

Mining GPS Data for Trajectory Recommendation

Peifeng Yin¹, Mao Ye², Wang-Chien Lee¹, and Zhenhui Li³

¹ Department of Computer Science and Engineering, Pennsylvania State University

² Pintrest, San Francisco Bay Area, CA

³ College of Information Science and Technology, Pennsylvania State University
pzy102@cse.psu.edu, m.daniel.ye@gmail.com, wlee@cse.psu.edu, jessieli@ist.psu.edu

Abstract. The wide use of GPS sensors in smart phones encourages people to record their personal trajectories and share them with others in the Internet. A recommendation service is needed to help people process the large quantity of trajectories and select potentially interesting ones. The GPS trace data is a new format of information and few works focus on building user preference profiles on it. In this work we proposed a trajectory recommendation framework and developed three recommendation methods, namely, Activity-Based Recommendation (ABR), GPS-Based Recommendation (GBR) and Hybrid Recommendation. The ABR recommends trajectories purely relying on activity tags. For GBR, we proposed a generative model to construct user profiles based on GPS traces. The Hybrid recommendation combines the ABR and GBR. We finally conducted extensive experiments to evaluate these proposed solutions and it turned out the hybrid solution displays the best performance.

1 Introduction

With the rapid development of mobile devices, wireless networks and Web 2.0 technology, a number of location-based sharing services, e.g., Foursquare¹, Facebook Place², Everytrail³ and GPSXchange⁴, have emerged in recent years. Among them, Everytrail and GPSXchange are particularly unique because they allow users to share their outdoor experiences by uploading GPS *trajectory data* of various outdoor activities, e.g., hiking and biking. By sharing trajectory information, these Web 2.0 sites provide excellent resources for their users to plan or explore outdoor activities of interests.

The rich amount of trajectories available in those web sites brings significant challenges for users to find what they search for. Also, different from conventional items with enrich texts, it is difficult to judge whether the trajectory is interesting or not based on the activity tag or GPS raw data. Therefore, in order to automatically discover interesting trajectories, a trajectory recommendation service is highly desirable.

¹ <http://www.foursquare.com>

² <http://www.facebook.com/places/>

³ <http://www.everytrail.com>

⁴ <http://www.gpsxchange.com/>

Conventional collaborative filtering (CF) techniques do not fit the problem trajectory recommendation. The CF requires people to access the same items to compute user interest similarity. However, in trajectory sharing website, there are no two people who generate exactly the same trajectory and the user similarity can not be calculated by “accessing the same item”.

In this work, we explore the ideas of content-based recommendation techniques [1, 8, 13]. We consider two types of trajectory “content”, activity tags and GPS points. The activity tags, such as hiking or biking, are annotated by the users themselves. The trajectory is represented as a sequence of GPS points with corresponding time stamps.

Recommendation based on tags is named as *activity-based recommendation (ABR)*, which utilizes the tag content (if available) to make trajectory recommendation. Since the tags are manually labeled by the creator, they can be treated as a good feature for a trajectory. Unfortunately, activity tags are not always available for a GPS trajectory. In the Everytrail data we collected, about 12.61% of the trajectories do not have tags. Additionally, ABR may not be able to make recommendation if there are too many candidates with the same tag. For example, in our collected data, 14% of all tagged trajectories, are tagged with “hiking”. One intuitive solution would be using geographical region as a filtering to eliminate infeasible candidates. However, it does not really solve the problem. For example, after constraining the search result into “San Fran”, we still found 96 hiking trajectories in the collected Everytrail dataset. Finally, trajectories with the same tag may have different moving patterns, which the ABR is unable to capture. Let’s consider two hiking fans. The first one likes to take a gentle walk so she can take a lot of photographs but the other one treats hiking as a physical exercise. Naturally, the two trajectories, although both labeled as “hiking”, may contain very different features, which ABR fails to capture.

