Collaborative Filtering Meets Mobile Recommendation: A User-centered Approach

Vincent W. Zheng[1], Bin Cao[1], Yu Zheng[2], Xing Xie[2] and Qiang Yang[1]
[1]Department of Computer Science and Engineering, Hong Kong University of Science and Technology
Clearwater Bay, Kowloon, Hong Kong, China
[2]Microsoft Research Asia, 4F, Sigma Building, No.49 Zhichun Road, Haidian District, Beijing 100190, China
{vincentz,caobin,qyang}@cse.ust.hk,{yuzheng,xingx}@microsoft.com

Abstract

With the increasing popularity of location tracking services such as GPS, more and more mobile data are being accumulated. Based on such data, a potentially useful service is to make timely and targeted recommendations for users on places where they might be interested to go and activities that they are likely to conduct. For example, a user arriving in Beijing might wonder where to visit and what she can do around the Forbidden City. A key challenge for such recommendation problems is that the data we have on each individual user might be very limited, while to make useful and accurate recommendations, we need extensive annotated location and activity information from user trace data. In this paper, we present a new approach, known as user-centered collaborative location and activity filtering (UCLAF), to pull many users' data together and apply collaborative filtering to find like-minded users and like-patterned activities at different locations. We model the user-location-activity relations with a tensor representation, and propose a regularized tensor and matrix decomposition solution which can better address the sparse data problem in mobile information retrieval. We empirically evaluate UCLAF using a real-world GPS dataset collected from 164 users over 2.5 years, and showed that our system can outperform several state-of-the-art solutions to the problem.

Introduction

Today, many mobile devices come with positioning functions such as Geographical Positioning System (GPS) and sensors. Through these devices, many new services can be provided for users, including geospatial Web services (Simon and Fröhlich 2007), location-based activity recognition (Liao, Fox, and Kautz 2005), etc. Much of the location data are accumulated in the form of location traces and user activities. For example, Figure 1 shows a GPS location data management system used by a visitor, who can annotate her GPS trajectory at the Forbidden City area in Beijing with some of her own comments (depicted as small pink boxes attached on the trace). These annotations can indicate the activities that she conducted at various interesting locations. With more and more user trajectories and annotations, our question is: how can we make use of these data to make useful and targeted recommendations for the users?

In this paper, we aim to mine useful knowledge from many users' GPS trajectories based on their partial location and activity annotations to provide targeted collaborative location and activity recommendations for each user. In particular, we try to answer the following questions: for a specific user, if she wishes to do some sightseeing or food-hunting in a large city such as Beijing, where should she go, given her previous GPS traces and other similar users' GPS histories? Also, if she has already visited some places such as the Forbidden City palaces, what else can she do in that area? The first question corresponds to location recommendations given some activity queries, where an activity may refer to various human behaviors such as food-hunting, shopping, watching movies/shows, enjoying sports/exercises, tourism, etc. The second query corresponds to activity recommendations from some given locations. In this paper, we answer both questions with user-centered collaborative location and activity filtering (UCLAF).

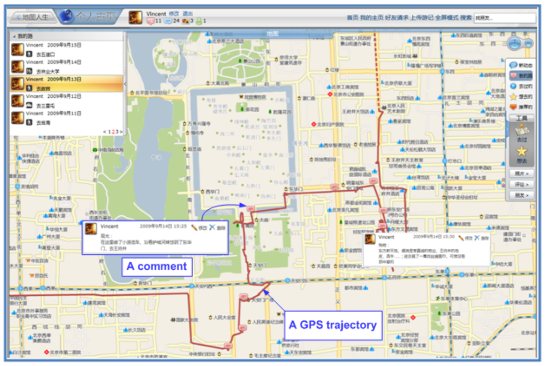


Figure 1: GPS data management services.

Our work is motivated by the observation that location recommendation and activity recommendation are tightly related in nature, since the activities are usually location-dependent. However a major challenge is that, the ratings provided by users in the form of annotated location-activity data are often *grossly insufficient and sparse*. To solve this problem, when making recommendations, we need to take many users' information into account. This requires us to make our inferences collaboratively, instead of separately, based on many users' data. Therefore, we propose a collaborative location and activity filtering framework, where we take user as a first-class entity in the data modeling. In particular, we model the user-location-activity relations in a tensor, and formulate the reasoning problem as a collaborative filtering (CF) problem. We can mine additional information, such as the user-user similarities, location features (e.g. points of interests in the area) and activity-activity correlations to help with the CF problem. Then, we exploit a regularized tensor and matrix decomposition method to discover interesting locations and activities for our recommendations. Our method is evaluated on a real-world GPS dataset, which was collected from 164 users over a period of 2.5 year and with a total GPS trajectory length over 139,310 kilometers. We will show that our method can outperform many state-of-art solutions to the recommendation problem.

