2015 IEEE International Conference on Data Mining

Content-Aware Collaborative Filtering for Location Recommendation based on Human Mobility Data

Defu Lian[†], Yong Ge[‡], Fuzheng Zhang[§], Nicholas Jing Yuan[§], Xing Xie[§], Tao Zhou[†], Yong Rui[§]
[†]Big Data Research Center, University of Electronic Science and Technology of China
[‡]University of North Carolina at Charlotte
[§]Microsoft Research, Beijing, China
Email: {dove.ustc,ygestrive,zhutouster} at gmail.com, {fuzzhang,nichy,xingx,yongrui} at microsoft.com

***Abstract*—Location recommendation plays an essential role in helping people find places they are likely to enjoy. Though some recent research has studied how to recommend locations with the presence of social network and geographical information, few of them addressed the cold-start problem, specifically, recommending locations for new users. Because the visits to locations are often shared on social networks, rich semantics (e.g., tweets) that reveal a person's interests can be leveraged to tackle this challenge. A typical way is to feed them into traditional explicit-feedback content-aware recommendation methods (e.g., LibFM). As a user's negative preferences are not explicitly observable in most human mobility data, these methods need draw negative samples for better learning performance. However, prior studies have empirically shown that sampling-based methods don't perform as well as a method that considers all unvisited locations as negative but assigns them a lower confidence. To this end, we propose an Implicit-feedback based Content-aware Collaborative Filtering (ICCF) framework to incorporate semantic content and steer clear of negative sampling. For efficient parameter learning, we develop a scalable optimization algorithm, scaling linearly with the data size and the feature size. Furthermore, we offer a good explanation to ICCF, such that the semantic content is actually used to refine user similarity based on mobility. Finally, we evaluate ICCF with a large-scale LBSN dataset where users have profiles and text content. The results show that ICCF outperforms LibFM of the best configuration, and that user profiles and text content are not only effective at improving recommendation but also helpful for coping with the cold-start problem.**

I. Introduction

As cities develop, the growing number of locations of interest, such as hotels, attractions, restaurants, and so on, offer people more opportunities for amusement than ever before. At the same time, since novelty seeking is regarded as a basic requirement for human activities [1], people love to explore neighborhoods and visit locations tailored to their interests. Therefore, location recommendation has been exploited to help people discover interesting places [2], [3] and speed up users' familiarization with their surroundings.

The advent of location-based social networks (LBSNs), such as Foursquare, Jiepang, Yelp, and so on, makes it possible to analyze large-scale human mobility data. With massive data support, location recommendation has become a popular research topic recently. Prior research has mainly investigated how to leverage spatial patterns [3], [4], [5], [6], temporal effects [7], [8], spatio-temporal influence [9], social influence [10], [11], text-based analysis [12], [13], and implicit characteristics of human mobility [14], [15], [16] to recommend locations. However, some of these methods require each user to have sufficient training data while others assume locations have accumulated ample review information (e.g., tips), so it is a great challenge for 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 interests, an essential element for capturing users' visiting behavior [17]. Therefore, they can be exploited to address the cold-start challenge and even improve location recommendation. A typical way is to feed them into traditional explicit-feedback content-aware recommendation frameworks, such as LibFM [18], SVDFeature [19], regression-based latent factor model [20] or MatchBox [21]. Since a user's negative preferences for locations are not explicitly observable in human mobility data, these frameworks need draw negative samples from unvisited locations for better learning performance. However, it has been empirically shown that sampling-based frameworks don't perform as well as an algorithm that treats all unvisited locations as negative yet assigns them a lower preference confidence [14], [16]. Last but not least, it is not intuitive for these frameworks to capture another characteristic of mobility data: the varying confidence for positive preference with visit frequency or stay duration.

With this in mind, we propose a novel Implicit-feedback based Content-aware Collaborative Filtering (ICCF) framework. It steers clear of sampling negative items, by treating all unvisited locations as negative and exploiting a similar weighting matrix to [22], [23] for modeling the preference confidence. Specifically, it assigns unvisited locations a lower confidence for negative preferences and assigns visited ones a varying confidence for positive preferences with the visit frequency. Taking a user-item matrix, a user-feature matrix (e.g., gender, age and tweets) and an item-feature matrix (e.g., categories, descriptions) as input, ICCF maps each user, each item and each feature of the users and items onto a joint latent space, such that dot product between the images of two objects defines a preference score. For example, 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. The mapping procedure is achieved by a novel variable substitution technique to split the learning of ICCF into two weighted least square problems w.r.t the latent factors of users and items, and two (sparse) multivariate linear regression problems w.r.t the latent factors of the features of users and items. Based on alternative least square among latent

1550-4786/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDM.2015.69

factors of these four types, the time complexity of each round of optimization is in linear proportion to the number of non-zero entries in the user-item matrix, user-feature matrix, and item-feature matrix, scaling linearly with the data size and the feature size.

After deep analysis to ICCF, we offer a good explanation to it, such that user (item) features refine the similarity between users (items) on implicit feedback, and the refining procedure is connected with matrix factorization with manifold regularization [24]. Therefore, ICCF not only becomes an alternative for the latter techniques, but it also becomes possible to provide general rules for normalizing features of items and users, and to incorporate domain-specific knowledge, such as document similarity between user tweets (e.g., vector space model) and age proximity between users. In addition to previous advantages, ICCF is also capable of selecting correlated features directly by imposing ℓ_1-norm on the latent factors of user features and item features, and then exploiting proximal gradient descent [25] for parameter learning.

We then apply ICCF for location recommendation based on human mobility data of 36M visit records obtained from a location-based social network, where items have two levels of categories and users have profiles, including gender and age, and rich semantic text content, including tweets and tags, crawled from a social network. Based on the evaluation results of five-fold cross validation on mobility data, corresponding to the warm-start case, we observed that ICCF was greatly superior to LibFM of the best configuration. This result implies explicit-feedback based content-aware collaborative filtering algorithms don't work sufficiently well on implicit feedback datasets, confirming the necessity of developing ICCF. In addition, based on this evaluation, we found that user profiles and semantic content can make significant improvement over the counterpart without any content. In addition to the warm-start evaluation, we also performed a cold-start evaluation with user-based five-fold cross validation (splitting users into five folds). The results indicated that both user profiles and semantic content were useful for tackling the cold-start problem in location recommendation based on human mobility data, and that user profiles were more effective than semantic content.