Considering these weak points of ABR discussed above, we also exploit the sampled points in GPS trajectories for recommendation and call the proposed technique *GPS based recommendation (GBR)*. The raw GPS data contains plentiful movement information (e.g., speed, change of speed, etc.), which captures the user’s outdoor experiences implicitly. For example, techniques for using raw GPS data to infer the transportation modes (e.g., taking bus, taking subway, biking and walking) of trajectories have been studied [22, 21, 17, 18, 7, 6]. However, these techniques are not applicable to our trajectory recommendation service since we aim to capture users’ moving habits and use them to differentiate the trajectories of the same activity type. Take the example of hiking fans mentioned earlier, existing techniques can only classify them as “hiking”. However, what a recommender system needs are more personalized moving habits, e.g., gentle walking or intense trotting. We argue that such information is embedded in GPS data and we aim to mine them out to facilitate trajectory recommendation.

The rest of the paper is organized as follows. Section 2 formally defines the problem, introduces ABR and reviews the related work. Section 3 and 4 respectively detail the GPS feature extraction and the generative model in GBR. Section 5 presents the evaluation of our proposed solutions. Finally, Section 6 concludes the paper.

2 Preliminaries

In this section, we first formally introduce the trajectory recommendation problem and discuss the sub-tasks to tackle the proposed problem. Then we provide a comprehensive literature review on recommendation and trajectory related research work.

2.1 Problem Formulation

A trajectory consists of two parts, i.e., an activity tag (could be absent) and a raw GPS trace. Formally, a trajectory is represented as $T = \langle a, T^G \rangle$, where $a \in \{ \text{"hiking"}, \text{"biking"}, \dots, \text{"null"} \}$ denotes the activity tag and T^G stands for the raw GPS trace.

The GPS trace is obtained via GPS sensor which sampled the moving object's current location together with the sampled time stamp. Thus the original format is a series of triple tuples defined below.

Definition 1 (Raw GPS Trace) *A GPS trace $T^G = \{pt_1, \dots, pt_n\}$ is defined as a series of sample points, $pt_i = \langle x_i, y_i, t_i \rangle$ where x_i, y_i represent the latitude and longitude of the i th point and t_i stands for the time stamp.*

The recommendation problem is to find a subset of candidate trajectories that could be of interest to an active user. More formally, given a collection of trajectories $S = \{T_1, \dots, T_n\}$ and a person u , recommendation needs to find k trajectories $S' = \{T_{r_1}, \dots, T_{r_k}\}$ that u is most interested in. Suppose we have a ranking function $\text{Score}(T, u)$ that can compute the "interest degree" of a trajectory to a user, the recommendation can be formulated as follows.

Definition 2 (Top- k Trajectory Recommendation) *Given a trajectory set $S = \{T_1, \dots, T_n\}$, the recommendation service for user u needs to find a subset of k trajectories $S' = \{T_{r_1}, \dots, T_{r_k}\}$ so that $\forall T_i \in S - S'$, we have*

$$\text{Score}(T_i, u) \leq \min_{T_j \in S'} \text{Score}(T_j, u) \quad (1)$$

The above definition reveals three problems for trajectory recommendation. The first two problems are how to represent the trajectory (Feature Extraction) and the user (User Profile Modeling) in a proper way to facilitate the computation of a ranking score. And the final one is how to design an effective ranking function $\text{Score}(T, u)$ to measure the "interest degree".

2.2 Activity-based Recommendation

The ABR tries capturing a person's activity preferences based on her previously shared trajectories. This preference to different activities is represented as a series of probabilities, whose values are obtained by maximizing the joint probability of observed data.

Let $A = \{a_1, \dots, a_n\}$ denote the collection of all activity tags and $p_i, 1 \leq i \leq n$ denote the probability that the user u is interested in activity a_i . Obviously $\sum_{i=1}^n p_i = 1$. For the user's previously published trajectories, the activity tags

are $X = \{x_1, \dots, x_m\}$ where $x_j \in A, 1 \leq j \leq m$. X is the observed data for the user and the solution is to guess the user’s preference, or exactly the value of p_i based on these experiences. We assume that the instance $x_j \in X$ is independent of each other and the probability of observing X is given in Equation (2).

$$P(X|p_1, \dots, p_m) = \prod_{j=1}^m P(x_j|p_1, \dots, p_m) = \prod_{j=1}^m \sum_{i=1}^n p_i \cdot \mathbf{1}_{x_j=a_i} = \prod_{i=1}^n p_i^{n_i} \quad (2)$$

where n_i represents the number of trajectories that is tagged with a_i in X .