Related Work

In the past, few work has been done on collaborative location and activity recommendations. Most of the previous work focused on either recommending some specific types of locations, such as shops (Takeuchi and Sugimoto 2006) restaurants (Horozov, Narasimhan, and Vasudevan 2006; Park, Hong, and Cho 2007), and hot spots for tourism (Zheng et al. 2009); or only recognizing the user activities from sensor data rather than providing location and activity recommendations together (Bui 2003; Patterson et al. 2005; Zheng, Hu, and Yang 2009).

In location recommendation, for example, a CityVoyager system (Takeuchi and Sugimoto 2006) uses an item-based CF method to recommend to a user some shops that are similar to her previously visited shops. In (Horozov, Narasimhan, and Vasudevan 2006), a Geowhiz system uses a user-based CF method to recommend to a user some restaurants that some other similar users usually visited. In (Zheng et al. 2009), a HITS-based model is proposed to recommend the tourism hot spots that are popular and highly recommended by the experienced users. Compared with these single type location recommenders, our system is able to handle multiple location types w.r.t. different activity queries.

For activity recommendation, our work is related to the study for activity recognition, which aims to infer what a user is doing from various sensor observations. For example, in (Liao, Fox, and Kautz 2005), a hierarchical conditional random field model is built based on GPS data to recognize whether a user is at work, or sleeping at home, etc. In (Wyatt, Philipose, and Choudhury 2005), some object use common sense knowledge is mined from the Web to help recognize the activities of daily living such as brushing teeth. But our activity recommendation is different from pure activity recognition because: first, rather than recognizing the activity for a user in real time, we aim to find all the interesting activities that can be performed at some locations; second, we collaboratively model the users other than treat them independently; third, our solution can meanwhile help accomplish the location recommendation task.

Our previous work (Zheng et al. 2010) presented a possible solution to conduct location recommendation and activity recommendation together by merging all the users' ratings together. However, this solution only considered the location-activity relations without modeling users specifically, and thus made the same recommendations to different users. In this work, we extend our previous method by taking users into account. We carefully model the user-location-activity relations, so that we can provide more targeted recommendations to each user.

User-Location-Activity Modeling with Tensor

From the GPS data, we can extract three entities, i.e. users, locations and activities, denoting that some user visited some place and did something there. We propose to model such user-location-activity relations in a 3-d tensor, with each dimension corresponding to an entity above. In particular, we denote such a tensor as $\mathcal{A} \in \mathbb{R}^{m \times n \times r}$, where m is the number of users, n is the number of locations and r is the number of activities. Then an entry a_{ijk} in $\mathcal{A}$ denotes the frequency of a user i visiting location j and doing activity k there. As each user has limited number of annotations on the locations and activities (so we have no idea what a user was doing at some places), the tensor constructed from the GPS data is expected to be sparse. This inspires us to use collaborative filtering to fill the tensor for recommendations.

One possible solution to the CF problem is to follow the memory-based methods, such as (Herlocker et al. 1999; Wang, de Vries, and Reinders 2006), and design some similarity metrics based on user and/or location and/or activity for filling the tensor entries. A drawback of such methods is that, in general, the similarity metrics are not adaptive to different datasets or contain some parameters that should be tuned other than learned. So we follow another direction to solve the CF problem, by proposing a model-based method. We will show in the experiments that our method can outperform the memory-based methods. The model-based methods benefit from the statistical and machine learning techniques, and view CF as a missing-value prediction problem through matrix factorization (Srebro and Jaakkola 2003; Singh and Gordon 2008). However, most of these methods only modeled two-party relations in matrix forms. In contrast, we model the three-party relations in a tensor; and beyond standard tensor decomposition (Lathauwer, Moor, and Vandewalle 2000), we will further exploit additional matrix inputs for a regularized decomposition solution.