II. Preliminary

Given a mobility data of M users visiting N locations, location recommendation first converts it into a user-location 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, where u and i are reserved for indexing users and locations, respectively. Following common symbolic notation, upper case bold letters denote matrices, lower case bold letters denote column vectors without any specification, and non-bold letters represent scalars.

Given the user-location frequency matrix $\mathbf{C}$, although recommendation algorithms such as Bayesian Personalized Ranking based Matrix Factorization [26] and Bayesian non-negative matrix factorization [27] have been exploited for recommendation locations, they are still not comparable to the best top-k location recommender based on mobility data: weighted regression-based One Class Collaborative Filtering (OCCF) algorithms [23], [22], as shown in prior studies [14], [16]. This fact will be validated again in later experiments on two other different implicit feedback datasets. In a mobility dataset, a user's visit to a location only implies her positive preference. The negative preferences for unvisited locations have not explicitly been observed. Moreover, the visit frequency to locations determines the confident level of positive preference, such that a higher frequency corresponds to a larger confidence. The weighted regression-based OCCF algorithms just capture these two characteristics by assuming the confidence level for positive preference increases with the frequency and treating unvisited locations as negative. At the same time, since the confidence in the negative attitude is significantly less than the positive attitude of visited locations, the confidence in the negative preference is assigned a lower value than the positive preference. It could be efficient to consider all unvisited locations as negative, as long as the confidence level in the preferences for most negative locations is assigned the same value, e.g., 1. We thus follow [22] and set the overall confidence as,

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

where $\alpha(c_{u,i}) > 0$ is a monotonically increasing function with respect to $c_{u,i}$ so that the positive confidence increases with the visit frequency. The objective function best tailored to top-k location recommendation based on mobility data is then represented as follows:

$$\mathscr{L} = \frac{1}{2}\sum_{u,i} w_{u,i}(r_{u,i} - \mathbf{p}'_u \mathbf{q}_i)^2 + \frac{\lambda}{2}\Big(\sum_u \|\mathbf{p}_u\|^2 + \sum_i \|\mathbf{q}_i\|^2\Big),$$

where $r_{u,i} = \mathbb{I}(c_{u,i} > 0)$ is an entry of 0/1 rating matrix $\mathbf{R}$, indicating whether a user u has visited a location i, and $\mathbf{p}_u \in \mathbb{R}^K$ and $\mathbf{q}_i \in \mathbb{R}^K$ are latent factors of the user u and the location i, respectively. The right part of the objective function is a regularized term to avoid over-fitting. This form of objective actually approximates whether users have visited locations in a weighted way, by mapping users and locations into a joint latent space of dimension $K \ll \min(M, N)$. Although it is observed that this objective is summed over all entries in the user-location matrix, it can be efficiently optimized via alternative least square. The time complexity for each round of optimization is linearly proportional to the number of non-zero entries of the frequency matrix. Below, we call this model Implicit-feedback based Collaborative Filtering algorithm (ICF).

However, ICF 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 [28]. From this research viewpoint, some popular content-aware collaborative filtering frameworks, such as LibFM, MatchBox, and SVDFeature, have been recently 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 item content into these explicit feedback frameworks requires drawing pseudo-negative samples from unvisited locations. The need of drawing negative sample and the lack of different levels of confidence cannot allow them to achieve the comparable top-k recommendation performance to ICF when no content is provided, as discussed above.

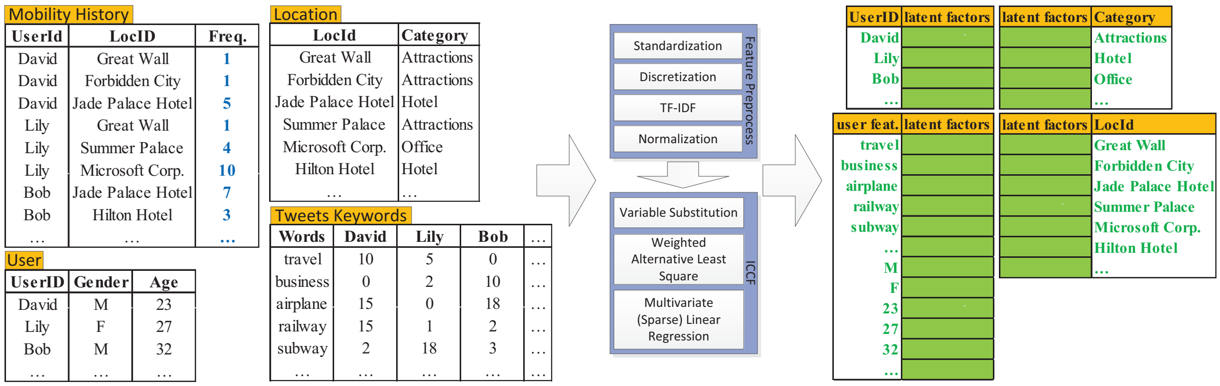


Fig. 1. The framework of ICCF

III. Implicit feedbacks based Content-aware Collaborative Filtering

Without any extension from ICF to incorporate the content 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 negative sample drawing. Additionally, it is similar to ICF, varying the preference confidence with the visit frequency, and assigning a lower confidence to the negative preference for unvisited locations. The overall framework is illustrated in Fig. 1, where users have features, such as profiles and text content, provided in social networks, such as Twitter and Facebook, and items have features, like category hierarchy. After performing tokenization on text content and discretizing continuous feature(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 fth feature of user u and $y_{i,l}$ is the value of the lth feature of location i. However, before feeding them into ICCF, these two feature matrices may required further preprocessing by applying tf-idf transformation and normalization (or standardization). After feeding them for learning ICCF, users and items, 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 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 perspective of extendability and explainability, as well as convergence rate.