To learn the value of p_i , we need to maximize the Equation (2) under the constraint that the sum of all probabilities is equal to 1, i.e., the objective function as shown in Equation (3).

$$\mathcal{L}(p_1, \dots, p_n) = \log P(X|p_1, \dots, p_n) + \lambda(1 - \sum_{i=1}^n p_i) = \sum_{i=1}^n n_i \log p_i + \lambda(1 - \sum_{i=1}^n p_i) \quad (3)$$

where λ is a Lagrange multiplier.

The objective function is solved by setting each partial differential $\frac{\partial \mathcal{L}}{\partial p_i}$ to 0. For ABR, the ranking function is thus defined as:

$$\text{Score}_{abr}(T, u) = \log \sum_{i=1}^n p_i \cdot \mathbf{1}_{T.activity=p_i} \quad p_i = \frac{n_i}{\sum_{j=1}^n n_j} \quad (4)$$

2.3 Related Work

Due to the wide use of GPS-equipped smart phones, much attention is focused on the use of the trajectory data to improve people’s life, among which *transportation mode detection* is most related to our work.

Zheng et. al. [21, 22] collected 47 people’s GPS data and compared different machine learning techniques to classify transportation modes. The methods however can not be used for recommendation. Trajectory recommendation requires to give a ranking score to each candidate trajectory while classification algorithms, e.g., decision tree, can only output binary values. In [17, 18], Reddy et. al. compared and even ranked different types of trajectory features. One of the most important features in their work is the instant acceleration recorded by accelerometer. This information is usually unavailable for common trajectory information since most of the smart phones are not equipped with accelerometer. In [6, 7], different trajectories of moving objects, including eye-tracking, are collected for transportation mode classification.

Trajectories contain plenty of valuable information. Previous classification works explored different types of features that can well capture the trajectory modes. However they did not pay attention to user’s moving habit that is also contained in trajectory data. Li et. al. [10, 11] tries to mine moving patterns from GPS data of animals. GPS data in our case are records of a person’s trips that happen at different places and few of them overlap with each other. Therefore no periodic patterns can be mined out of such “scattered” data. In [9, 14, 15], Discrete Fourier Transformation is also used to extract features from trajectory

data. However, their goal is for clustering, which is quite straightforward with the extracted data. Our work is to develop generative model based on these features to learn user moving habits for recommendation.

Other works related to recommendation are based on semantic information of trajectory [3, 4, 20]. These works treats trajectory as a sequence of "meaningful places" and use the semantic information of the locations, e.g., restaurant, shopping centers. In our case, trajectories do not have semantic tags. Furthermore, not all trajectories contain meaningful locations. For example, a hiking trajectory is unlikely to pass places such as restaurant, shopping center.

3 GPS Feature Extraction

In this section we focus on extracting features from GPS data. Specifically, we introduce two types of features, i.e., partial-view feature (PVF) and entire-view feature (EVF). The PVF mainly consists of physic values such as speed, velocity, etc., and is easy to understand.

Specifically, given a trajectory's raw GPS data, average velocity, average acceleration and other physical measurements can be easily computed and they represent some characteristics of that trajectory. In this work, the PVF contains the total length of trajectory Len , the total time of the trajectory $Time$ and top- pf_1 maximum velocity $\hat{V}_1, \dots, \hat{V}_{pf_1}$ and top- pf_2 acceleration $\hat{A}_1, \dots, \hat{A}_{pf_2}$.

The EVF tries to capture the global features and is harder to understand semantically. We adopt Discrete Fourier Transform (DFT) to transform the GPS data and a discussion is provided in Section 3.2.

3.1 Entire-view Feature

Before applying DFT on GPS, there are two issues need to be addressed. Firstly, different trajectories may have different lengths, i.e., different number of sampling points. If we take the whole GPS trace as input, DFT will generate features that have different dimensions. This situation makes it difficult to compare two trajectories as they might be in different frequency spectrums. Secondly, there are three kinds of signals that can be obtained from GPS traces, i.e., distance signal, velocity signal and acceleration signal. We need to decide which one should be used as DFT input.