Our Solution

The main idea of our model is illustrated in Figure 2. We first model the user-location-activity relations in a tensor $\mathcal{A}$. As the tensor can be sparse, our objective is to fill the tensor for location and activity recommendations. To help the collaborative filtering in the tensor, we exploit some additional information w.r.t. each tensor entity. In particular, we have the user-user matrix $B \in \mathbb{R}^{m \times m}$ which encodes the user-user similarities in a social network. We aim to use this similarity information to uncover the like-minded users in CF. We also have a location-feature matrix $C \in \mathbb{R}^{n \times p}$, with each feature referring to the (normalized) number of POIs (point of interests, e.g. museums) in that location (Zheng et al. 2009). For activities, we have a matrix $D \in \mathbb{R}^{r \times r}$ representing the activity-activity correlations showing how likely

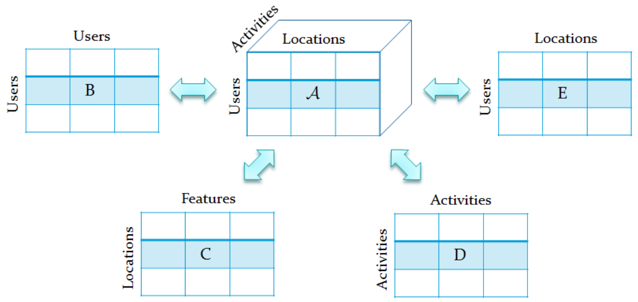


Figure 2: Model illustration in a tensor/matrix form.

an activity may happen if another activity happens. This matrix can be obtained by using the common sense on the Web, with some search-engine based similarity methods proposed in (Zheng, Hu, and Yang 2009) and (Wyatt, Philipose, and Choudhury 2005). Beyond the tensor to model the user-location-activity, we also extract another matrix $E \in \mathbb{R}^{m \times n}$ from the GPS data to model the user-location visiting relations. This matrix could be helpful to model the case when we only know a user visited some place but have no idea what she was doing there. Note that, all these four matrices are given beforehand, and taken as inputs for our model. We will study the impacts of these matrices in the experiments by tuning their corresponding model parameters.

To fill in entries in the tensor $\mathcal{A}$, we follow the model-based methods (Srebro and Jaakkola 2003; Singh and Gordon 2008) to decompose the tensor $\mathcal{A}$ for some low-dimensional representations w.r.t. each tensor entity (i.e. users, locations and activities) based on the incomplete tensor (as it is sparse). In decomposition, we force the low-dimensional representations to be shared with the additional matrices so as utilize their information. After such low-dimensional representations are obtained, we can reconstruct the tensor by filling all the missing entries in the tensor. In our model, we propose a PARAFAC-style tensor decomposition (Cichocki et al. 2009) framework to integrate the tensor with the additional matrices for a regularized decomposition. Specifically, our objective function is

$$\begin{aligned} \mathcal{L}(X,Y,Z,U) &= \tfrac{1}{2}\left\|\mathcal{A} - [\![X,Y,Z]\!]\right\|^2 + \tfrac{\lambda_1}{2}\mathrm{tr}(X^T L_B X) \\ &+ \tfrac{\lambda_2}{2}\left\|C - YU^T\right\|^2 + \tfrac{\lambda_3}{2}\mathrm{tr}(Z^T L_D Z) + \tfrac{\lambda_4}{2}\left\|E - XY^T\right\|^2 \\ &+ \tfrac{\lambda_5}{2}(\|X\|^2 + \|Y\|^2 + \|Z\|^2 + \|U\|^2), \end{aligned} \tag{1}$$

where the variables $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k] \in \mathbb{R}^{m \times k}$, $Y = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k] \in \mathbb{R}^{n \times k}$ and $Z = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k] \in \mathbb{R}^{r \times k}$. $[\![X, Y, Z]\!] = \sum_{i=1}^{k} \mathbf{x}_i \circ \mathbf{y}_i \circ \mathbf{z}_i$, where "$\circ$" denotes the outer product. Another variable $U \in \mathbb{R}^{p \times k}$. L_B is the Laplacian matrix of B, defined as $L_B = Q - B$ with Q being a diagonal matrix whose diagonal entries $Q_{ii} = \sum_j B_{ij}$. $tr(\cdot)$ denotes the trace of a matrx. L_D is the Laplacian matrix of D. $\|\cdot\|$ denotes the Forbenius norm. $\lambda_1 \sim \lambda_5$ are model parameters, when $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0$, our model degenerates to the standard PARAFAC tensor decomposition, showing that our model is more flexible to utilize other information about the targeted entities.