A. Prediction and Loss function

As mentioned above, ICCF takes a user-location frequency matrix $\mathbf{C}$, 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 rating matrix $\mathbf{R}$ according to Eq. (1). It then follows MatchBox [21] and SVDFeature[19] to define the prediction preference of a user u for a location i as $\hat{r}_{u,i} = (\mathbf{p}_u + \mathbf{U}'\mathbf{x}_u)'(\mathbf{q}_i + \mathbf{V}'\mathbf{y}_i)$, where each row of the latent matrix $\mathbf{U} \in \mathbb{R}^{F\times K}$ and $\mathbf{V} \in \mathbb{R}^{L\times K}$ represent latent factors of features of user and item. Consequently, not only users and items, but also their features are mapped into a joint latent space, where inner product between them indicates one's preference for another. For example, 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 user u for restaurants. If the ids of both users and items 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 [18], going further, encapsulates $\{\tilde{\mathbf{x}}_u\}$ and $\{\tilde{\mathbf{y}}_i\}$ into unified feature vectors but it additionally allows the interaction between users/items and their features. However, such a representation is inappropriate to be adopted in this case, since it is difficult to satisfy that the preference confidence for different user-location pairs should be distinguished from each other according to the previous discussion. Based on the prediction function, an objective loss function, taking into account the varying confidence of preference with the visit frequency, is then formulated as follows:

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

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 matrix, and the necessity of developing a novel efficient optimization algorithm. This is because the objective function of existing frameworks is only dependent 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 alternative least square over each row of $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$. Below we will make a try in this way and elaborate the coupling difficulty of this approach between the optimization of latent factors of different features (i.e., different rows), incurred from the varying confidences with frequency and the overlap of user/item between different features. Taking the latter case 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/items 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 alternative least square, and the second of which learns the preference (i.e., $\mathbf{U}$ and $\mathbf{V}$) of user features and item features by means of multivariate (sparse) linear regression. The self preference of a user or an item (i.e., $\mathbf{p}_u$ and $\mathbf{q}_i$) is then obtained with a subtraction operation.

B. 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,

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

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

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

where $\mathbf{I}_K$ is an identity matrix of size $K \times K$. Although this yields the analytic solution, its time complexity depends on 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 $\mathcal{O}(\|\mathbf{R}\|_0 K^2)$ 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 the latent factor of one feature will change the term corresponding to the overlapped features, where the overlap 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 incurs from two reasons: the varying confidence with the visit frequency and the overlap of users/items between features. Analyzing the procedure w.r.t $\tilde{\mathbf{v}}_m$, the m^{th} row of $\tilde{\mathbf{V}}$, will yield a similar coupling conclusion. This coupling can be broken by splitting them into two stages, the first of which only depends on the former and the other of which only depends on the latter. This is achieved by a variable substitution technique to first learn the summed preference of users/items, according to the relationship between $\tilde{\mathbf{x}}_u$ and $\mathbf{x}_u$ and the relationship between $\tilde{\mathbf{y}}_i$ and $\mathbf{y}_i$,

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

The loss function in Eq.(2) is then converted into the following:

$$\mathscr{L} = \frac{1}{2}\sum_{u,i} w_{u,i}(r_{u,i} - \tilde{\mathbf{p}}_u'\tilde{\mathbf{q}}_i)^2 + \frac{\lambda}{2}\sum_u \|\tilde{\mathbf{p}}_u - \mathbf{U}'\mathbf{x}_u\|^2 + \frac{\lambda}{2}\sum_i \|\tilde{\mathbf{q}}_i - \mathbf{V}'\mathbf{y}_i\|^2 + \frac{\lambda}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2). \quad (3)$$

It is easy to see that this new loss function is quadratic with respect to any one of four types of variables $\{\tilde{\mathbf{p}}_u\}, \{\tilde{\mathbf{q}}_i\}$, $\mathbf{U}, \mathbf{V}$ and thus successfully decomposes the original coupling difficulty. Then alternative least square over these four types of variables can be exploited for optimization. Starting from $\{\tilde{\mathbf{p}}_u\}$, setting the gradient with respect to $\tilde{\mathbf{p}}_u$ to zero, we obtain the update rules for $\tilde{\mathbf{p}}_u$ as follows,

$$\tilde{\mathbf{p}}_u = (\tilde{\mathbf{Q}}'\mathbf{W}^u\tilde{\mathbf{Q}} + \lambda\mathbf{I}_K)^{-1}(\tilde{\mathbf{Q}}'\mathbf{W}^u\mathbf{r}_u + \lambda\mathbf{U}'\mathbf{x}_u), \quad (4)$$

where $\mathbf{W^u}$ is an $N \times N$ diagonal matrix, subject to $W_{i,i}^u = w_{u,i}$, $\mathbf{r}_u$ is a column rating vector of user u from the rating matrix $\mathbf{R}$, and $\tilde{\mathbf{Q}}$ is a matrix stacking all $\tilde{\mathbf{q}}_i$ by rows. Similarly, the update of latent factor $\tilde{\mathbf{q}}_i$ is given as:

$$\tilde{\mathbf{q}}_i = (\tilde{\mathbf{P}}'\mathbf{W}^i\tilde{\mathbf{P}} + \lambda\mathbf{I}_K)^{-1}(\tilde{\mathbf{P}}'\mathbf{W}^i\mathbf{r}_i + \lambda\mathbf{V}'\mathbf{y}_i) \quad (5)$$

where $\mathbf{W}^i$ is an $M \times M$ diagonal matrix, subject to $W_{u,u}^i = w_{u,i}$ and $\mathbf{r}_i$ is a column rating vector of location i.

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

$$\mathscr{F}(\mathbf{U}) = \frac{1}{2}\|\tilde{\mathbf{P}} - \mathbf{X}\mathbf{U}\|_F^2 + \frac{\gamma}{2}\|\mathbf{U}\|_F^2. \quad (6)$$

The optimal $\mathbf{U}$ of this objective function is

$$\mathbf{U} = (\mathbf{X}'\mathbf{X} + \gamma\mathbf{I}_F)^{-1}\mathbf{X}'\tilde{\mathbf{P}}. \quad (7)$$

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,

$$\mathbf{U} = \mathbf{X}'(\mathbf{X}\mathbf{X}' + \gamma\mathbf{I}_M)^{-1}\tilde{\mathbf{P}}. \quad (8)$$

If the matrix to be inverted in the dual solution is still of large size, an alternative solution is to apply conjugate gradient descent. Since it is a standard procedure, it will not be introduced here in detail, but note that it requires to be extended to a multivariate case.