For the first problem, a sliding window of fixed size is used to split the GPS trace into several segments. DFT coefficients of these segments are then refined to form a GPS feature of the same size. This processing method is similar to music compression and classification [12, 16]. As for the second problem, we choose speed signal because i) it suffers less impact of sampling rates than the distance signal and ii) it is more accurate in reflecting the moving status than acceleration signal. Given two trajectories which have the same sampled data points (i.e., latitude, longitude and the number of points) except for the time stamp, the DFT features will be same. However, the moving status for the two trajectories could be quite different if the sampling rates are not the same. The speed series can avoid this weakness. Also, note that the acceleration signal is converted from the velocity signal under the assumption that the object is moving at a constant acceleration between two sampled points. Each manipulation of the GPS data,

e.g., linear interpolation, converting to distance signal etc., may introduce some errors. Therefore the velocity signal is better than the acceleration signal in terms of accuracy.

A final feature of size $ws \times r$ is obtained by keeping the top- r max value for each dimension. The process is both shown in Figure 1. The use of overlapping window is to mitigate the “loss” at the edges of the window⁵. The refined value r is to mitigate the impact of path condition. because extreme cases are usually free of road limitation. For example, if it is found that a user drove 35 miles per hour for some segment while 25 miles per hour for other. The it is more reasonable to believe that the speed 35 mph is more likely to represent the user’s habit instead of 25 mph.

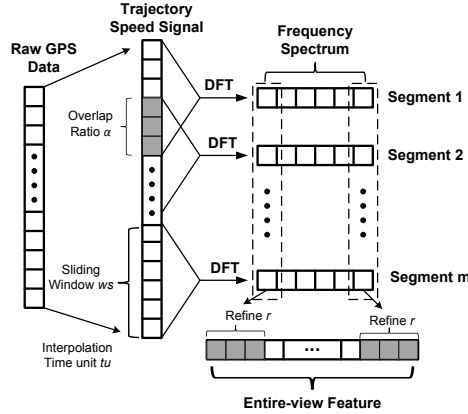
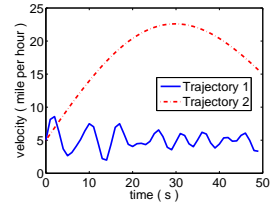
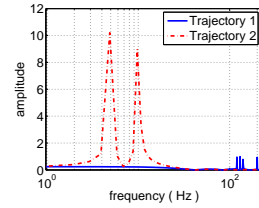


Fig. 1. Illustration of EVF extraction. Four parameters, interpolation time unit tu , window size ws , overlap ratio α and refine value r , are involved.



(a) Original Speed Signal



(b) Frequency Spectrum

Fig. 2. Illustration of feature meaning.

3.2 Discussion of the Entire-view Feature

The basic waves have two components, i.e., frequency and amplitude. The frequency indicates the intensity of speed change. That is, lower frequency suggests that the moving is smoother while higher one means the moving object frequently changes its speed. The amplitude of the wave reflects the strength of a signal, or exactly, the absolute value of the speed. We use the example shown in Figure 2 to illustrate this point.

As shown in Figure 2(a), the speed signal of two trajectories are drawn. It is easily seen that the trajectories are quite different. Firstly, trajectory 1 has a smaller speed than trajectory 2. Also, the signal of trajectory 2 is smooth while the speed of trajectory 1 suffers frequent change. Figure 2(b) displays the result of DFT. Trajectory 2’s amplitudes are bigger than that of Trajectory 1, corresponding to its original bigger speed. Furthermore, the frequency of trajectory

⁵ http://en.wikipedia.org/wiki/Window_function#Overlapping_windows

1 lies mainly in higher spectrum while that of trajectory 2 in lower part. This is consistent with the change of their original speed signal where low frequency reflects smooth signal and high frequency corresponds to intense change.

4 User Profile Modeling and Trajectory Ranking

In this section we formally define the proposed generative model and introduce the designed ranking functions.

4.1 Generative Model

Formally, suppose a person u has k latent habits $\mathbf{z} = \{z_1, \dots, z_k\}$ which satisfies some distribution. For each habit z_i , there are n groups of parameters $\langle \mu_1^i, \sigma_1^i \rangle, \dots, \langle \mu_n^i, \sigma_n^i \rangle$ corresponding to each dimension of trajectory feature. For each dimension, we assume that the value follows a Gaussian distribution $\mathcal{N}(\mu_j^i, \sigma_j^i)$ with mean μ_j^i and standard variance σ_j^i . The generative model works as follows. The person u first chooses a latent habit z_k with probability π_k . Then this habit generates the amplitude for each frequency in the feature space, which is determined by a Gaussian distribution. The generated feature finally constructs a trajectory.

