

Diversified Trajectory Pattern Ranking in Geo-Tagged Social Media*

Zhijun Yin¹ Liangliang Cao¹ Jiawei Han¹ Jiebo Luo² Thomas Huang¹

University of Illinois at Urbana-Champaign¹
Kodak Research Laboratories²

zyin3@illinois.edu, cao4@ifp.uiuc.edu, hanj@cs.uiuc.edu,
jiebo.luo@kodak.com, huang@ifp.uiuc.edu

Abstract

Social media such as those residing in the popular photo sharing websites is attracting increasing attention in recent years. As a type of user-generated data, wisdom of the crowd is embedded inside such social media. In particular, millions of users upload to Flickr their photos, many associated with temporal and geographical information. In this paper, we investigate how to rank the trajectory patterns mined from the uploaded photos with geotags and timestamps. The main objective is to reveal the collective wisdom recorded in the seemingly isolated photos and the individual travel sequences reflected by the geo-tagged photos. Instead of focusing on mining frequent trajectory patterns from geo-tagged social media, we put more effort into ranking the mined trajectory patterns and diversifying the ranking results. Through leveraging the relationships among users, locations and trajectories, we rank the trajectory patterns. We then use an exemplar-based algorithm to diversify the results in order to discover the representative trajectory patterns. We have evaluated the proposed framework on 12 different cities using a Flickr dataset and demonstrated its effectiveness.

1 Introduction

Social media is becoming increasingly popular with the development of Web 2.0. Social media websites such

as Flickr, Facebook, and YouTube host overwhelming amounts of photos and videos. In such a media sharing community, image or video files are contributed, tagged, and commented by users all over the world. Extra information can be incorporated within social media, such as geographical information captured by GPS devices. Studying such social media attracts both academic and industrial interests, and it has been a hot research area in recent years [31, 11, 30, 29, 3, 9, 27].

The goal of this paper is to explore the common wisdom in photo sharing community. We study millions of personal photos in Flickr, which are associated with user tags and geographical information. The geographical information is captured by low-cost GPS chips in cell phones and cameras, and often saved in the header of image files. Going beyond the recent work on using Flickr photos to match tourist interests in terms of locations [30, 29, 3], this paper focuses on trajectory patterns. We would like to discover trajectory patterns interesting to two kinds of users. First, some users are interested in the most important trajectory patterns. When they visit a new city, they would like to follow those trajectories that concentrate on popular locations that lots of people are interested in. Second, some users are interested in exploring a new place in diverse ways. They are not only interested in the most important trajectories, but also eager to explore other routes to cover the entire area. Instead of focusing on how to mine frequent trajectory patterns, in this paper we put more effort into ranking the mined trajectory patterns and diversifying the ranking results. Trajectory pattern ranking helps the first kind of users who are interested in top important trajectories, while diversification helps the second kind of users that are willing to explore the diverse routes. There are some studies on trip planning using Flickr. In [9, 8], Choudhury et al. formulated trip planning as directed orienteering problem. In [27], Lu et al. used dynamic programming for

*Research was sponsored in part by the U.S. National Science Foundation under grants CCF-0905014 and CNS-0931975, by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS-CTA), and Air Force Office of Scientific Research MURI award FA9550-08-1-0265. Cao and Huang were sponsored in part by a Beckman Institute (Illinois) Seed Grant and in part by an NSF Grant IIS 1049332 EAGER. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

trip planning. In [23], Kurashima et al. recommended tourist routes by combining topic models and Markov model. However, diversified trajectory pattern ranking has not been investigated in geo-tagged social media before. In this paper, after aggregating the trajectories by pattern mining, we investigate the problem of ranking trajectory patterns and diversifying the ranking results.

Trajectory mining has been investigated in many datasets including animal movement [26], hurricane tracking [26], urban traffic [17, 16], human travel history [36, 35], etc.. Giannotti et al. developed a spatial-temporal pattern mining paradigm that discovers trajectory patterns [17]. In this paper, we mine trajectory patterns from geo-tagged social media, where trajectory patterns are represented by a sequence of locations according to temporal order. Usually the pattern mining result is a set of mined patterns with their frequencies. However, this kind of representation has several disadvantages. First, there are too many trajectory patterns in the result. It is difficult for users to go through all the patterns in the list to discover the interesting ones. As a result, the interesting trajectory patterns are buried in the massive result set. Second, if we only return the top frequent trajectory patterns as results, the results may not be interesting to all users. One reason is that top frequent trajectory patterns are usually short and not informative. Third, many trajectory patterns are similar, so redundant information exists in the results. Many frequent trajectory patterns share common sub-patterns, so it is not interesting to output all of them. To overcome the above problems, we propose an algorithm to rank trajectory patterns by considering the relationships among users, locations and trajectories, and introduce an exemplar-based algorithm to diversify trajectory pattern ranking results.

This paper is organized as follows. In Section 2, we formulate our problem of diversified trajectory pattern ranking in geo-tagged social media. In Section 3, we introduce how to apply pattern mining technique in geo-tagged social media. We show how to rank the trajectory patterns in Section 4 and discuss diversification of trajectory pattern ranking results in Section 5. We demonstrate the effectiveness of our methods in Section 6, summarize related work in Section 7, and conclude the paper in Section 8.

2 Problem Definition

In geo-tagged social media sites such as Flickr, a large collection of photos is uploaded by users. Each photo is taken by a user at specific location and time. The location is annotated by GPS coordinate (x, y) , where x refers to longitude and y refers to latitude. Here are a few definitions used in this paper.

DEFINITION 2.1. *Location is a popular region that users visit.*

DEFINITION 2.2. *Trajectory is a sequence of locations visited by a user according to temporal order during the same day¹.*

DEFINITION 2.3. *Trajectory pattern is a sequence of locations whose frequency is no smaller than minimum support. The sequence frequency is defined as the number of users visiting the locations according to the order in the sequence. In this paper, we only discuss the trajectory patterns without duration constraints.*

The problem of diversified trajectory pattern ranking in geo-tagged social media is formulated as given a collection of geo-tagged photos along with users, locations and timestamps, how to rank the mined trajectory patterns with diversification into consideration.

Our framework consists of three main components: (1) extracting trajectory patterns from the photo collection, (2) ranking the trajectory patterns by estimating their importance according to user, location and trajectory pattern relations, and (3) diversifying the ranking result to identify the representative trajectory patterns from all the candidates. We will explain these components one by one. Though geo-tagged social media is taken as an example, our technique can be applied to other trajectory pattern ranking scenarios.