Complexity Analysis. To update $\{\tilde{\mathbf{p}}_u\}$, making use of the trick mentioned in [22], i.e., $\tilde{\mathbf{Q}}'\mathbf{W}^u\tilde{\mathbf{Q}} = \tilde{\mathbf{Q}}'(\mathbf{W}^u - \mathbf{I})\tilde{\mathbf{Q}} + \tilde{\mathbf{Q}}'\tilde{\mathbf{Q}}$, each $\{\tilde{\mathbf{p}}_u\}$ can be updated in $\mathcal{O}(\|\mathbf{r}_u\|_0 K^2 + K^3)$ by assuming the inverse of a $K \times K$ matrix costs $\mathcal{O}(K^3)$ time. When updating $\{\tilde{\mathbf{p}}_u\}$ for all users in sequence, the total time is $\mathcal{O}(\|\mathbf{R}\|_0 K^2 + MK^3 + \|\mathbf{X}\|_0 K)$. Applying a similar trick for updating $\{\tilde{\mathbf{q}}_i\}$, it can be completed in $\mathcal{O}(\|\mathbf{R}\|_0 K^2 + NK^3 + \|\mathbf{Y}\|_0 K)$ time.

To update $\mathbf{U}$ based on conjugate gradient descent, the time complexity only depends 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. Similarly, based on conjugate gradient descent, updating $\mathbf{V}$ costs $\mathcal{O}(\|\mathbf{Y}\|_0 K\#iter)$.

In summary, the total time of one round of optimization w.r.t four types of variables is $\mathcal{O}\big((\|\mathbf{X}\|_0 + \|\mathbf{Y}\|_0)K\#iter + \|\mathbf{R}\|_0 K^2\big)$, where $\|\mathbf{R}\|_0 > \max(M,N) \times K$ is assumed satisfied. In other words, the time complexity of one round of optimization is in linear proportion to the number of non-zero entries in the user-item matrix, user-feature matrix, and item-feature matrix. Besides, parallel updating among $\{\tilde{\mathbf{p}}_u\}$ and among $\{\tilde{\mathbf{q}}_i\}$ is possible since there is no dependence between their individual update, so that in practice, the time complexity can be greatly reduced given multiple CPUs in a single machine or in a distributed computing environment, such as Hadoop and Spark.

C. Explanability

If the analytical update formulation of $\mathbf{U}$ is substituted back to the update of $\tilde{\mathbf{p}}_u$, we find that $\tilde{\mathbf{p}}_u$ is not only dependent on mobility data in terms of the user-location matrix, but also the latent factors of similar users, where similarity is measured as a function of dot product between their feature vectors. Therefore, it is directly correlated with matrix factorization on manifold by means of graph Laplacian regularization [24]. It also becomes possible to incorporate domain-specific similarity, such as user age proximity or document similarity between user tweets, into this framework by applying the kernel trick, i.e., replacing $\mathbf{X}\mathbf{X}'$ with a Gram matrix $\mathbf{K}$. In this case, the update of $\tilde{\mathbf{p}}_u$ becomes

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

where $\mathbf{k}_u$ is a column vector corresponding to the u^{th} row of matrix $\mathbf{K}$. The similarity in content 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 requires the inverse of a large sparse matrix and 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 kernel matrix using a method such as eigenvalue decomposition or Random Kitchen Sinks [29], whose feasibility is guaranteed by the Mercer theorem, which states that kernels can be expressed as an inner product in some Hilbert space [30]. Additionally, in some special cases, this formulation provides general rules or explanations for normalizing features. For example, document similarity is usually represented as 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.

D. Feature Selection

Since the optimization of features' latent factors is achieved by ℓ_2-regularized multivariate linear regression, it is intuitive to impose an ℓ_1-norm on them to obtain a sparse solution and thus select correlated features. Taking the object function with respect to $\mathbf{U}$ as an example, the objective function is defined as

$$\mathscr{L}(\mathbf{U}) = \overbrace{\frac{1}{2}\|\tilde{\mathbf{P}} - \mathbf{X}\mathbf{U}\|_F^2 + \frac{\gamma}{2}\|\mathbf{U}\|_F^2}^{\mathscr{F}(\mathbf{U})} + \rho\|\mathbf{U}\|_1.$$

Since a squared Frobenius norm and ℓ_1-norm is decomposable w.r.t each column, we can take turns to perform lasso regression directly on each column of $\tilde{\mathbf{P}}$ and $\mathbf{U}$ based on optimization algorithms such as coordinate descent [31], proximal gradient descent [32], interior-point methods [33], and so on. However, it is more efficient to develop lasso regression in a multivariate case since different columns can share matrix multiplication operations with each other. Due to the wide use of the proximal gradient descent, we make use of it to extend lasso toward a multivariate case. Proximal gradient descent is an iterative algorithm, generally used in convex optimization whose regularization penalty may not be differentiable. Based on a current estimate of parameters, proximal gradient descent first constructs a convex upper bound whose value at the current estimate equals the current objective value, and then minimizes the upper bound to get a new estimate of the parameters. Exploiting an upper bound proposed in [32], the parameters are updated as,

$$\mathbf{U}_{t+1} = S_{\rho\alpha}(\mathbf{U}_t - \alpha\nabla\mathscr{F}(\mathbf{U}_t)), \tag{9}$$

where $S_{\rho\alpha}(\mathbf{X})_{i,j} = \mathrm{sign}(x_{i,j}) \cdot \max(|x_{i,j}| - \rho\alpha, 0)$ is a soft threshold operator and $\alpha \in (0, \lambda_{\max}(\mathbf{X}'\mathbf{X}))$ should be satisfied to guarantee the convergence of the algorithm. $\lambda_{\max}(\mathbf{A})$ is the maximum eigenvalue of matrix $\mathbf{A}$, which may be difficult to compute for large-scale problems. Thus, we could make use of the backtracking step-size rule, which starts from an initial estimate, and then decays its value until the following inequality holds,

$$\|\mathbf{X}(\mathbf{U}_{t+1} - \mathbf{U}_t)\|_F^2 \leq (\alpha^{-1} - \gamma)\|\mathbf{U}_{t+1} - \mathbf{U}_t\|_F^2. \tag{10}$$

Based on the variant of proximal gradient descent, such as alternating direction method of multipliers, the convergence rate of algorithms could speed up, but they will be not elaborated here, since it is outside of the scope of this paper.