Given a set of trajectories $\mathbf{S} = \{\mathbf{T}_1, \dots, \mathbf{T}_N\}$, where $\mathbf{T}_i = \langle f_1^i, \dots, f_n^i \rangle$ is a trajectory with n features. If all trajectories in S is known to be generated by the person u , then the parameters $\theta = \{\pi, \mu, \sigma\}$ can be learned by maximizing the following log-likelihood function:

$$L(\theta) = \log P(\mathbf{S}|\theta) + \lambda \left(\sum_{i=1}^k \pi_i - 1 \right) = \sum_{i=1}^N \sum_{j=1}^k \tau_{ij} \left(\log \pi_j + \sum_{m=1}^n \log \mathcal{G}(\cdot) \right) + \lambda \left(\sum_{i=1}^k \pi_i - 1 \right) \quad (5)$$

where $\mathcal{G}(f_m^i, \mu_m^j, \sigma_m^j) = \frac{1}{\sqrt{2\pi}\sigma_m^j} e^{-\frac{(f_m^i - \mu_m^j)^2}{2(\sigma_m^j)^2}}$ is the probability of value f_m^i for Gaussian distribution $\mathcal{N}(\mu_m^j, \sigma_m^j)$, λ is the Lagrange multiplier and τ_{ij} is an indicator function whose value is 1 if and only if the trajectory T_i is generated by habit z_j .

We use *EM algorithm* [5] to solve this problem. In the following, let $\theta(t)$ denote the values of parameters at t^{th} iteration.

E-Step

$$E[\tau_{ij}] = \frac{P(T_i|z_j)P(z_j)}{\sum_{m=1}^k P(T_i|z_m)P(z_m)} = \frac{\prod_{d=1}^n \mathcal{G}(f_d^i, \mu_d^j(t-1), \sigma_d^j(t-1))\pi_j^{(t-1)}}{\sum_{m=1}^k \prod_{d=1}^n \mathcal{G}(f_d^i, \mu_d^m(t-1), \sigma_d^m(t-1))\pi_m^{(t-1)}} \quad (6)$$

M-Step

$$\pi_j^{(t)} = \frac{\sum_{i=1}^N E[\tau_{ij}]}{\sum_{i=1}^N \sum_{m=1}^k E[\tau_{im}]} \quad \mu_m^j(t) = \frac{\sum_{i=1}^N E[\tau_{ij}]f_m^i}{\sum_{i=1}^N E[\tau_{ij}]}$$

$$\sigma_m^j(t) = \sqrt{\frac{\sum_{i=1}^N E[\tau_{ij}](\mu_m^j(t) - f_m^i)^2}{\sum_{i=1}^N E[\tau_{ij}]}}$$

4.2 Ranking Function

We provide two ranking functions, of which one is developed from the generative model and the other is hybrid of ABR and GBR.

The generative model discussed above captures the user’s moving habit and can then be used for recommendation. Each user has a profile expressing her latent habits $\langle z_1, \dots, z_k \rangle$ with probability $\langle \pi_1, \dots, \pi_k \rangle$. For each habit z_i , there is a n -dimension Gaussian distribution $\{\langle \mu_1, \sigma_1 \rangle, \dots, \langle \mu_n, \sigma_n \rangle\}$. The ranking of a trajectory to a given user’s profile is to compute the probability that the trajectory is generated by that user. More specifically, for a trajectory $T = \langle f_1, \dots, f_n \rangle$, its ranking score is computed below.

$$\text{Score}_{gm}(T, u) = \log P(T|u) = \log \sum_{i=1}^k P(T|z_i)P(z_i) = \log \sum_{i=1}^k \pi_i \prod_{j=1}^n \mathcal{G}(f_j, \mu_j, \sigma_j) \quad (7)$$

The probability $P(u)$ is omitted since it is a constant to all candidate trajectories.