3 Trajectory Pattern Mining Preliminary

In this section, we introduce how to mine frequent trajectory patterns in geo-tagged photo collections such as those in Flickr. Since the GPS coordinates of photos are at a very fine granularity, we need to detect locations before extracting trajectory patterns. This process is similar to discovering ROI (region of interest) in [17]. With the detected locations, we can generate the trajectories for each user according to his visiting order of the locations during the same day. Then we can mine the frequent trajectory patterns using sequential pattern mining. If a trajectory pattern repeats frequently, we consider it as a *frequent trajectory pattern*.

3.1 Location Detection We cluster the photo GPS coordinates to detect the locations. The location clustering process should satisfy the following criteria: (1) The close-by points should be grouped together, (2) clustering should accommodate arbitrary shapes, and

¹We can extend the trajectory to span multiple days. In this paper, we only discuss the scenario within the same day without loss of generality.

(3) the clustering parameters should be easy to set according to the application scenario. Considering all the aspects, we use the mean-shift algorithm [7, 10] to extract locations. Mean-shift is an iterative procedure that shifts each data point to the average of data points in its neighborhood.

$$m(x) = \frac{\sum_{s \in S} K(s-x)s}{K(s-x)s}$$

where S is the set of all the data points. The difference $m(x) - x$ is the mean shift. $x \rightarrow m(x)$ is performed for all $s \in S$ simultaneously to move the data point to its sample mean and this process repeats until convergence. In our case, the location of a photo is considered as a data point. If one user takes multiple photos at the same place, it is only counted as one point. Here we use a flat kernel function and set the bandwidth λ as 0.001. 0.001 in GPS coordinate is approximately 100m. We assume that a reasonable extent for typical locations is around 100 meters in diameter, so we use 0.001 as the bandwidth for mean-shift clustering. We use a flat kernel as follows.

$$K(x) = \begin{cases} 1 & \text{if } \|x\| \leq \lambda \\ 0 & \text{if } \|x\| > \lambda. \end{cases}$$

3.2 Location Description After getting the locations in Section 3.1, we need to generate the descriptions for the locations. For each photo in Flickr, besides the GPS location, we also have associated tags that are contributed by users. Here we use a generative mixture model to extract useful tags for each location. The details can be found in Appendix A. As an example, we crawled 27974 photos in London area from Flickr and clustered 883 locations. The top locations associated with their tag descriptions are listed in Table 1.

3.3 Sequential Pattern Mining After the location clustering step, we generate the trajectories according to the visiting order of the locations. We use the PrefixSpan algorithm [28] to extract the frequent sequential patterns and treat them as trajectory patterns. Given a set of sequences, sequential pattern mining algorithm will find all the sequential patterns whose frequencies are no smaller than the minimum support. The frequency of a pattern is defined as the number of sequences subsuming the pattern. In this paper, we set the minimum support threshold as 2 to collect as many trajectory patterns as possible for ranking. Here we show a sequential pattern mining example.

Example. Given 5 sequences in Table 2 and a minimum support of 2, we find the following three patterns: *londoneye* \rightarrow *bigben* has a frequency of 4, since it is

Table 1: Top locations in London and their descriptions. The number in the parentheses is the number of users visiting the place.

londoneye(528), trafalgarsquare(456), britishmuseum(230), bigben(205), waterloobridge(198), towerbridge(185), piccadillycircus(182), royalfestivalhall(175), coventgarden(169), centrepoint(169), parlamentsquare(150), cityhall(141), oxfordcircus(138), lloyds(121), buckinghampalace(107), naturalhistorymuseum(97), canarywharf(94), bricklane(91), toweroflondon(91), brighton(90), embankment(88), soho(80), stpancras(77), stpaulscathedral(77), leicestersquare(76), gherkin(75), stjamespark(68), barbican(67), victoriaandalbertmuseum(64)
--

Table 2: Sequential pattern mining example.

ID	Travel sequence
1	<i>londoneye</i> \rightarrow <i>bigben</i> \rightarrow <i>trafalgarsquare</i>
2	<i>londoneye</i> \rightarrow <i>bigben</i> \rightarrow <i>downingstreet</i> \rightarrow <i>trafalgarsquare</i>
3	<i>londoneye</i> \rightarrow <i>bigben</i> \rightarrow <i>westminster</i>
4	<i>londoneye</i> \rightarrow <i>tatemodern</i> \rightarrow <i>towerbridge</i>
5	<i>londoneye</i> \rightarrow <i>bigben</i> \rightarrow <i>tatemodern</i>

contained in sequences 1, 2, 3 and 5. *londoneye* \rightarrow *bigben* \rightarrow *trafalgarsquare* has a frequency of 2, since it is contained in sequences 1 and 2. *londoneye* \rightarrow *tatemodern* has a frequency of 2, since it is contained in both sequences 4 and 5.

After clustering 883 locations in London in the previous stage, we generate 4712 trajectories. We mine 1202 trajectory patterns from these trajectories and list the top frequent trajectory patterns in London in Table 3, where the location descriptions are generated in Section 3.2. From Table 3, we find that the most frequent trajectory patterns contain important locations but reveal limited information. These frequent patterns are short and not informative, so we need a better ranking mechanism to organize the mined patterns.

4 Trajectory Pattern Ranking

In this section, we discuss how to rank the mined trajectory pattern without considering diversification. First, we discuss the needs of trajectory pattern ranking and introduce the general idea about our ranking strategy. Second, we describe our ranking algorithm in detail. Third, we analyze the complexity of the algorithm and give the convergence proof.

Table 3: Top frequent trajectory patterns in London.

Trajectory pattern	Frequency
londoneye → bigben	21
bigben → londoneye	19
londoneye → tatemodern	18
londoneye → royalfestivalhall	15
londoneye → trafalgarsquare	14
londoneye → waterlooobridge	12
towerbridge → cityhall	12
royalfestivalhall → londoneye	11
tatemodern → londoneye	11
bigben → parlamentsquare	10

4.1 General Idea In the previous stage, we extract all the frequent sequential patterns as trajectory patterns. In this way, the common movement behaviors are extracted from the data collection. However, there are too many trajectory patterns and it is difficult for users to browse all the candidates. Therefore, a ranking mechanism is needed for these trajectory patterns. We can simply rank all these trajectory patterns by their frequencies as in Table 3, where frequency refers to the number of users visiting the sequence. As one can see in Table 3, all the top ten trajectory patterns ranked by frequency are of length 2. Although the top frequent trajectory patterns cover the important locations such as *londoneye*, *bigben* and *tatemodern*, they are not informative for the reason that people are more interested in the sequential order of locations. In order to derive better importance measure of a trajectory pattern, we need to consider more aspects about geo-tagged social media.