IV. EXPERIMENT

A. Dataset and Experimental setup

Dataset. ICCF was evaluated on a large-scale location-based social network dataset. This dataset was crawled from Jiepang, a Chinese location-based social network, similar to Foursquare, and spanned almost two years. Although check-ins are not publicly available on Jiepang itself, they are synchronized to other social networks, i.e., Weibo, so that they can be obtained by the open APIs of these social networks. We then select POIs that are visited by at least ten users and users who have been to at least ten distinct locations. Finally, 265,951 users and 189,850 POIs are then reserved, and the density of these users on these POIs is 3.69×10^{-4}.

Each user is linked to other social networks, which are able to collect rich semantic content, such as tweets and tags, and profile information, including age and gender, from users. We thus crawled this information with the hope of promoting the recommendation performance in the warm-start case and addressing the cold-start problem. According to statistics, 62.4% of users had gender information and 35.8% of users had age information. Since users also had profiles on Jiepang, we obtained them by crawling their profile pages and supplemented them into their Weibo profiles. After this, 100% of users had gender information while the portion of users with age information increased a little. Additionally, we also crawled each users' tags and tweets. We observed that each user had 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 sparse. For each tweet, we applied

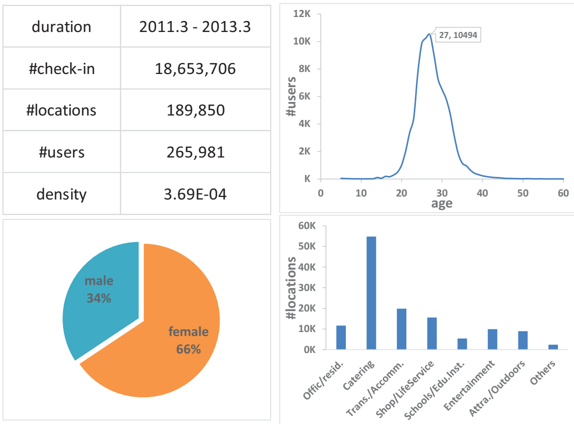


Fig. 2. Statistics information of the dataset

Jieba[1] to perform word segmentation. After that, we removed stop words and used tf-idf to choose the top 8,000 distinct words as vocabulary [34]. Thus each user had around 1.5K words on average. According to previous suggestions, we then performed ℓ_2-normalization on the tf-idf word vector for each user so that each user's word vector had a unit ℓ_2-norm. We also converted the tags of each user into a word vector, applied tf-idf to reduce the weight of more common tags, and then performed ℓ_2-normalization. Moreover, each location had two levels of category hierarchy, where the first level contained 8 coarser categories and the second level contained 157 finer categories. 67.2% of the locations have both coarser and finer categories. The overall statistics are presented in Fig 2.

Metric. We evaluated recommendation algorithms on sets of held-out visited locations. Presenting each user with the top k candidate locations sorted by their prediction preference, we assessed recommendation performance by checking how many of these locations actually appeared in each user's mobility data. Two widely used metrics, i.e., recall at k and precision at k, in prior work [3], [4], [7], were exploited. Formally, abbreviated as r@k and p@k, the recall at k and the precision at k are defined as,

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

where $\mathbb{S}_u(k)$ is the collection of top k recommended locations for a user u and $\mathbb{V}_u$ is the set of visited locations.

Framework. We conducted two types of evaluation. One was in-matrix recommendation, examining the improvement in recommendation by introducing profiles and content, and the other was out-matrix recommendation, testing the capability of handling the cold-start problem in these cases.

In the case of **in-matrix recommendation**, we randomly split mobility data of each user into five folds. For each fold, we fit a model to the other four folds and tested the within-fold locations for each user. We formed a predictive preference for the test set, generating a list of the top k recommended locations, and then calculate the metrics. After evaluating each fold, we reported averaged metrics.

[1]https://github.com/fxsjy/jieba

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, and then tested the recommendations for each user on the within-folds visited locations. Since each user in the test fold lacked training data, collaborative filtering failed in this case.

Based on these two schemes, the evaluation was conducted from three perspectives. First, we compared ICCF with LibFM, being a representation of explicit-feedback content-aware collaborative filtering algorithms, in four different settings, for location recommendation. Here, MatchBox was not chosen as a baseline as it is almost similar to LibFM. In addition, we underestimated the memory consumption of the provided toolkit [35] so that we didn't successfully obtain results on a personal computer with 16 gigabytes of memory when setting the dimension of latent space to 50. We next studied the effect of user profiles and content at promoting recommendation performance and their impact on recommendation in the cold-start case. Finally, we explained the effect of user profiles and content at performance gain.

Setting. The parameters of this model were set based on a grid search. The dimension of the latent space was set as 50 (it will be studied in later experiments) and the weight was set as $\alpha(c_{u,i}) = 1 + \log(1 + c_{u,i} \times 10^{\epsilon})$, where $\epsilon = 30$. When no features were taken into account, λ was set as 0.01; otherwise, it was set as 350. γ and ρ was set based on [25].

B. Experimental Results

Comparsions. We first compared ICCF with ICF and LibFM in four different settings. Since mobility data doesn't include negative samples, i.e., unattractive locations, training LibFM requires sampling negative ones from unvisited locations for each user. The first two settings, denoted as LibFM(c3-) and LibFM(c10-), respectively, vary the number of negative samples. In particular, a "3-" inside the parentheses indicates there were three times as many negative samples as positive ones. Since there were two tasks, i.e., classification and regression, provided in LibFM, we would like to examine their differences in recommendation. We thus compared LibFM(r10-) with LibFM(c10-) given the same number of negative items. In addition, we compared LibFM(c3-) with LibFM(c3-b) with no features included in order to inspect the effect of using user profiles and content. The results are shown in Fig 3(a)-3(b).