In (1), the first term decomposes the user-location-activity tensor $\mathcal{A}$ as an outer-product of three low-dimensional representations w.r.t. each entity (i.e. X for the users, Y for the locations and Z for the activities). The second term poses a regularization term on the users, forcing the low-dimensional representations of two users as close as possible if they are similar as suggested by the additional information. The third term borrows the similar idea with collective matrix factorization (Singh and Gordon 2008), by sharing the low-dimensional location representation Y with the tensor decomposition. The fourth term is a regularization term similar to the second term, forcing the low-dimensional representations of two activities as close as possible w.r.t. their correlations. The fifth term is similar to the third term, and shares the low-dimensional user representations X and location representations Y with the tensor decomposition. The last term is a regularization term so as to avoid overfitting.

In general, there is no closed form solution for Eq.(1), so we turn to numerical methods, such as gradient descent (Singh and Gordon 2008), to solve the problem. By taking the derivatives over the objective function, we have

$$\begin{aligned} \nabla_X \mathcal{L} &= -A^{(1)}(Z * Y) + X\left[(Z^T Z) \odot (Y^T Y)\right] \\ &\quad + \lambda_1 L_B X + \lambda_4 (XY^T - E)Y + \lambda_5 X, \\ \nabla_Y \mathcal{L} &= -A^{(2)}(Z * X) + Y\left[(Z^T Z) \odot (X^T X)\right] \\ &\quad + \lambda_2 (YU^T - C)U + \lambda_4 (XY^T - E)^T X + \lambda_5 Y, \\ \nabla_Z \mathcal{L} &= -A^{(3)}(Y * X) + Z\left[(Y^T Y) \odot (X^T X)\right] \\ &\quad + \lambda_3 L_D Z + \lambda_5 Z, \\ \nabla_U \mathcal{L} &= \lambda_2 (YU^T - C)^T Y + \lambda_5 U, \end{aligned} \tag{2}$$

where $A^{(i)}$ denotes the mode-i tensor unfolding with $A^{(1)} \in \mathbb{R}^{m \times nr}, A^{(2)} \in \mathbb{R}^{n \times mr}, A^{(3)} \in \mathbb{R}^{r \times mn}$. In particular, a tensor entry $a_{i_1 i_2 i_3}$ has a corresponding position (i_n, j) in each mode's unfolding: for mode-1, $j = i_2 + (i_3 - 1)n$; for mode-2, $j = i_1 + (i_3 - 1)m$; for mode-3, $j = i_1 + (i_2 - 1)m$. Besides, "$*$" denotes the Khatri-Rao product: for two matrices $V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_J] \in \mathbb{R}^{I \times J}$ and $W = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_J] \in \mathbb{R}^{T \times J}$, their Khatri-Rao product is defined as $V * W = [\mathbf{v}_1 \otimes \mathbf{w}_1, \mathbf{v}_2 \otimes \mathbf{w}_2, \cdots, \mathbf{v}_J \otimes \mathbf{w}_J] \in \mathbb{R}^{IT \times J}$, where "$\otimes$" denotes the Kronecker product. "$\odot$" denotes the Hadamard product (or, entry-wise product).

Algorithm and Its Complexity

Given the incomplete tensor and additional information matrices, our goal is to complete the tensor for output. As shown in Algorithm 1, we use an iterative algorithm to solve the problem. In each iteration, we calculate the gradients for the objective function $\mathcal{L}$ according to (2), and then get the updated objective function until it reaches the minimum. In the algorithm, T is the maximal number of iterations and γ is the step size. We set $T = 1000$ and $\gamma = 0.0001$ in our experiments. Finally, after the iteration stops, we will reconstruct the user-location-activity tensor $\hat{\mathcal{A}}$ by using the low-dimensional representations X, Y, Z. To do recommendations, for an existing user u $(1 \leq u \leq m)$, we have her location-activity matrix as $G \in \mathbb{R}^{n \times r}$ with $G(l, a) = \hat{\mathcal{A}}(u, l, a)$ for $1 \leq l \leq n, 1 \leq a \leq r$. Then we are ready for recommendations: given a location input l, we will