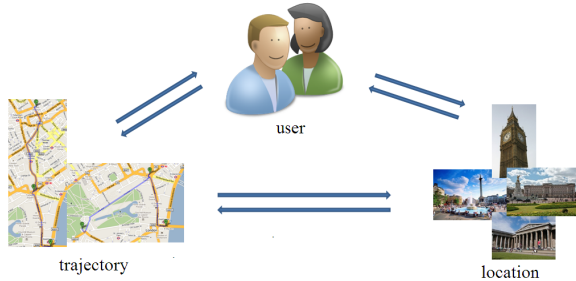


Figure 1: Relationship among user, location and trajectory in geo-tagged social media

In geo-tagged social media, relationships among users, locations and trajectories are embedded in the dataset as illustrated in Figure 1. Specifically, a trajectory is a sequence of locations visited by users, and its importance should be influenced both by users and locations. Here we propose a ranking algorithm to explore such relations. The assumptions that we make here are listed as follows.

1. A trajectory pattern is important if many important users take it and it contains important locations,
2. a user is important if the user takes photos at important locations and visits the important trajectory patterns, and
3. a location is important if it occurs in one or more important trajectory patterns and many important users take photos at the location.

4.2 Ranking Algorithm We denote the importance scores of users, locations and trajectory patterns as P_U , P_L and P_T . P_U , P_L and P_T are all vectors. Each element in the vector is the importance score of the corresponding unit. The relationship between P_U , P_L and P_T are as follows according to the above assumptions.

$$P_L = M_{LT} \cdot P_T \quad P_U = M_{UL} \cdot P_L$$

$$P_T = M_{TU} \cdot P_U \quad P_U = M_{TU}^T \cdot P_T$$

$$P_L = M_{UL}^T \cdot P_U \quad P_T = M_{LT}^T \cdot P_L$$

M_{TU} is the trajectory-user matrix, and its entry indicates the ownership of a trajectory for a user. M_{UL} is the user-location matrix indicating whether a user takes a photo at a location. M_{LT} is the location-trajectory matrix indicating whether a location is contained in a trajectory or not.

We summarize the trajectory pattern ranking algorithm in Algorithm 1. The importance scores of users, locations and trajectories mutually enhance each other according to the assumptions until convergence.

Algorithm 1 Trajectory pattern ranking

Input: M_{TU} , M_{UL} , M_{LT}

Output: A ranked list of trajectory patterns

1. Initialize $P_T^{(0)}$
 2. Iterate

$$P_L = M_{LT} \cdot P_T^{(t)} \quad P_U = M_{UL} \cdot P_L$$

$$P_T = M_{TU} \cdot P_U \quad P_U = M_{TU}^T \cdot P_T$$

$$P_L = M_{UL}^T \cdot P_U \quad P_T^{(t+1)} = M_{LT}^T \cdot P_L$$

$$P_T^{(t+1)} = P_T^{(t+1)} / \|P_T^{(t+1)}\|_1$$
 until convergence.
 3. Output the ranked list of trajectory patterns in the decreasing order of P_T^* , i.e., the converged P_T .
-

To illustrate the updating procedure, we use a toy example with 3 users and 4 trajectory patterns to characterize how P_L , P_U , and P_T evolve in each iteration in Table 4. Note that the importance scores are normalized for better understanding.

Table 4: A toy example for trajectory pattern ranking. Trajectory patterns with user and location information.

Trajectory index	User index	Location sequence
T1	U1	L1, L2, L3
T2	U1	L3, L4
T3	U2	L1, L3
T4	U3	L3, L4

Importance scores updating process.

Iteration		1	2	3	4
Trajectories	T1	0.2500	0.3077	0.3090	0.3088
	T2	0.2500	0.2308	0.2300	0.2299
	T3	0.2500	0.2308	0.2310	0.2313
	T4	0.2500	0.2308	0.2300	0.2299
Users	U1	0.3333	0.6061	0.6060	0.6062
	U2	0.3333	0.2020	0.2020	0.2022
	U3	0.3333	0.1919	0.1920	0.1917
Locations	L1	0.2500	0.2516	0.2520	0.2516
	L2	0.2500	0.1887	0.1890	0.1887
	L3	0.2500	0.3113	0.3110	0.3113
	L4	0.2500	0.2484	0.2480	0.2484

We list the trajectory pattern ranking result in London in Table 5. Although the trajectory pattern frequencies are low, they are visited by important users and cover important locations. Compared with the top frequent trajectory patterns in Table 3, the trajectory patterns ranked by our model are more informative.

In Algorithm 1, not only can we estimate the importance of trajectories, but we can also get the location importance according to P_L . The top locations according to P_L from London are listed in Table 6. The user count of the locations stands for the number of the users visiting the location. We find that the location importance is not determined by user count alone. For example, *horseguards* is important for trajectory pattern ranking even though it only has a user count of 25. The reason is that *horseguards* is on the popular trajectory from *bigben* to *trafalgarsquare*.

4.3 Convergence and Complexity In Algorithm 1, we consider the importance vector of trajectory patterns P_T^* as the eigen vector for $M^T \cdot M$ for the largest eigen value, where $M = M_{TU} \cdot M_{UL} \cdot M_{LT}$. We find that Algorithm 1 is a normalized power iteration method to detect the eigen vector of $M^T \cdot M$ for the largest eigen value if $P_T^{(0)}$ is not orthogonal to it. Therefore, the convergence of the algorithm is guaranteed according to convergence property of normalized power iteration method [21].

In Algorithm 1, each iteration requires the multiplication of $M^T \cdot M$ and P_L , so it need $O(|E|)$, where $|E|$ is the number of nonzero elements in $M^T \cdot M$. In total, the time complexity of the algorithm is $O(k|E|)$, where k is the iteration number.

5 Trajectory Pattern Diversification

In this section, we discuss the diversification of trajectory pattern ranking results. First, we discuss why we need to diversify the ranking results. Second, we propose an exemplar-based algorithm to discover representative trajectory patterns. Third, we analyze the algorithm complexity and give some discussion.