The ABR introduced in Section 2.2 is aware of a person’s historical activities but can not distinguish two trajectories with the same activity. On the other hand, the GBR described in Section 4 ranks trajectories purely based on its GPS data and the user profile. In other words, it gives different ranking scores to trajectories disregarding their activity tags. Therefore it indicates a potential improvement when the two methods are combined. Equation (8) shows the hybrid ranking function that merges the two recommendation methods.

$$\text{Score}_{hybrid}(T, u) = (1 - \lambda)\text{Score}_{gm}(T, u) + \lambda\text{Score}_{abr}(T, u) \quad (8)$$

where $0 \leq \lambda \leq 1$ is a balance parameter adjusting the weight of the two separate ranking scores. Note that when $\lambda = 0$ the hybrid ranking becomes pure generative model and $\lambda = 1$ reduces the method to pure ABR.

5 Evaluation

All the experiments are based on a real data set collected from Everytrail. Everytrail is a trajectory sharing website encouraging people to publish their trip trajectories recorded by the smart phones. Each trajectory may also be associated with an activity tag, e.g., hiking, road biking, driving and so on. We crawled the website from June 05, 2010 to August 07, 2010 and obtains 8,444 users and 63,760 trajectories in total.

5.1 Data Preparation

Because the trajectory data is uploaded by different people and there is no strict examination, the raw data contains much noise for mining. We take the

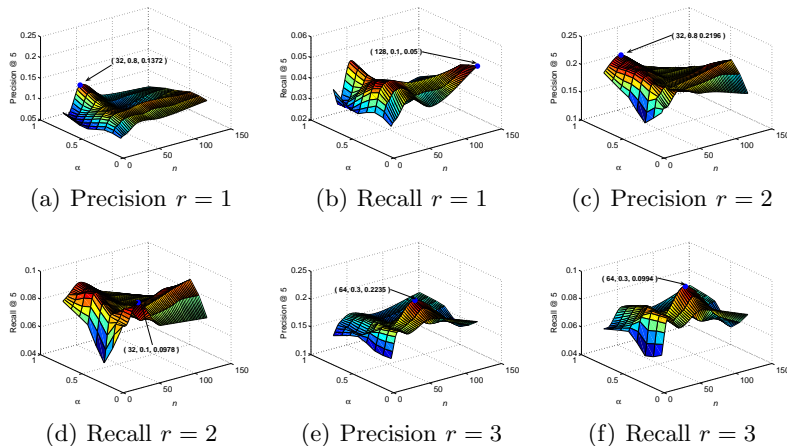


Fig. 3. Parameter Tuning for the EVF

following steps to clean the data. Firstly, all untagged trajectories are removed to guarantee a fair comparison of ABR and GBR. Also, we removed trajectories which have “illegal” GPS traces, defined as those i) whose sampled time stamp are not monotone increasing ii) whose total time length is less than 300 seconds. Finally, users whose shared trajectories are less than 20 are removed since small sample may hurt the model accuracy and thus the recommendation performance. After this preprocessing, there are 252 users and 9,120 trajectories remaining.

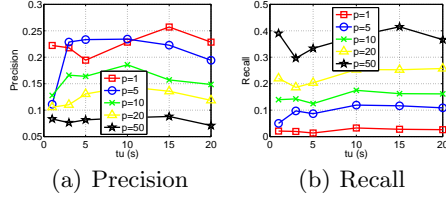
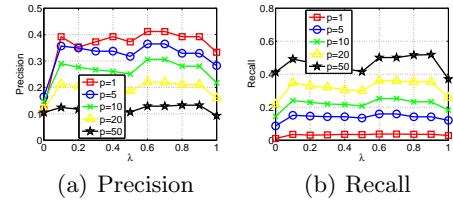
For each user in the preprocessed data, we randomly masked 20% of her trajectories. These masked data serve as the test data to evaluate our proposed recommendation methods and the remaining part is used as training data to build the user’s profile. Furthermore precision and recall are metrics used to evaluate the recommendation performance.

5.2 Parameter Tuning for Entire-view Feature

Figure 3 shows the tuning process for the remaining parameters, where ws ranges from 8 to 128, α from 0.1 to 0.8 and r from 1 to 3. Also, the optimal configuration for different r is highlighted in the figure. As is shown, best performance is achieved when the feature dimension is 64, the overlapping window percentage is 30% and the refined window size is 3.