Algorithm 1 The UCLAF Algorithm

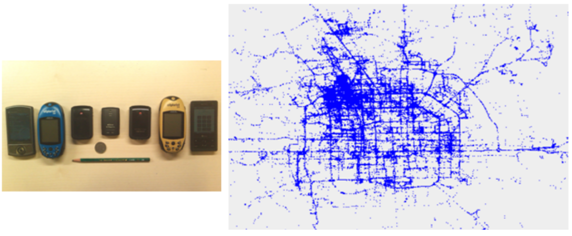
```
Input: An incomplete location-activity-user tensor $\mathcal{A}$ and four additional information matrices $B, C, D, E$.
Output: The complete tensor for $\mathcal{A}$, denoted by $\hat{\mathcal{A}}$.
begin
  $t = 1$;
  while $t < T$ and $\mathcal{L}_t > \mathcal{L}_{t+1}$ do
    Get the gradients $\nabla_X$, $\nabla_Y$, $\nabla_Z$ and $\nabla_U$ by Eq.(2);
    $X_{t+1} = X_t - \gamma\nabla_X$, $Y_{t+1} = Y_t - \gamma\nabla_Y$, $Z_{t+1} = Z_t - \gamma\nabla_Z$, $U_{t+1} = U_t - \gamma\nabla_U$;
    if $\mathcal{L}_t < \mathcal{L}_{t+1}$ then break; end if
    $t = t + 1$;
  end while
  return $\hat{\mathcal{A}} = [\![X_t, Y_t, Z_t]\!]$.
end
```

rank the l-th row of S in a decreasing order, and recommend to user u some top activities to do there. Similarly, given an activity input a, we will rank the a-th column of G in a decreasing order, and recommend to user u some top locations to go. We can also recommend to a new user in a similar way, except that we will construct such a location-activity matrix by summing up the tensor along its user dimension: $G' \in \mathbb{R}^{n\times r}$ with $G'(l,a) = \sum_{u=1}^{m} \hat{\mathcal{A}}(u,l,a)$.

Let us consider the algorithm's complexity. At each iteration, to get the gradient ∇_X, we spend $O(mnr + m^2)$ w.r.t. the input data dimensions m, n, r. Similarly, for ∇_Y, we have $O(mnr)$; for ∇_Z, we have $O(mnr)$; for ∇_U, we have $O(n)$. Thus in total, for getting the gradients, we spend $O(mnr + m^2 + mnr + mnr + n) = O(mnr + m^2)$. To calculate the objective function, we spend $O(mnr + m^2 + r^2)$. As the maximal number of iterations is constant, the total complexity of this algorithm is $O(T \times (mnr + m^2 + r^2)) = O(mnr + m^2 + r^2)$, showing that our algorithm is efficient.

Experiments

We have 164 users who carried GPS devices to record their trajectories from April 2007 to Oct. 2009 in the city of Beijing, China, as shown in Figure 3. The dataset consists of 12,765 GPS trajectories with a total length over 139,310 kilometers. To make sure that we recommend meaningful locations and activities, we remove the GPS points for work places and homes, and consider 5 different types of activities in this study, including “Food & Drink”, “Shopping”, “Movies & Shows”, “Sports & Exercise” and “Tourism & Amusement”. In addition, we also follow (Zheng et al. 2009) to cluster the raw GPS points into 168 meaningful locations for recommendation. So in the experiments, the number of users $m = 164$, the number of locations $n = 168$, the number of activities $r = 5$. Besides, the number of location features $p = 14$. The user comments attached to the GPS data were manually parsed into activity annotations for the 168 locations. These annotations were used to construct the user-location-activity tensor: if a user performed an activity at some location, the corresponding entry in the tensor will be added by 1; otherwise, the entry is 0. After this tensor construction, 1.04% of the entries are larger than 0.



(A) GPS devices (B) Data distribution in Beijing

Figure 3: GPS devices and data (Zheng et al. 2010).

Our system works as follows: a user can log in the system and enter some query, such as “Bird's Nest” for activity recommendation; then the system, according to her GPS history and other users' experiences, will recommend to the user a ranking list of activities with “Tourism & Amusement > Sports & Exercise > . . . ”. Similarly, when the user enters “Tourism” for location recommendation, the system will output a ranking list of famous Beijing tourism spots “Summer Palace > Forbidden City > . . . ”.

Comparison with Baselines

We employ 5 baselines for comparison: user-based CF (UCF), location-based CF (LCF), activity-based CF (ACF), unifying user-location-activity CF (ULA), high-order singular value decomposition (HOSVD). The first 3 baselines (i.e. UCF, LCF and ACF) are memory-based methods, adapted from (Herlocker et al. 1999) for tensor CF and taking only tensr as input. The fourth baseline, ULA, is also a memory-based method, adapted from (Wang, de Vries, and Reinders 2006) to take both the tensor and the additional matrices into consideration. The fifth baseline, HOSVD, is a model-based method used in (Symeonidis, Nanopoulos, and Manolopoulos 2008) to model the user-item-tag relations for tag recommendation. It only takes the tensor information as input.