5.1 General Idea Although our algorithm in Section 4 works well in ranking trajectory patterns, the results may not satisfy the needs of all the people. Take the top ranked trajectory patterns in London in Table 5 as an example, the results are useful for people who are new to London, because it clearly illustrates the popular routes together with important sites such as *londoneye*, *bigben*, and *tatemodern*. However, for others who want to explore more areas in London, the result in Table 5 is highly concentrated in only a few regions. Besides, some trajectories are very similar. For example, in Table 5, Trajectory 1 (*londoneye* → *bigben* → *downingstreet* → *horseguards* → *trafalgarsquare*) and Trajectory 5 (*westminster* → *bigben* → *downingstreet* → *horseguards* → *trafalgarsquare*) are almost the same except the starting points. If we recommend both Trajectory 1 and Trajectory 5 in top ten ranked results, it may not be so useful. To overcome the problem, we need to diversify the trajectory pattern ranking results in order to identify the representative trajectory patterns. To generate diversified result, we would like to have three properties on the diversification algorithm. First, similar trajectory patterns need to be aggregated together. Second, good exemplars of trajectory patterns need to be selected. Third, those trajectory patterns ranked highly in our ranking algorithm should get higher priority to be exemplars.

5.2 Diversification Algorithm Since we value diversified trajectories, we need a measure to model how different or how similar two trajectories are. The measure should penalize the trajectories that have common sub-patterns. Therefore, we defined the similarity between two trajectories i and j based on the longest common subsequence(LCSS) as follows.

$$LCSS(i, j) = \begin{cases} 0 & \text{if } m = 0 \text{ and } n = 0; \\ LCSS(Rest(i), Rest(j)) + 1 & \text{if } |s_{i,x} - s_{j,x}| < \epsilon \text{ and } |s_{i,y} - s_{j,y}| < \epsilon; \\ \max\{LCSS(Rest(i), j), LCSS(i, Rest(j))\} & \text{otherwise.} \end{cases}$$

where m is the length of trajectory i and n is the length of trajectory j . $Rest(i)$ the subsequence of i without the first element. s_i is the first element in trajectory i .

Table 5: Top ranked trajectory patterns in London.

Rank	Trajectory pattern	P_T	Freq.
1	londoneye → bigben → downingstreet → horseguards → trafalgarsquare	0.0037	2
2	londoneye → bigben → tatemodern	0.0029	2
3	tatemodern → bigben → londoneye	0.0029	3
4	londoneye → bigben → parlamentsquare → westminster	0.0028	2
5	westminster → bigben → downingstreet → horseguards → trafalgarsquare	0.0028	2
6	royalfestivalhall → londoneye → bigben	0.0027	2
7	londoneye → royalfestivalhall → tatemodern	0.0027	3
8	tatemodern → londoneye → royalfestivalhall	0.0027	2
9	londoneye → tatemodern → towerbridge	0.0027	2
10	londoneye → towerbridge → tatemodern	0.0027	2

Table 6: Top ranked locations in London with normalized P_L scores and frequency.

Location	P_L	# User	Location	P_L	# User
londoneye	0.0157	528	southwark	0.0062	57
trafalgarsquare	0.0125	456	stpaulscathedral	0.0058	77
bigben	0.0121	205	downingstreet	0.0053	52
tatemodern	0.0119	491	horseguards	0.0051	25
royalfestivalhall	0.0093	175	londonbridge	0.0049	37
towerbridge	0.0089	185	embankment	0.0047	23
cityhall	0.0077	141	harrods	0.0047	39
waterloobridge	0.0076	198	toweroflondon	0.0046	91
parlamentsquare	0.0075	150	naturalhistorymuseum	0.0046	97
piccadillycircus	0.0074	182	monument	0.0046	59
britishmuseum	0.0074	230	victoriaandalbertmuseum	0.0045	64
gherkin	0.0073	75	bank	0.0044	63
lloyds	0.0070	121	royalacademy	0.0040	34
coventgarden	0.0070	169	oxfordstreet	0.0040	51
buckinghampalace	0.0064	107	bloomsbury	0.0038	27

Based on the defined similarity, we would like to detect the exemplary trajectory patterns, which provide a discrete or heavily quantized description of the whole dataset. The similarity measure $LCSS(i, j)$ can be viewed as how well trajectory i represents trajectory j . Suppose trajectory i is represented by an exemplar trajectory $r(i)$, we can see that trajectory i becomes an exemplar if $r(i) = i$. The sets $\{r(i) | 1 \leq i \leq N\}$ are all the representative trajectory patterns. The optimal set of exemplars corresponds to the ones for which the sum of similarities of each point to its exemplar is maximized.

There are several ways of searching for the optimal exemplars such as vertex substitution heuristic p-median search and affinity propagation [14]. Here we use Frey and Dueck’s affinity propagation algorithm to discover the trajectory pattern exemplars. Affinity propagation considers all data points as potential exemplars and iteratively exchanges messages between data points until it finds a good solution with a set of ex-

emplars. There are two kinds of messages. One type is called “responsibility” message, while the other type is called “availability” message. Responsibility message $r(i, k)$ is sent from trajectory i to candidate exemplar k , which reflects the accumulated evidence for how well-suited candidate k is to serve as the exemplar for trajectory i . Availability message $a(k, i)$ is sent from candidate exemplar k to trajectory i , which represents the accumulated evidence for how appropriate trajectory i is to choose candidate k as its exemplar. In each iteration, we update the responsibility message for all the trajectory and exemplar candidate pairs and let all candidate exemplars compete for ownership of the trajectories. The responsibility score is derived from LCSS similarity scores and availability scores from the trajectory to other potential exemplars as follows.

$$r(i, k) \leftarrow s(i, k) - \max_{k' \neq k} \{a(i, k') + s(i, k')\}$$

where s is the similarity score that is LCSS similarity between trajectories.

Table 7: Trajectory pattern diversification results in London.

Rank	Trajectory pattern
1	bigben → downingstreet → horseguards → trafalgarsquare
2	spitalfields → shoreditch(1) → shoreditch(2) → shoreditch(3) → shoreditch(4)
3	charingcross → londoneye
4	bricklane(1) → bricklane(2)
5	londoneye → royalfestivalhall → tatemodern
6	oldstreet(1) → oldstreet(2)
7	piccadillycircus → soho → oldcomptonstreet
8	londonbridge → cityhall → towerbridge
9	gherkin → lloyds → londonbridge → southwark
10	leicestersquare → chinatown

Meanwhile, we update availability message for all the pairs to determine whether a candidate exemplar is a good representative of the trajectory. The availability score is aggregated from the responsibility scores from the exemplar to other trajectories.

$$a(k, i) \leftarrow \min\{0, r(k, k) + \sum_{i' \neq \{k, i\}} \max(0, r(i', k))\}$$

Self-availability score is updated as follows.

$$a(k, k) \leftarrow \sum_{i': i' \neq k} \max(0, r(i', k))$$

We exchange the messages between trajectories until a set of trajectory exemplars are identified. In the end, $r(i) = \arg \max_k [r(i, k) + a(k, i)]$. Trajectory i becomes an exemplar if $r(i) = i$.