Based on the results, we **first** observed that ICCF outperformed LibFM of the best configuration by a significant margin. One of the reasons is that ICCF considers all unvisited locations as negative but assigns them a lower confidence for negative preferences while LibFM only samples some of them and treats them as equal to positive ones. Moreover, LibFM improved with the increasing number of negative samples, according to the comparison between LibFM(c3-) and LibFM(c10-). This confirms the effect of unvisited locations as negative samples at recommendation based on mobility data [14]. However, the increase of negative samples will reduce the efficiency of LibFM so that it is impractical to incorporate all unvisited locations. **Second**, incorporating user profiles and semantic content improved the performance of recommendation, by comparing ICCF with ICF and by comparing LibFM(c3-) with LibFM(c3-b). Thus, this information is useful for promoting recommendation performance. **Third**, regression-based LibFM greatly outperformed classification-based LibFM , which is consistent with the observation in [36]. This indicates regression-based loss objective plays an important role in top-k recommendation from implicit feedback.

In order to understand the superiority of ICCF, we are required to compare the basis of ICCF, i.e., ICF, with other types of matrix factorization, ranging from Bayesian Personalized Ranking Matrix Factorization [26], Max Margin Matrix Factorization [36][2] to Probabilistic Factor Model [38] based on Gamma and Poission matrix factorization(GaP) [27]. The comparison was performed on three datasets, including this mobility dataset, Last.fm [39], and MovieLens 10M ratings data[3], where the first two are naturally implicit feedback while the last one is converted by following [26] to predict whether a user is likely to rate a movie. In these three datasets, to clean the list of items, we only used items on which at least 10 users had one or more actions (checking in,listening,watching), and to clean the list of users, we only considered users who had one or more actions on at least 10 items. After the cleaning process, there were 358,587 users and 63,920 songs left on Last.fm, and 69,878 users and 9,708 movies left on MovieLens. The comparison results are shown in Fig 4(a)-(c), where all parameters of the baselines were tuned by a grid search.

Based on these figures, we found that ICF consistently and greatly outperformed other factorization models, although BPRMF also performed well for top-k recommendation. It is even possible to outperform ICF when using global AUC as a metric of recommendation since it optimizes this metric directly. However, optimizing global AUC is not equivalent to optimizing cut-off recall or precision, so it may not perform well for the top-k recommendation [40]. In addition, matrix factorization of directly optimizing top-k ranking measure doesn't necessarily perform as well as regression-based objectives, as observed in [36], let alone imposing the weights on the latter. However, its unsatisfactory performance at recommendation may depend on the implementation of MyMediateLite, which performs stochastic gradient descent based on hinge loss instead of logit loss compared to BPRMF. Although PFM can model the skewness of the visit frequency [15], [4] based on Poission distribution, it doesn't model the varying confidence for the negative and positive preferences so that it is also not as good as ICF and BPRMF. In a word, ICF is a comparatively suitable model-based recommendation algorithm for implicit feedback datasets, and thus confirms the motivation of designing content-aware recommendation algorithms based on it. Last but not least, we studied the effect of different weight settings, i.e., $\epsilon = 10$ and $\epsilon = 30$, corresponding to ICF(10) and ICF(30), respectively. Their comparison indicated it was likely to improve performance by imposing larger weights to visited locations in some cases.

Studying effect of profiles and content. After demonstrating the total effect of profiles and content, we still cannot comprehend their individual effects. Therefore, we examined them carefully from in-matrix recommendation and out-matrix recommendation perspectives. One thing to note is that we didn't observe the impact of tags on improving recommendation due to the extreme sparsity and thus didn't show the experimental results.

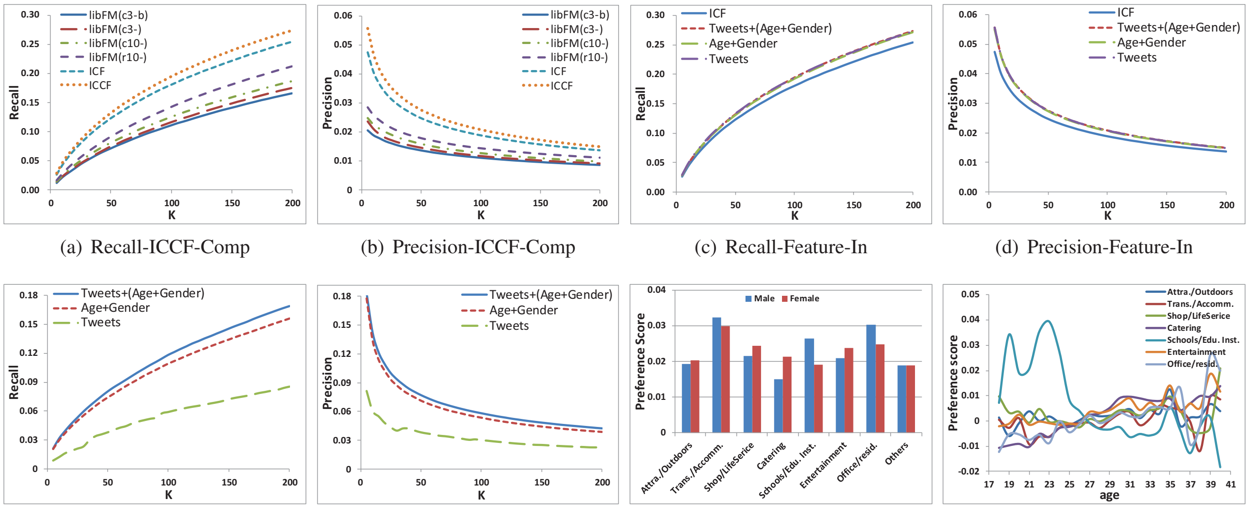
The results of in-matrix recommendation are shown in Fig 3(c)-3(d). Both profiles and content had an effect at improving the performance of recommendation individually compared to ICF. However, when they were integrated together, there was no further significant performance improvement. The reasons may be two fold: one is that profiles and content are correlated with each other; another is that regularized coefficients of these two types of latent factors are not optimally tuned since we have only searched over a small numerical range. This thus motivates Bayesian treatment for learning ICCF, which will be studied in our future research. Compared to the performance of ICF, the small improvement of the integrated one with two types of features implies limited information gain from these two types of knowledge about users.