In particular, for UCF, we consider CF on each user-location matrix w.r.t. each activity independently. On each matrix, we follow (Herlocker et al. 1999) and use Pearson correlation as the user similarity weights. We find the top N similar users for some target user (with missing entries) and then compute their weighted average to predict the missing entry. Similarly, we have LCF and ACF by considering CF on each location-activity matrix w.r.t. each user individually. In the experiments, we set $N = 4$, since we find that the prediction results do not depend on N significantly.

In ULA, for each missing entry in the tensor, we will extract a set of top N_u similar users, top N_l similar locations and top N_a similar activities, and then use the ratings from all these users on the corresponding locations and activities, in a weighted manner to calculate the entry value. Specifically, we adopt the idea of (Wang, de Vries, and Reinders 2006) and design the prediction function as

$$\hat{\mathcal{A}}_{i,j,k} = \frac{\sum_{u\in R_i} S_{u,i}\mathcal{A}_{u,j,k}}{4\sum_u S_{u,i}} + \frac{\sum_{l\in R_j} S_{l,j}\mathcal{A}_{i,l,k}}{4\sum_l S_{l,j}} + \frac{\sum_{a\in R_k} S_{a,k}\mathcal{A}_{i,j,a}}{4\sum_a S_{a,k}} + \frac{\sum_{u\in R_i, l\in R_j, a\in R_k} S_{u,l,a}\mathcal{A}_{u,l,a}}{4\sum_{u,l,a} S_{u,l,a}},$$

where $S_{u,i}$ is the similarity for users i and u learned from the user-user matrix B; $S_{l,j}$ is the similarity for locations j and l learned from the location-feature matrix C and the user-location matrix E by equally combining the cosine similarities calculated from each; $S_{a,k}$ is the similarity for activities k and a learned from activity-activity matrix D; $S_{u,l,a}$ is the similarity between $\mathcal{A}_{i,j,k}$ and $\mathcal{A}_{u,l,a}$ for some u, l, a belong to the neighboring sets R_i, R_j, R_k of user i, location j and activity k, respectively. It's designed as

$$S_{u,l,a} = 1/\sqrt{(1/S_{u,i})^2 + (1/S_{l,j})^2 + (1/S_{a,k})^2}.$$

In the experiments, we also set $N_u = N_l = N_a = 4$, as similar to the previous cases.

We report the comparison results in Table 1. To have these results, we randomly split 50% of the tensor data for training and hold out the other 50% for testing. For all the comparisons here, we run for 5 times to generate the mean values and standard deviations of the results. In our model, we set the model parameters $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = 0.1$ and the low dimension $k = 4$. We will study the impact of the parameters in the next section. For HOSVD, we preserved 30% of the information in the original tensor for dimensionality reduction, as suggested in the experiments of (Symeonidis, Nanopoulos, and Manolopoulos 2008). For evaluation, we employ two metrics; one is RMSE (root mean square error) to measure the tensor reconstruction loss on the hold-out test data. For RMSE, the smaller, the better. The other metric is nDCG (normalized discounted cumulative gain) (Zheng et al. 2009) to measure the ranking results of our retrieved location/activity list. For location recommendation to some user given some activity query, we first rank the locations that were held out in the test data for current user and activity. Then, we compare this ranking with the optimal ranking as suggested by the test data to generate the nDCG value. Finally, we take the average the nDCG values over all the users and activities to output the results in the "nDCG$_{loc}$" column at Table 1. Similarly, we have the results for activity recommendation in the "nDCG$_{act}$" column. For nDCG, the larger, the better.

	RMSE	nDCG$_{loc}$	nDCG$_{act}$
UCF	0.027 ± 0.006	0.297 ± 0.024	0.807 ± 0.007
LCF	0.009 ± 0.000	0.532 ± 0.021	0.614 ± 0.019
ACF	0.022 ± 0.005	0.408 ± 0.012	0.785 ± 0.006
ULA	0.015 ± 0.003	0.291 ± 0.022	0.799 ± 0.012
HOSVD	0.006 ± 0.001	0.390 ± 0.021	0.913 ± 0.004
UCLAF	$\mathbf{0.006 \pm 0.001}$	$\mathbf{0.599 \pm 0.036}$	$\mathbf{0.959 \pm 0.009}$

Table 1: Comparison with baselines, by "mean ± std".