To incorporate the information of ranking results in Section 4, we can give higher ranked trajectories larger self-similarity scores in message passing. Specifically, in responsibility update process, the trajectory with higher self-similarity will have larger responsibility value, which means the trajectory is more appropriate to serve as the exemplars. In this way, the important trajectories identified by our ranking algorithms are more likely to be exemplars.

We list the diversified trajectory patterns in London in Table 7. We find that the results in Table 7 are much more diverse than those in Table 5. Instead of being limited to the top locations, the results in Table 7 cover many other interesting trajectory patterns. For example, the routes 2, 4 and 6 are three trajectory patterns to explore the street art such as graffiti in Shoreditch. Besides the popular trajectory patterns such as *bigben* → *downingstreet* → *horseguards* → *trafalgarsquare*, several other interesting trajectory patterns in different areas of London such as *londonbridge* → *cityhall* → *towerbridge* and *gherkin* → *lloyds* → *londonbridge* → *southwark* are also discovered.

5.3 Discussion In the above exemplar-based algorithm, each iteration updates the message among all the trajectory and exemplar candidate pairs. Since each trajectory can be an exemplar candidate, the time complexity is $O(N_T^2)$, where N_T is the number of the trajectory patterns. Therefore, the total complexity is $O(kN_T^2)$, where k is the iteration number of message passing.

The algorithm satisfies the three desired properties for diversification. First, since we use LCSS to calculate the similarity between trajectories, similar trajectory patterns will be aggregated together. For example, Trajectory 1 (*londoneye* → *bigben* → *downingstreet* → *horseguards* → *trafalgarsquare*) and Trajectory 5 (*westminster* → *bigben* → *downingstreet* → *horseguards* → *trafalgarsquare*) in Table 5 are unlikely to be output together because they have a large similarity. Second, affinity propagation does a good job to find high-quality exemplars, so good representative trajectory patterns are selected. Third, those trajectory patterns ranked highly in our ranking algorithm have high preference scores, so they have higher priority to be exemplars.

6 Experiments

In this section, we describe the experiments to demonstrate the effectiveness of our methods. We propose some measures to evaluate our methods quantitatively and present top ranked trajectory patterns in different cities. We also make use of our trajectory pattern ranking result to recommend locations according to current trajectories.

6.1 Data Set and Baseline Methods We crawled images with GPS records using Flickr API ². The GPS location for each image is represented by a two dimensional vector of longitude and latitude. Each

²<http://www.flickr.com/services/api/>

image is also associated with user-provided tags, of which the number varies from zero to over ten. We collect data for 12 popular cities and the statistics of the dataset are shown in Table 8. The locations are obtained by using the method in Section 3.1. The location descriptions are obtained by using the method in Section 3.2. The trajectory patterns are obtained by using the method in Section 3.3.

Table 8: Statistics of the Flickr datasets (Loc refers to location. Traj refers to Trajectory. Pat refers to trajectory patterns.)

City	Photo	User	Loc	Traj	Pat
Barcelona	7764	1799	201	1320	189
Berlin	5401	1242	167	965	136
Chicago	7418	1589	219	1343	164
DC	5091	1368	173	826	123
Los Angeles	7336	2149	208	1095	105
London	27974	5455	883	4712	1202
Madrid	4014	1072	146	651	131
Milan	5200	1239	127	813	44
New York	16365	3836	568	2692	549
Paris	10826	2699	357	1923	620
Rome	6143	1609	158	1086	193
San Francisco	15841	3047	578	2631	245

To demonstrate the effectiveness of our methods, we compare the performance of the following methods.

1. FreqRank: Rank trajectory patterns by sequential pattern frequency as in Section 3.3.
2. ClassicRank: The method used in [36] to mine classic travel sequences. The classical score of a sequence is the integration of the following three aspects. 1) The sum of hub scores of the users who have taken this sequence. 2) The authority scores of the locations contained in this sequence. 3) These authority scores are weighted based on the probability that people would take a specific sequence. We calculate the user hub score and the location authority score using MUL .
3. TrajRank: Rank trajectory patterns using our method as in Section 4.
4. TrajDiv: Diversify trajectory patterns using our method as in Section 5.

6.2 Comparison of Trajectory Pattern Ranking

To evaluate the results of trajectory pattern ranking, we follow [36] to label the trajectory patterns using three scores (i.e., highly interesting (2), interesting (1), not interesting (0)). We use NDCG(normalized discounted

cumulative gain) to compare the performances of the different methods.

$$NDCG_p = \frac{DCG_p}{IDCG_p} \quad DCG_p = rel_1 + \sum_{i=2}^p \frac{rel_i}{\log_2 i}$$

where $IDCG_p$ is the DCG_p value of ideal ranking list. rel_i is the i -th trajectory’s score.

We show the comparison of trajectory pattern ranking in Figure 2. The results are compared based on the NDCG@10. From Figure 2, we find that TrajRank performs the best on average, which means the ranking mechanism considering the relationship among user, location and trajectory pattern works well. FreqRank and ClassicRank lie between TrajDiv and TrajRank. TrajDiv does not perform as well as other methods, because TrajDiv focuses on selecting the most representative trajectory patterns instead of choosing the most important ones and it trades importance for coverage.

6.3 Comparison of Trajectory Pattern Diversification

To evaluate the results of diversification of trajectory pattern ranking, we use two measures to compare different methods. One is *location coverage*, i.e., the number of covered locations in the top results. A good set of representative trajectory patterns should cover more locations. The other measure is *trajectory coverage* based on the edit distances [6] between different trajectory patterns. The trajectory coverage score is calculated by the summation of the edit distance of each trajectory pattern in the dataset to the closest one in the top result. The score is normalized by the summation of the edit distance of each trajectory pattern to the closest one in the dataset. The larger the trajectory coverage score is, the more representative the result list is.