Although it has been shown that profiles and 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 studied this case, showing the results in Fig 3(e)-3(f). Since collaborative filtering will fail in this case, it cannot be shown in these two figures. From them, we can see that both profiles and content of users are effective at recommending locations in the cold-start case, and that profiles are more effective than content. Moreover, when integrating them together, it makes further improvement in the recommendation performance, indicating they complement each other. Nevertheless, the impact of incorporating content with profiles is not as large as expected. One reason is that the content is not so correlated with recommending novel and attractive locations as profiles. Below, we will provide a detailed analysis.

Explaning effect of profiles and content. In order to deeply understand the amount of useful information that profiles and semantic content offer location recommendation, we analyzed the learned relationship in terms of dot product in the latent space between locations and user profiles and content. The results are shown in Fig 3(g)- 3(h), and Fig 5. We observed that male users preferred to show their visit to offices, residence, hotels, and educational institutions while female users were more likely to visit shops, entertainment places, restaurants, etc. Therefore, the male and the female have different preferences when visiting locations. Based on the relationship between age and visited locations, we found that young users (around 18-26 years) preferred to visit campus-related locations, such as teaching buildings, universities, and so on. This is because most of these users were students, living in and around campus. And it was more likely for users older than 26 years to visit restaurants and entertainment places, since such a visit to them is more interesting to share with their friends. However, the preferences of older users were much weaker than for younger ones. Finally, we measured the relationship between user tweets and locations by their dot product in the latent space and chose the top 100 correlated keywords. We then observed that most of the words associated with locations were geographical. Taking the locations of attractions and outdoors as an example, they

[2]We use the implementation of MyMediaLite[37]

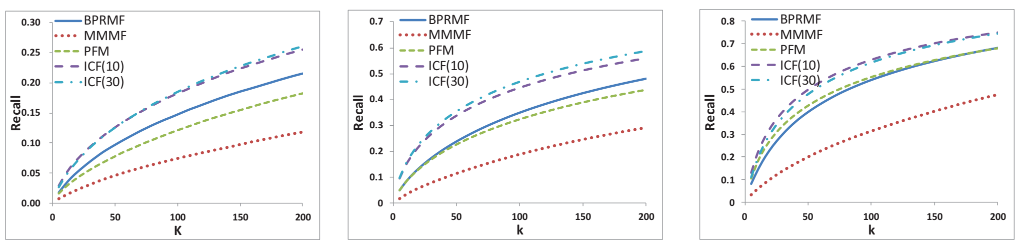
[3]http://grouplens.org/datasets/movielens/



(a) Recall-ICCF-Comp (b) Precision-ICCF-Comp (c) Recall-Feature-In (d) Precision-Feature-In

(e) Recall-Feature-Out (f) Precision-Feature-Out (g) Explanation-Gender (h) Explanation-Age

Fig. 3. Result of evaluation and explanation to the effect of user's profiles and content



(a) Jiepang (b) Last.fm (c) MovieLens

Fig. 4. Comparison between different matrix factorization models on different implicit feedback datasets

could be correlated with "railway stations", "services zone", "hotels", and so on. Therefore, such a correlation may not only be explicit but also implicit, indicating their effectiveness at promoting recommending performance and dealing with the cold-start case. Unfortunately, most of the important keywords were geographical, without making full use of other irrelevant keywords to locations, so that information gain brought from user content was not as large as expected. However, only 35% of users indicated their ages and there was no dependence between the features of users or items 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 user content.

C. Discussion and Future Work

We have previously mentioned that predicting a user's profiles based on her semantic content before learning ICCF can improve recommendation performance, thus indicating a promising research direction. Moreover, its regularized coefficients are manually tuned, costing lots of human efforts. Thus, a Bayesian treatment for learning ICCF should be important in putting it into wider use in recommendations based on implicit feedback. By studying the effect of imposing additional sparse constraints in terms of ℓ_1-norm regularization, we didn't observe significant improvement, although this may be because these parameters were not optimally tuned. Therefore, in future research, more focus should be placed on developing sparse learning for feature selections, such as sparse Bayesian learning (SBL), so that it is not only connected with the Baysian treatment to ICCF learning, but it is also appropriate for feature selection. Last but not least, we have shown that a user's features in terms of profiles and semantic content refine their similarity and thus the procedure of learning ICCF is closely connected to kernel learning. Since different types of features shape user similarity from different perspectives, it is possible to propose optimization algorithms for recommendations based on implicit feedback by multiple kernel learning.

V. Related Work

In this paper, we propose a novel and efficient content-aware collaborative filtering framework for recommendation on implicit feedback, and apply it for location recommendation on human mobility data. Thus, related work consists of two parts. One is location recommendation, and the other is 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 focus on recommending specific types of locations while others are generalized for any type of location. For example, Horozov et al. [41] developed a user-based collaborative filtering system to recommend

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

Fig. 5. Correlation between user's tweets and locations

restaurants to a user. Zheng et al. [42] designed a random walk style model for tourism hot spot recommendation. Zheng et al. considered location recommendation and activity recommendation together, so that they could provide location recommendation w.r.t. different types of activities [2], [43]. Ye et al. studied 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 [3]. Following this, more sophisticating models, such as jointly modeling geographical and social influence, and performing model-based collaborative filtering, e.g., matrix factorization [15], [4], [14], topic modeling [5], were proposed with the aim of seamless integration. In addition to geographical information, text content was also associated with many locations, since users often leave tips for commenting venues on location-based social networks after the visit to locations. Thus, some prior research exploited this content by topic modeling [13] and sentiment analysis [44], [12], and incorporated these text modeling techniques with collaborative filtering via collective matrix factorization, preference matrix refinement, regularization or empirical linear combination, and so on. Also, since mobility records are always time-stamped and people reveal different preferences at different times, recommendations can be time-aware [7], [8].