As we can see from the table, our model consistently outperforms the other baselines, showing the effectiveness of our modeling with tensor and incorporating additional information for collaborative location and activity recommendations. We also note that the nDCG for activity recommendation is usually higher than that for location recommendation. This is because the number of activities is much smaller than that of locations for ranking, especially when we measure the rankings over some part of the activities/locations on the test data. Besides, we found that ULA does not necessarily deliver better results. This implies that our designed prediction function does not model the data characteristics perfectly, thus encouraging us to study more sophisticated memory-based methods for comparison. Also note that, the nDCG values shown in the table are not necessarily close to our previous results in (Zheng et al. 2010), as they are tested with different datasets and experimental conditions.

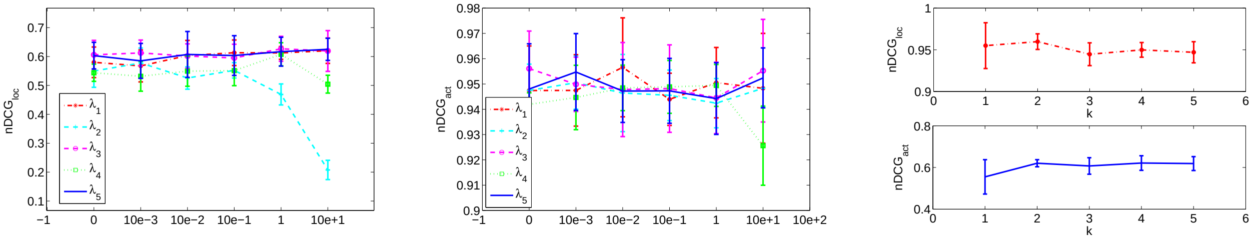
Impact of the Model Parameters

We first study the impact of $\lambda_1 \sim \lambda_5$. Recall the objective function in Eq.(1), where each λ_i ($i = 1, ..., 4$) controls the contribution for each additional information. We vary each λ_i from 0 to 10, and report the nDCG values for location recommendation in Figure 4(a) and activity recommendation in Figure 4(b). As can be seen from both plots, in general, using the additional information (when λ_i's are larger than 0) could be better than not using it (when λ_i's equal to 0). Besides, our model is not sensitive to most of these parameters. This implies that, the information in the additional matrices of this dataset could be limited, and meanwhile, our model is robust. We also notice that, as λ_2 increases, the nDCG for location recommendation decreases quickly. This could be because the location-feature matrix is noisy in data extraction from the POI database, as the λ_2 increases, the impact of the noise becomes bigger. As this location-feature matrix is not directly related to activities, we don't observe similar performance decrease in activity recommendation.

In Figure 4(c), we vary the low dimension k from 1 to 5 (as the minimal dimension in the tensor is 5, i.e. the number of activities), and report the nDCG results. As can be seen, our model's performances, for both location recommendation and activity recommendation, are insensitive to the change of k. Due to space limit, we didn't report the results under RMSE for all the parameters (i.e. λ_i's and k), but we observed the similar patterns from experiments.

Conclusion and Future Work

In this paper, we developed a novel user-centered approach to mine knowledge from GPS trajectory data to make mobile recommendations on locations and activities. With the recommendation system, we can answer two typical questions in our daily life. The first question is about location recommendation: if we want to do something, where shall we go? The second question is about activity recommendation: if we plan to visit some place, what can we do there? We show that these two questions are inherently related, as they can be seen as a collaborative filtering problem in a user-location-activity rating tensor. We designed a user-centered collaborative location and activity filtering algorithm, based on regularized tensor and matrix decomposition, to solve the problem. Our model is flexible to exploit additional information about the targeted entities to enhance the system performance. We evaluated our system on a real-world GPS dataset, and showed on average 19% improvement on location recommendation and 22% improvement on activity recommendation over the simple memory-based collaborative filtering baselines (i.e. UCF, LCF and ACF). In the future,



(a) Impact of λ_i's to location recommend.

(b) Impact of λ_i's to activity recommend.

(c) Impact of the low dimension k

Figure 4: Impact of model parameters.

we will consider how to update our model online as more users accumulate data continuously. We would also like to consider other potentially useful information in our recommendation algorithms.