The location and trajectory coverage of the top 10 results are shown in Figure 3 and Figure 4. The location coverage of ClassicRank is good, but its trajectory coverage is poor. The coverage of TrajRank is low for the reason that it focus on several important locations without diversification. Compare with other methods, TrajDiv covers much more locations and its trajectory coverage is also much higher than other methods. It indicates that TrajDiv selects more representative trajectory patterns than other methods. In other words, TrajDiv gives a more comprehensive view of the trajectory patterns.

6.4 Top Ranked Trajectory Patterns

In Figure 5, we show the top ranked trajectory patterns mined by TrajRank for several cities. In Figure 5(a), people in London first visit London Eye and then go to Big Ben. Next they will go along the Parliament St. to Downing

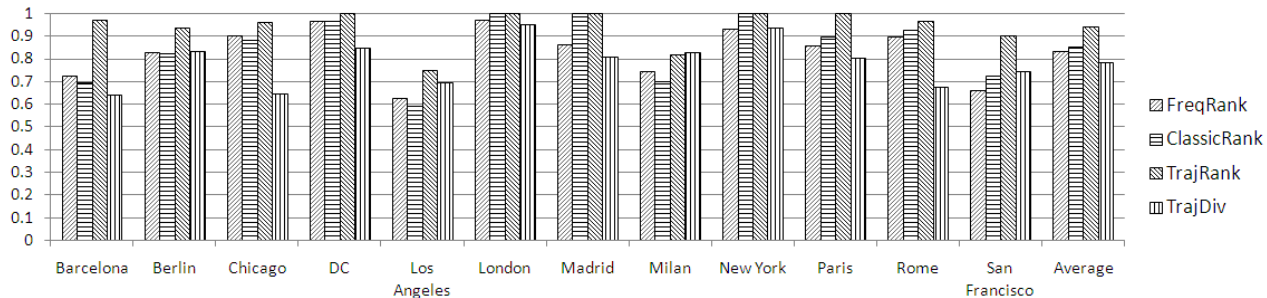


Figure 2: NDCG@10 comparison

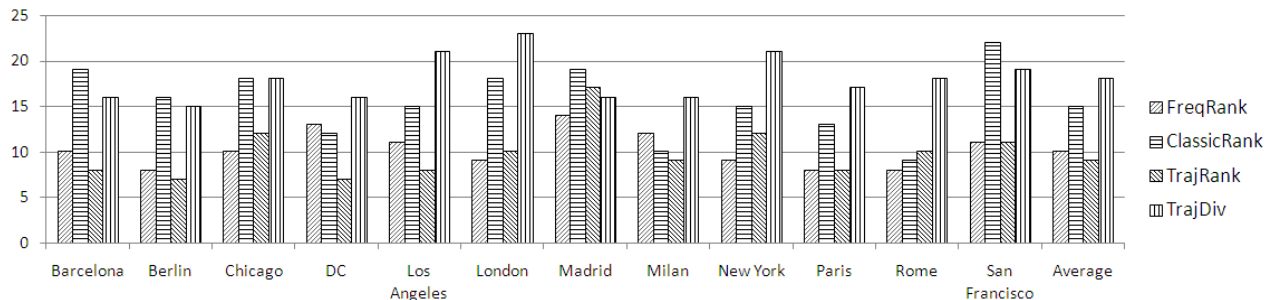


Figure 3: Location coverage comparison

Street. Later they will go through Whitehall and pass Horse Guards on the way to Trafalgar Square. In Figure 5(b), the top ranked trajectory pattern is along the Fifth Avenue. People first start at the Apple flagship store on the Fifth Avenue close to the Central Park. Then they visit St Patrick’s Cathedral and go to Rockefeller Center in the end. In Figure 5(c), people in Paris first visit Eiffel Tower and then go to Louvre Museum. Later they go along the River Seine to Notre Dame de Paris at last.

6.5 Location Recommendation Based on Trajectory Pattern Ranking Trajectory pattern ranking leads to many interesting applications such as location recommendation based on current trajectory. We can consider the current trajectory as the prefix of the extracted trajectory patterns and rank the next location according to the ranking scores of the trajectory patterns (i.e., prefix + next potential location). In Table 9, we list some location recommendations based on current trajectory in London. For example, if you take London Eye as the starting point, you can go to many places of interest. If you visit London Eye and Big Ben, your next destination can be in the Trafalgar Square direction. If you start from London to Tate Modern, the recommendation includes South Bank and Tower Bridge, which means you can either go along River Thames to Tower Bridge or go back to South Bank.

7 Related Work

In this section, we discuss the related work to our study including trajectory mining, GPS data mining, search result diversification and social media spatial mining.

Trajectory Mining. Many research studies have been done in trajectory mining area, such as trajectory clustering [26], classification [25], outlier detection [24], etc.. Trajectory clustering is closely related to our task. Various trajectory clustering algorithms have been developed for the applications like traffic flow, animal movement, hurricane track, etc. Many similarity measure between trajectories have been proposed [34, 5, 2, 6]. In this paper, we use longest common subsequence(LCSS) to calculate the similarity between different routes. Instead of calculating the similarity between trajectories directly, Lee et al. proposed a partition and group framework called TRACCLUS to discover the common sub-trajectory in [26]. After they obtain the distance between partitioned trajectories, a density-based clustering algorithm DBSCAN [13] is applied. Gaffney et al. [15]proposed a model-based clustering method using EM algorithm, where trajectories are represented by a regression mixture model. In this paper, we use an exemplar-based algorithm to select the representative exemplars to diversify the trajectory patterns.

GPS data mining. Lots of GPS-related data have been generated with the increasing prevalence of GPS devices. Zheng et al. mined the interesting locations and