Figure 4 shows experiments on time unit tu which varies from 3 seconds to 20 seconds. It can be seen that the time unit should be neither too small nor too large. For a small time unit, many pseudo points have to be added via linear interpolation and it may introduce too many errors. On the other hand, a big time unit may discard some true sample points, which may miss important information. Judging from the result, tu is properly set to 5 seconds.

Finally Figure 5 shows the tuning of λ for hybrid recommendation. Note that when $\lambda = 0$ the method is reduced to the ABR and when $\lambda = 1$ it is equivalent to the GBR. Based on the result we set λ to 0.9 for the rest experiments.

Fig. 4. Parameter Tuning for tu Fig. 5. Parameter Tuning for λ

5.3 Comparison of Different Features

After determining the optimal configuration of EVF, we compare its recommendation performance with two other features, namely, partial-view feature (PVF) and combined feature. The PVF, as introduced in Section 3, consists of top- pf_1 maximum velocity, top- pf_2 acceleration, the length and time of the trajectory. In the empirical study, the best performance of PVF is obtained when both pf_1 and pf_2 are set to 3. Combined feature is the combination of the two types of features. As is seen in Figure 6, the combined feature outperforms the other two in terms of both precision and recall, especially for top-1 recommendation. This scenario proves our earlier argument that these two types of features represent different aspects of a trajectory. The PVF aims to capture the locally extreme characteristics while the EVF places more emphasis on the global picture of a trajectory. They are complimentary to each other and can achieve the best performance when combined together.

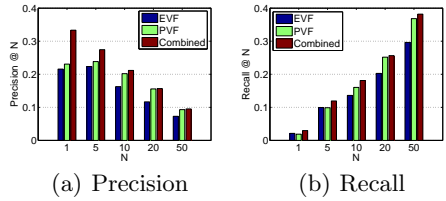


Fig. 6. Feature Comparison

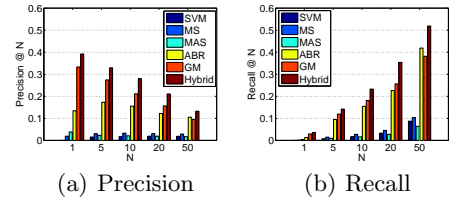


Fig. 7. Method Comparison

5.4 Comparison of Different Recommendation Approaches

In this set of experiment, we evaluate the performance of the proposed recommendation methods, i.e., ABR in Section 2.2, GBR in Section 4 and Hybrid Recommendation where the λ is set to 0.9. We also include several baseline methods for comparison. The details of them are listed below.

- Support Vector Machine (SVM)⁶. The SVM treats each user as a class and all the trajectories' combined features as training data. Then each candidate trajectory will be assigned a series of probabilities indicating how likely it belongs to each class (user). These values are thus treated as ranking scores.

⁶ In this experiment we used LIBSVM [2]

- Maximum Similarity (MS). This method computes the cosine similarity of the candidate trajectory’s combined feature to each of those uploaded by the user and uses the maximum one as its score.
- Maximum Average Similarity (MAS). This method computes a centroid for a collection of feature vectors of the user’s trajectories. Ranking score is then determined by computing the cosine similarity between the candidate trajectory to this centroid.

The MS and MAS can also be treated as an variant of item-based recommendation [19]. Note that the above baselines share the same features (i.e., GPS combined features) with our proposed GBR but adopt different approaches for user-profile modeling and trajectory ranking. To distinguish the difference, we use generative model (GM) to denote our proposed GBR.

The experiment results are shown in Figure 7. It is easily seen that the generative model outperforms other methods that use GPS feature. These recommendation methods, SVM, MS, MAS, are solely based on user’s previous trajectories and thus may be too biased to history data. The generative model, instead of limiting to the data, tries to learn the users’ hidden moving pattern and thus can achieve higher precision. ABR is the only method that relies on the tag of the trajectory. In Figure 7, its performance, in terms of both precision and recall, is worse than GM except for $N = 50$. As mentioned in Section 1, this method can not distinguish trajectories of the same activity. This explains its low precision and recall when N is small. The performance improvement for bigger N , however, suggests a user’s concentration on the number of different trajectory’s activities. Finally, the hybrid recommendation, which combines the ranking functions of ABR and GM, shows the best performance. The ABR is aware of a person’s historical activities but can not distinguish two trajectories with the same activity. On the other hand, the proposed generative model ranks trajectories without any knowledge of their activities. The hybrid recommendation method integrates both of their advantages, taking care of textual information and the GPS traces. It can thus not only narrow down the range of candidate trajectories but also distinguish those with the same activity tag. This explains why the hybrid approach displays the best performance.