Acknowledgments

This research was supported by grants from Microsoft Research Asia MRA08/09.EG03 and CERG/NSFC N_HKUST 624/09. The authors also would like to thank the reviewers for their helpful comments and suggestions.

References

Bui, H. H. 2003. A general model for online probabilistic plan recognition. In *IJCAI'03: Proc. of the 18th International Joint Conf. on Artificial Intelligence*, 1309–1318. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Cichocki, A.; Zdunek, R.; Phan, A. H.; and ichi Amari, S. 2009. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multiway Data Analysis and Blind Source Separation*. Wiley.

Herlocker, J. L.; Konstan, J. A.; Borchers, A.; and Riedl, J. 1999. An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proc. of the 22nd Annual ACM SIGIR Conference*, 230–237. New York, NY, USA: ACM.

Horozov, T.; Narasimhan, N.; and Vasudevan, V. 2006. Using location for personalized poi recommendations in mobile environments. In *SAINT '06: Proc. of the International Symposium on Applications on Internet*, 124–129. Washington, DC, USA: IEEE Computer Society.

Lathauwer, L. D.; Moor, B. D.; and Vandewalle, J. 2000. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications* 21(4):1253–1278.

Liao, L.; Fox, D.; and Kautz, H. 2005. Location-based activity recognition. In Weiss, Y.; Schölkopf, B.; and Platt, J., eds., *NIPS '05: Advances in Neural Information Processing Systems 18*, 787–794. Cambridge, MA: MIT Press.

Park, M.-H.; Hong, J.-H.; and Cho, S.-B. 2007. Location-based recommend-ation system using bayesian user's preference model in mobile devices. In *UIC '07: Proc. of Ubiquitous Intelligence and Computing*, 1130–1139. Publisher Springer Berlin / Heidelberg.

Patterson, D. J.; Fox, D.; Kautz, H. A.; and Philipose, M. 2005. Fine-grained activity recognition by aggregating abstract object usage. In *ISWC '05: Proc. of the 9th IEEE International Symposium on Wearable Computers*, 44–51. Washington, DC, USA: IEEE Computer Society.

Simon, R., and Fröhlich, P. 2007. A mobile application framework for the geospatial web. In *WWW '07: Proc. of the 16th International Conference on World Wide Web*, 381–390. New York, NY, USA: ACM.

Singh, A. P., and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *KDD '08: Proc. of the 14th ACM SIGKDD Intl. Conf. on Knowledge Discovery & Data Mining*, 650–658. New York, NY, USA: ACM.

Srebro, N., and Jaakkola, T. 2003. Weighted low-rank approximations. In *ICML '03: Proc. of the 21th International Conference on Machine Learning*, 720–727. AAAI Press.

Symeonidis, P.; Nanopoulos, A.; and Manolopoulos, Y. 2008. Tag recommendations based on tensor dimensionality reduction. In *RecSys '08: Proc. of the ACM Conference on Recommender systems*, 43–50. New York, NY, USA: ACM.

Takeuchi, Y., and Sugimoto, M. 2006. Cityvoyager: an outdoor recommendation system based on user location history. In *UIC '06: Proc. of Ubiquitous Intelligence and Computing*, 625–636. Publisher Springer Berlin / Heidelberg.

Wang, J.; de Vries, A. P.; and Reinders, M. J. T. 2006. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR '06: Proc. of the 29th annual ACM SIGIR conference*, 501–508. New York, NY, USA: ACM.

Wyatt, D.; Philipose, M.; and Choudhury, T. 2005. Unsupervised activity recognition using automatically mined common sense. In *AAAI '05: Proc. of the 20th National Conference on Artificial Intelligence*, 21–27. AAAI Press.

Zheng, Y.; Zhang, L.; Xie, X.; and Ma, W.-Y. 2009. Mining interesting locations and travel sequences from gps trajectories. In *WWW '09: Proc. of the 18th Intl. World Wide Web Conference*, 791–800. New York, NY, USA: ACM.

Zheng, V. W.; Zheng, Y.; Xie, X.; and Yang, Q. 2010. Collaborative location and activity recommendations with gps history data. In *WWW '10: Proc. of the 19th International World Wide Web Conference*. New York, NY, USA: ACM.

Zheng, V. W.; Hu, D. H.; and Yang, Q. 2009. Cross-domain activity recognition. In *Ubicomp '09: Proc. of the 11th International Conference on Ubiquitous Computing*, 61–70. New York, NY, USA: ACM.