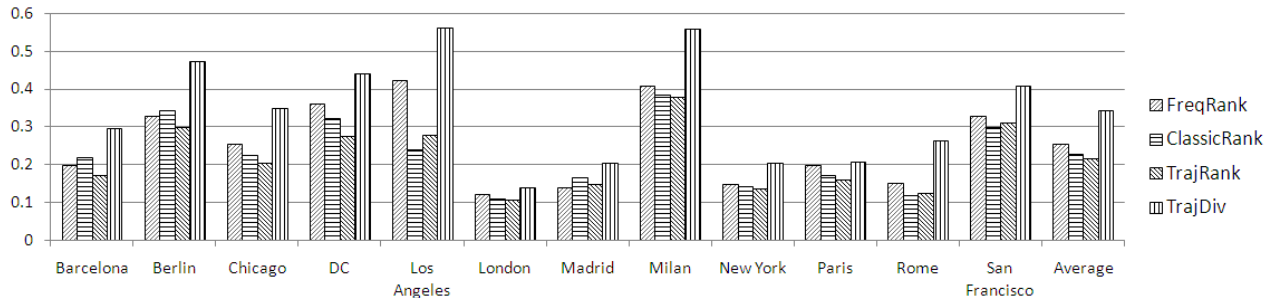


Figure 4: Trajectory coverage comparison

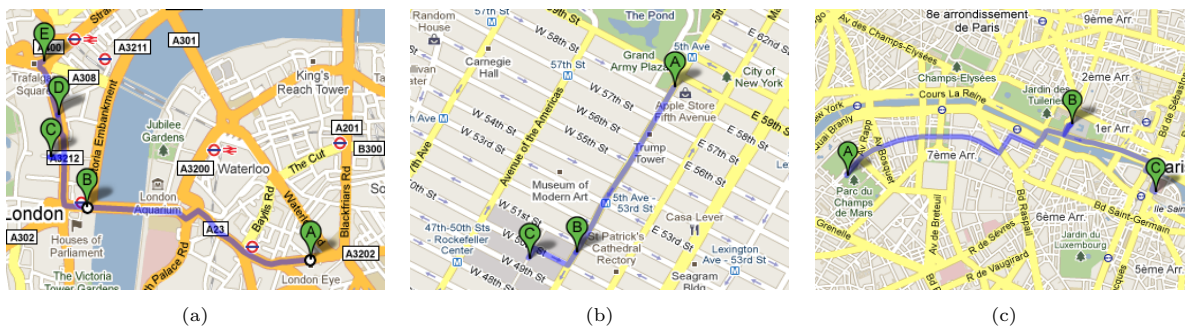


Figure 5: Top ranked trajectory patterns: (a) London, (b) New York City, and (c) Paris.

travel sequences from GPS trajectories in [36]. They modeled location histories with a tree-based hierarchical graph (TBHG) and used a HITS-based inference model to infer the interest of a location and a user travel experience. Classical travel sequence among locations are extracted based on both the interests of these locations and travel experiences. In [35], Zheng et al. made use of GPS log data to recommend location and activity through a collective matrix factorization method. Different from GPS log dataset, the data used in this paper is sparser. GPS logging devices can record the movement continuously, while in our setting only a limited number of uploaded photos with sporadic geoinformation are available. Therefore, those techniques used in GPS log data mining cannot be applied to our case directly. In this paper, we mine the frequent sequential patterns as the consolidated representation of trajectory patterns.

Search result diversification. There are many studies in diversifying search results [18, 1, 32]. However, most of search result diversification studies focus on diversifying results in Web search or recommendation system. How to apply these techniques into trajectory pattern ranking is not straightforward. In this paper we adopt longest common sub-sequence(LCSS) as the similarity measure and use an exemplar-based algorithm to diversify the trajectory pattern ranking results.

Social media spatial mining. With the devel-

opment of the Web 2.0, many researchers focus on mining social media. Amitay et al. described a system called Web-a-Where for associating geography with Web pages. Rattenbury et al. [31] proposed a Scale-structure Identification method to extract the event and place semantics from Flickr tags based on the time and location metadata. To enhance semantic and geographic annotation of web images on Flickr, Cao et al. used Logistic Canonical Correlation Regression (LCCR) to improve the annotation by exploiting the correlation between heterogeneous features and tags. Crandall et al. [11] predicted the locations for the photos on Flickr from the visual, textual and temporal features. In [33], Serdyukov et al. predicted the locations for the Flickr photos by a language model on the user annotations. They extended the language model by tag-based smoothing and cell-based smoothing. Besides mining the location information for Flickr images, blog is also a good source to extract landmarks [22, 19, 20]. Silva et al. [12] also proposed a system for retrieving multimedia travel stories by using location data. In this paper, we explore trajectory pattern ranking in Flickr photo dataset. Some studies focus on mining trip information based on the sequence of locations. Popescu et al. [30, 29] showed how to extract clean trip related information from Flickr metadata. They extracted the place names from Wikipedia and generated the trip by mapping the photo tags to location names. In this pa-

Table 9: Location recommendation based on current trajectory in London

Current trajectory	Recommended next destination
londoneye	bigben, tatemodern, trafalgarsquare, southbank, parlamentsquare, towerbridge, piccadillycircus, buckinghampalace
londoneye → bigben	downingstreet, horseguards, trafalgarsquare, parlamentsquare
londoneye → bigben → downingstreet	horseguards, trafalgarsquare
londoneye → tatemodern	southbank, towerbridge, piccadillycircus
londoneye → trafalgarsquare	buckinghampalace

per, we mine trajectory patterns from the spatial distribution of the photos instead of using the location names mapping, and we also propose a more sophisticated method to rank the mined trajectory patterns. In [9, 8], Choudhury et al. formulated trip planning as directed orienteering problem. In [27], Lu et al. used dynamic programming for trip planning. In [23], Kurashima et al. recommended tourist routes by combining topic models and Markov model. In this paper we apply trajectory pattern mining in geo-tagged social media. We solve the problem of diversified trajectory pattern ranking, which has not been investigated before.

8 Conclusion and Future Work

In this paper, we investigate the problem of trajectory pattern ranking and diversification based on geo-tagged social media. We extract trajectory patterns from Flickr geo-tagged photos using sequential pattern mining and propose a ranking strategy that considers the relationships among user, location and trajectory. To diversify the ranking results, we use an exemplar-based algorithm to discover the representative trajectory patterns. We test our methods on the photos of 12 different cities from Flickr and show our methods outperform the baseline methods in trajectory pattern ranking and diversification.

The results in this paper also point to several interesting future directions. First, so far we apply our methods to the metropolitan areas and it would be interesting to mine meaningful trajectory patterns outside big cities. Second, it is interesting to incorporate the duration constraints in our trajectory pattern ranking and diversification framework. Third, we plan to consider the semantic correlations between the locations in addition to current algorithms. It will be interesting to incorporate our previous work on geological annotation [4] and the current mining algorithms.