6 Conclusion

In this paper we studied the problem of trajectory recommendation. Each trajectory usually consists a GPS trace and may contain an activity tag. A recommendation service is supposed to find potentially interesting trajectories and push them to particular people. We proposed a recommendation framework and divide the task into three subproblems, i.e., feature extraction, user profile modeling and trajectory ranking. Under this framework, we developed three recommendation methods, namely, Activity-Based Recommendation (ABR), GPS-Based Recommendation (GBR) and Hybrid recommendation. We conduct extensive experiments to evaluate our solutions. For GPS feature, it is shown that the combination of partial-view feature and entire-view feature achieves best performance. As for the recommendation approach, the hybrid one that combines ABR and GBR obtains highest precision and recall. In future work, we will focus on ways of integrating travelogues, if available, into recommendation methods.

References

1. M. Balabanovic and Y. Shoham. Content-based collaborative recommendation. *CACM*, 40(3):66–72, 1997.
2. C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
3. Z. Chen, H. T. Shen, and X. Zhou. Discovering popular routes from trajectories. In *ICDE*, 2011.
4. Z. Chen, H. T. Shen, X. Zhou, Y. Zheng, and X. Xie. Searching trajectories by locations: an efficiency study. In *SIGMOD*, pages 255–266, 2010.
5. A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.
6. S. Dodge, R. Weibel, and E. Forootan. Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects. *Computers, Environment and Urban Systems*, 33(6):419 – 434, 2009.
7. S. Dodge, R. Weibel, and P. Laube. Exploring movement-similarity analysis of moving objects. *SIGSPATIAL Special*, 1:11–16, November 2009.
8. A. M. Ferman, J. H. Errico, P. van Beek, and M. I. Sezan. Content-based filtering and personalization using structured metadata. In *JCDL*, page 393, 2002.
9. X. Li, W. Hu, and W. Hu. A coarse-to-fine strategy for vehicle motion trajectory clustering. In *Pattern Recognition*, 2006.
10. Z. Li, B. Ding, J. Han, R. Kays, and P. Nye. Mining periodic behaviors for moving objects. In *KDD*, pages 1099–1108, 2010.
11. Z. Li, J. Han, M. Ji, L.-A. Tang, Y. Yu, B. Ding, J.-G. Lee, and R. Kays. Movemine: Mining moving object data for discovery of animal movement patterns. *TIST*, 2010.
12. B. Logan. Mel frequency cepstral coefficients for music modeling. In *ISMIR*, 2000.
13. P. Melville, R. J. Mooney, and R. Nagarajan. Content-Boosted Collaborative Filtering for Improved Recommendations. In *AAAI/IAAI*, pages 187–192, 2002.
14. A. Naftel and S. Khalid. Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space. *Multimedia Systems*, 12(3):227–238, 2006.
15. A. Naftel and S. Khalid. Motion trajectory learning in the dft-coefficient feature space. In *IEEE Conf. Comput. Vision Syst.*, 2006.
16. S. Pfeiffer and T. Vincent. Formalisation of mpeg-1 compressed-domain audio features. Technical report, CSIRO mathematical and information, sciences, Australia, 2001.
17. S. Reddy, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Determining transportation mode on mobile phones. In *ISWC*, 2008.
18. S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. *ACM Trans. Sen. Netw.*, 6:13:1–13:27, March 2010.
19. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.
20. V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. In *WWW*, pages 1029–1038, 2010.
21. Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma. Understanding transportation modes based on gps data for web applications. *ACM Trans. Web*, 4:1:1–1:36, January 2010.
22. Y. Zheng, L. Liu, L. Wang, and X. Xie. Learning transportation mode from raw gps data for geographic applications on the web. In *WWW*, pages 247–256, 2008.