15*. AAAI, 2015.

[30] F. Aiolli, "Efficient top-n recommendation for very large scale binary rated datasets," in *Proceedings of RecSys'13*. ACM, 2013, pp. 273–280.
[31] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proceedings of KDD'11*. ACM, 2011, pp. 448–456.
[32] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of UAI'09*. AUAI Press, 2009, pp. 452–461.
[33] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: successive point-of-interest recommendation," in *Proceedings of IJCAI'13*. AAAI Press, 2013, pp. 2605–2611.
[34] J. Canny, "Gap: a factor model for discrete data," in *Proceedings of SIGIR'04*. ACM, 2004, pp. 122–129.
[35] R. Pan, Y. Zhou, B. Cao, N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *Proceedings of ICDM'08*. IEEE, 2008, pp. 502–511.
[36] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proceedings of ICDM'08*. IEEE, 2008, pp. 263–272.
[37] D. Lian, Y. Ge, N. J. Yuan, X. Xie, and H. Xiong, "Sparse bayesian content-aware collaborative filtering for implicit feedback," in *Proceedings of IJCAI'16*. AAAI, 2016.
[38] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
[39] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proceedings of SIGIR'16*, vol. 16, 2016.
[40] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review*, vol. 13, no. 5-6, pp. 393–408, 1999.
[41] A. Rahimi and B. Recht, "Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning," in *Proceedings of NIPS'09*, 2009, pp. 1313–1320.
[42] J. Mercer, "Functions of positive and negative type, and their connection with the theory of integral equations," *Philosophical transactions of the royal society of London. Series A*, pp. 415–446, 1909.
[43] T. Zhou, H. Shan, A. Banerjee, and G. Sapiro, "Kernelized probabilistic matrix factorization: Exploiting graphs and side information." in *SDM*, vol. 12. SIAM, 2012, pp. 403–414.
[44] M. Kula, "Metadata embeddings for user and item cold-start recommendations," in *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems*, vol. 1448, 2015, pp. 14–21.
[45] J. Weston, S. Bengio, and N. Usunier, "Large scale image annotation: learning to rank with joint word-image embeddings," *Machine learning*, vol. 81, no. 1, pp. 21–35, 2010.
[46] F. Yuan, G. Guo, J. M. Jose, L. Chen, H. Yu, and W. Zhang, "Lambdafm: learning optimal ranking with factorization machines using lambda surrogates," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 227–236.
[47] Y. Liu, T.-A. N. Pham, G. Cong, and Y. Quan, "An experimental evaluation of point-of-interest recommendation in location-based social networks," *Proceedings of VLDB'17*, vol. 10, no. 10, pp. 1010–1021, 2017.
[48] M. Weimer, A. Karatzoglou, and A. Smola, "Improving maximum margin matrix factorization," *Machine Learning*, vol. 72, no. 3, pp. 263–276, 2008.
[49] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "MyMediaLite: A free recommender system library," in *Proceedings of RecSys 2011*, 2011.
[50] P. Gopalan, J. M. Hofman, and D. M. Blei, "Scalable recommendation with hierarchical poisson factorization," in *Proceedings of UAI'15. AUAI Press*, 2015.
[51] H. Li, Y. Ge, and H. Zhu, "Point-of-interest recommendations: Learning potential check-ins from friends," in *Proceedings of KDD'16*. ACM, 2016.
[52] E. Cho, S. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proceedings of KDD'11*, 2011, pp. 1082–1090.

Defu Lian received his B.E. and Ph.D. in Computer Science from the University of Science and Technology of China (USTC) in 2009 and 2014. He is currently a lecturer of University of Electronic Science and Technology of China (UESTC). His main research interests include mobile data mining and recommender systems. He has published more than 30 papers in the journals and conference such as KDD, ICDM, IJCAI and ACM TIST. He also was a reviewer for several journals such as TKDE, TOIS, KAIS.

Yong Ge received his Ph.D. in Information Technology from Rutgers, The State University of New Jersey in 2013, an M.S. in Signal and Information Processing from the University of Science and Technology of China (USTC) in 2008, and a B.E. in Information Engineering from Xi'an Jiao Tong University in 2005. He is currently an Assistant Professor at the University of Arizona. His research interests include data mining and business analytics. He received the ICDM-2011 Best Research Paper Award, and the Dissertation Fellowship at Rutgers University in 2012. He has published prolifically in refereed journals and conferences, such as TKDE, TOIS, SIGKDD, and ICDM. He has served as a reviewer for numerous journals, including TKDE, TKDD and KAIS.

Fuzheng Zhang is now an associate researcher at Microsoft Research Asia. He received his Ph.D. in Computer Science from the University of Science and Technology of China in 2015, and B.S. both in Computer Science and Statistics and Finance from the University of Science and Technology of China in 2010. His research mainly focuses on user modeling and social network computing, by using techniques such as deep learning, data mining, natural language analysis, etc. He has published academic papers frequently on reputable international conferences and journals in his research area, such as KDD, WWW, Ubicomp, TIST. He received the best paper award of ICDM2013.

Nicholas Jing Yuan is currently a senior applied scientist lead at Microsoft. He was previously a researcher at Microsoft Research Asia. He earned his Ph.D in Computer Science in 2012 and his B.S in Mathematics in 2007, both from University of Science and Technology of China. During the past few years, he has published more than 60 papers in top-tier conferences and journals, including ACM SIGKDD, IEEE TKDE and WWW. His work has been featured in influential media outlets including multiple features in MIT Technology Review. He has been honored with the Microsoft Fellowship (2011), the Best Paper Award of ICDM (2013), the Best Student Paper Award of SIGKDD (2016). His research interests include behavioral data mining, spatial-temporal data mining and computational social science.

Xing Xie is currently a senior research manager in Microsoft Research Asia, and a guest Ph.D. advisor for the University of Science and Technology of China. He received his B.S. and Ph.D. in Computer Science from the University of Science and Technology of China in 1996 and 2001, respectively. His research interests include data mining, social computing and ubiquitous computing. He has published over 160 referred journal and conference papers, including IEEE TKDE and ACM SIGKDD. He currently serves on the editorial boards of ACM TIST, ACM IMWUT, GeoInformatica and PMC.

Tao Zhou received a B.S. in Physics from the University of Science and Technology of China, a Ph.D. in Physics from Fribourg university. He is currently a professor in University of Electronic Science and Technology of China. His main research interests include data mining, network science and collective dynamics. He has published many research articles in prestigious journals (e.g., Physics Reports, PNAS, Nature Communication). His works have been reported by several academic media outlets such as Nature News, PNAS News, MIT Technology Review.

Yong Rui received a B.S. from Southeast University, a M.S. from Tsinghua University, and a Ph.D. from the University of Illinois at Urbana Champaign. He was the Deputy Managing Director of Microsoft Research Asia (MSRA), leading research groups in multimedia search and mining, and big data analysis, and engineering groups in multimedia processing, data mining, and software/hardware systems. He has authored 16 books and book chapters, and more than 100+ refereed journal and conference papers. He is a fellow of IAPR and SPIE, a Distinguished Scientist of ACM, and a Distinguished Lecturer of both ACM and IEEE. He is the Editor-in-Chief of the IEEE Multimedia Magazine, an Associate Editor of the ACM Transactions on Multimedia Computing, Communication and Applications, the Advisory Board of the IEEE Transactions on Automation Science and Engineering, a Founding Editor of the International Journal of Multimedia Information Retrieval, and a Founding Associate Editor of the IEEE ACCESS.