References

- [1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM*, pages 5–14, 2009.
- [2] T. Bozkaya, N. Yazdani, and Z. M. Özsoyoglu. Matching and indexing sequences of different lengths. In *CIKM*, pages 128–135, 1997.
- [3] L. Cao, J. Luo, A. Gallagher, X. Jin, J. Han, and T. S. Huang. A worldwide tourism recommendation system based on geotagged web photos. In *ICASSP*, 2010.
- [4] L. Cao, J. Yu, J. Luo, and T. S. Huang. Enhancing semantic and geographic annotation of web images via logistic canonical correlation regression. In *ACM Multimedia*, pages 125–134, 2009.
- [5] L. Chen and R. T. Ng. On the marriage of lp-norms and edit distance. In *VLDB*, pages 792–803, 2004.
- [6] L. Chen, M. T. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In *SIGMOD Conference*, pages 491–502, 2005.
- [7] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8):790–799, 1995.
- [8] M. D. Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu. Automatic construction of travel itineraries using social breadcrumbs. In *HT*, pages 35–44, 2010.
- [9] M. D. Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu. Constructing travel itineraries from tagged geo-temporal breadcrumbs. In *WWW*, pages 1083–1084, 2010.
- [10] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.
- [11] D. J. Crandall, L. Backstrom, D. P. Huttenlocher, and J. M. Kleinberg. Mapping the world’s photos. In *WWW*, pages 761–770, 2009.
- [12] G. C. de Silva and K. Aizawa. Retrieving multimedia travel stories using location data and spatial queries. In *ACM Multimedia*, pages 785–788, 2009.
- [13] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.
- [14] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.
- [15] S. Gaffney and P. Smyth. Trajectory clustering with mixtures of regression models. In *KDD*, pages 63–72,

- 1999.
- [16] F. Giannotti, M. Nanni, D. Pedreschi, and F. Pinelli. Trajectory pattern analysis for urban traffic. In *GIS-IWCTS*, pages 43–47, 2009.
- [17] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory pattern mining. In *KDD*, pages 330–339, 2007.
- [18] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *WWW*, pages 381–390, 2009.
- [19] Q. Hao, R. Cai, X.-J. Wang, J.-M. Yang, Y. Pang, and L. Zhang. Generating location overviews with images and tags by mining user-generated travelogues. In *ACM Multimedia*, pages 801–804, 2009.
- [20] Q. Hao, R. Cai, J.-M. Yang, R. Xiao, L. Liu, S. Wang, and L. Zhang. Travelscope: standing on the shoulders of dedicated travelers. In *ACM Multimedia*, pages 1021–1022, 2009.
- [21] M. T. Heath. *Scientific Computing: An Introductory Survey*. McGraw-Hill, New York, second edition, 2002.
- [22] R. Ji, X. Xie, H. Yao, and W.-Y. Ma. Mining city landmarks from blogs by graph modeling. In *ACM Multimedia*, pages 105–114, 2009.
- [23] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura. Travel route recommendation using geotags in photo sharing sites. In *CIKM*, pages 579–588, 2010.
- [24] J.-G. Lee, J. Han, and X. Li. Trajectory outlier detection: A partition-and-detect framework. In *ICDE*, pages 140–149, 2008.
- [25] J.-G. Lee, J. Han, X. Li, and H. Gonzalez. *TraClass*: trajectory classification using hierarchical region-based and trajectory-based clustering. *PVLDB*, 1(1):1081–1094, 2008.
- [26] J.-G. Lee, J. Han, and K.-Y. Whang. Trajectory clustering: a partition-and-group framework. In *SIGMOD Conference*, pages 593–604, 2007.
- [27] X. Lu, C. Wang, J.-M. Yang, Y. Pang, and L. Zhang. Photo2trip: Generating travel routes from geo-tagged photos for trip planning. In *ACM Multimedia*, 2010.
- [28] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Trans. Knowl. Data Eng.*, 16(11):1424–1440, 2004.
- [29] A. Popescu and G. Grefenstette. Deducing trip related information from flickr. In *WWW*, pages 1183–1184, 2009.
- [30] A. Popescu, G. Grefenstette, and P.-A. Moëllic. Mining tourist information from user-supplied collections. In *CIKM*, pages 1713–1716, 2009.
- [31] T. Rattenbury, N. Good, and M. Naaman. Towards automatic extraction of event and place semantics from flickr tags. In *SIGIR*, pages 103–110, 2007.
- [32] R. L. T. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In *WWW*, pages 881–890, 2010.
- [33] P. Serdyukov, V. Murdock, and R. van Zwol. Placing flickr photos on a map. In *SIGIR*, pages 484–491, 2009.

- [34] M. Vlachos, D. Gunopulos, and G. Das. Rotation invariant distance measures for trajectories. In *KDD*, pages 707–712, 2004.
- [35] 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.
- [36] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *WWW*, pages 791–800, 2009.

A Location description generation

Given location L , we assume that the tags of the photos in location L are generated from a background model B and a corresponding location model θ_L . The background model B is represented by a tag distribution $p(t|B)$ where $\sum_t p(t|B) = 1$, while the location model θ_L is represented by another tag distribution $p(t|\theta_L)$ where $\sum_t p(t|\theta_L) = 1$. Given the photo collection C_L of location L , we want to find $p^*(t|\theta_L)$ that maximizes the following likelihood.

$$\log p(C_L|\theta) = \sum_t c(t, L) \log[(1 - \lambda)p(t|\theta_L) + \lambda p(t|B)]$$

where t is tag and $c(t, L)$ is the count of the users that contribute tag t in location L . Here we use the count of the users instead of the count of the photos for the reason that one user may take many photos at the same location. Therefore, the count of the users is better to represent the popularity of the tag for the specific location. λ is the tradeoff between the background model B and the location model θ_L . We set λ as 0.8 empirically here. $p(t|B)$ can be obtained by considering the tags in all the locations, i.e., $p(t|B) = \frac{c(t)}{\sum_t c(t)}$ where $c(t)$ is the count of the users that contribute tag t in the whole collection, and $p^*(t|\theta_L)$ that maximizes the above likelihood can be estimated by EM algorithm. We introduce hidden variable z_t , where $z_t = 1$ means that tag t is from background B and $z_t = 0$ means that tag t is from location model θ_L .

In E-step,

$$p(z_t = 1|t) = \frac{\lambda p(t|C)}{\lambda p(t|C) + (1 - \lambda)p(t|\theta_L)}$$

In M-step,

$$p(t|\theta_L) = \frac{c(t)(1 - p(z_t = 1|t))}{\sum_t c(t)(1 - p(z_t = 1|t))}$$

We calculate $p^*(t|\theta_L)$ for location L and use the top tags to annotate location L . The descriptions for top locations in London is listed in Table 1. From Table 1, we can find that the tags such as *uk* and *london* are not included in the top tags for the location with the above smoothing.