Compared with these methods, the differences from them lie in the following aspects. First, we mainly studied the effect of user content instead of location content on recommendation. User content should be more important than location content when addressing the cold start problem since it could be available earlier for inferring user interests. Second, we proposed a general framework for location recommendation based on human mobility data, which can incorporate any features without deep understanding into the factorization model. Such an objective is difficult to satisfy in prior work since the incorporation of any other feature requires expert knowledge for modifying the learning procedure. Third, prior research didn't take all the characteristics of implicit feedback into account and most of them needs to sample negatively preferred locations from unvisited ones.

Content-aware collaborative filtering is considered as a combination of content-based recommendation and collaborative filtering. In recent years, several general algorithms, including regression-based latent factor model [20], LibFM [18], MatchBox [21], and SVDFeature [19], have been proposed and showed almost equivalent in model representation but different from each other in optimization algorithms. For example, the first two algorithms made use of sampling methods for inferring latent factors while MatchBox leveraged approximate deterministic approaches for inference. Among the prior research, some methods have been implemented in open-source frameworks and widely used in many applications, such as music recommendation in KDDCup 2011, friendship prediction in KDDCup 2012, and so on. In addition to general algorithms taking different kinds of content, specific algorithms for taking text content of items have also been proposed, such as collaborative topic regression [34]. They have been exploited for news recommendation, scientific article discovery, etc.

In spite of the wide use, these algorithms are mainly designed for explicit feedback with both positively and negatively preferred samples, and optimized only over a small number of sampling entries from user-item matrices. The time complexity is in proportion to the number of sampling entries. However, due to only positively preferred items provided in implicit feedback, feeding them together with content into these existing frameworks requires to draw a comparable number of negatively preferred items with the positive ones for the sake of efficiency. In addition, according to prior empirical studies in searching for the best collaborative filtering for location recommendation from human mobility data, these frameworks without incorporating content could not achieve the comparable top-k performance to the best one. In contrast, the proposed algorithm targets content-aware collaborative filtering from implicit feedback and successfully addresses the disadvantages by taking all items without actions as negative while assigning them a lower confidence for negative preference, and achieving linear time optimization.

VI. Conclusions

In this paper, we first proposed an ICCF framework for content-aware collaborative filtering from implicit feedback datasets. It steered clear of sampling negative items, required by explicit-feedback based algorithms, and achieved efficient optimization, which is in linear proportion to the number of non-zero entries in user-item, user-feature and item-feature matrices. In ICCF, we showed that user features actually refined user similarity on mobility so that it is possible to incorporate domain-specific knowledge. We then applied ICCF for location recommendation on a large-scale LBSN dataset where users have gender, age, tweets, and tags. The experimental results indicated that ICCF was greatly superior to LibFM of the best configuration. By studying the effect of user profiles and semantic content, we found that they improved recommendation in the warm-start case and helped address the cold-start problem.

ACKNOWLEDGMENT

This research was partially supported by grants from the Natural Science Foundation of China (Grant No. 61502077) and the Fundamental Research Funds for the Central Universities of China (Grant No. ZYGX2015KYQD018).

REFERENCES

[1] C. R. Cloninger, T. R. Przybeck, and D. M. Svrakic, "The temperament and character inventory (tci): A guide to its development and use," 1994.

[2] 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*. AAAl Press, 2010.

[3] 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.

[4] 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.

[5] B. Hu and M. Ester, "Spatial topic modeling in online social media for location recommendation," in *Proceedings of RecSys'13*. ACM, 2013, pp. 25–32.

[6] J. Bao, Y. Zheng, and M. F. Mokbel, "Location-based and preference-aware recommendation using sparse geo-social networking data," in *Proceedings of GIS'12*, 2012, pp. 199–208.

[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*. 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. Lian and X. Xie, "Mining check-in history for personalized location naming," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 2, pp. 32:1–32:25, Apr. 2014.

[12] D. Yang, D. Zhang, Z. Yu, and Z. Wang, "A sentiment-enhanced personalized location recommendation system," in *Proceedings of the 24th ACM Conference on Hypertext and Social Media(HT'13)*. ACM, 2013, pp. 119–128.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

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

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

[22] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proceedings of ICDM'08*. IEEE, 2008, pp. 263–272.

[23] 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.

[24] 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.

[25] J. Liu, S. Ji, and J. Ye, "Slep: Sparse learning with efficient projections," *Arizona State University*, vol. 6, 2009.

[26] 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.

[27] J. Canny, "Gap: a factor model for discrete data," in *Proceedings of SIGIR'04*. ACM, 2004, pp. 122–129.

[28] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review*, vol. 13, no. 5-6, pp. 393–408, 1999.

[29] A. Rahimi and B. Recht, "Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning," in *Advances in neural information processing systems*, 2009, pp. 1313–1320.

[30] 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.

[31] T. T. Wu and K. Lange, "Coordinate descent algorithms for lasso penalized regression," *The Annals of Applied Statistics*, pp. 224–244, 2008.

[32] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[33] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l 1-regularized least squares," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 1, no. 4, pp. 606–617, 2007.

[34] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proceedings of KDD'11*. ACM, 2011, pp. 448–456.

[35] T. Minka, J. Winn, J. Guiver, S. Webster, Y. Zaykov, B. Yangel, A. Spengler, and J. Bronskill, "Infer.NET 2.6."

[36] M. Weimer, A. Karatzoglou, Q. V. Le, and A. Smola, "Maximum margin matrix factorization for collaborative ranking," *Advances in neural information processing systems*, 2007.

[37] G. Zeno, R. Steffen, F. Christoph, and S.-T. Lars, "MyMediaLite: A free recommender system library," in *Proceedings of RecSys 2011*, 2011.

[38] H. Ma, C. Liu, I. King, and M. R. Lyu, "Probabilistic factor models for web site recommendation," in *Proceedings of SIGIR'11*. ACM, 2011, pp. 265–274.

[39] O. Celma, *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer, 2010.

[40] S. J. Clémençcon and N. Vayatis, "Empirical performance maximization for linear rank statistics," in *Advances in neural information processing systems*, 2009, pp. 305–312.

[41] 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.

[42] 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.

[43] 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.

[44] H. Gao, J. Tang, X. Hu, and H. Liu, "Content-aware point of interest recommendation on location-based social networks." AAAI, 2015.