Mining Significant Semantic Locations From GPS Data

Xin Cao[†] Gao Cong[†] Christian S. Jensen[‡]

[†]School of Computer Engineering, Nanyang Technological University, Singapore
xcao1@e.ntu.edu.sg, gaocong@ntu.edu.sg

[‡]Department of Computer Science, Aarhus University, Denmark
csj@cs.au.dk

ABSTRACT

With the increasing deployment and use of GPS-enabled devices, massive amounts of GPS data are becoming available. We propose a general framework for the mining of semantically meaningful, significant locations, e.g., shopping malls and restaurants, from such data.

We present techniques capable of extracting semantic locations from GPS data. We capture the relationships between locations and between locations and users with a graph. Significance is then assigned to locations using random walks over the graph that propagates significance among the locations. In doing so, mutual reinforcement between location significance and user authority is exploited for determining significance, as are aspects such as the number of visits to a location, the durations of the visits, and the distances users travel to reach locations. Studies using up to 100 million GPS records from a confined spatio-temporal region demonstrate that the proposal is effective and is capable of outperforming baseline methods and an extension of an existing proposal.

1. INTRODUCTION

Mobile devices, e.g., phones and navigation systems, with built-in GPS (Global Positioning System) receivers are capable of generating massive amounts of GPS records that capture geo-location, time, and a number of other attributes such as heading and speed.

The GPS records from a moving object approximate the object's trajectory, which is a continuous function from time to space. Trajectories capture how moving objects use geographical space. For example, they capture the locations that the objects visit along with the durations of the visits. When projecting a trajectory onto space, a route results.

This paper develops techniques that exploit massive amounts of GPS records collected from multiple users for identifying top-k *significant semantic locations*. In particular, the aim is to identify locations that are semantically meaningful to users, e.g., shopping malls, restaurants, or tourist attractions, rather than simply identifying raw geographical coordinates.

Automatic extraction of meaningful stay locations and assigning significance to these are fundamentally important and useful. For example, the outcome can serve as travel recommendations that take into account location significance and the distance from the user to a location. This is somewhat analogous to web queries where web pages are ranked in terms of their inherent importance as well as their relevance to a query: location significance plays the role of web page importance, and the distance from a user to a location plays the role of the relevance of a web page to a query. Another example is that the outcome can be combined with the so-called location-aware keyword query [6].

Top-k hot semantic locations can be computed with regard to a specific context such as a certain user group (e.g., teens, senior citizens, tourists) or a certain time period (e.g., a vacation period or a time of day). This may be done by exploiting different GPS data for different contexts.

GPS data is already attracting attention. Scores of communities-based web sites, e.g., targeting hikers, bikers, and cyclists, enable users to share routes and trajectories. However, these sites simply focus on the sharing of trajectories. Some recent studies [2, 10, 24] consider the extraction of locations from GPS records, but do not consider location significance.

Notably, Zheng et al. [23] pioneer the extraction and ranking of locations from GPS data. We build on this work by considering semantic rather than raw locations. Semantic locations must be considered throughout to obtain the best possible results. For example, this avoids situations where different raw locations that turn out to represent the same semantic location are ranked as separate locations; and it avoids cases where a single raw location represents more than one semantic location. A post-processing step cannot fix such problems.

It is challenging to propose a model for location ranking that is capable of exploiting multiple factors. The mutual reinforcement of user authority and location significance is considered by Zheng et al. However, as their approach uses the HITS algorithm [11], improper weights may be assigned to links, as discussed by Bharat and Henzinger [5] (to be explained in Section 3.3.2). Our model is able to exploit this mutual reinforcement relationship better. More significantly, our framework is capable of systematically exploiting other factors such as the relationships between locations, the distances between locations, and the durations of visits.

Our framework includes techniques for extracting semantic locations from GPS records. This problem is nontrivial because GPS records with different coordinates may represent the same semantic location. Thus, we group stay points extracted from GPS records such that each group represents a unique real-world location. We leverage existing clustering algorithms for the initial grouping, and we enhance the clustering results by taking into account the patterns of visits to the clusters and the similarity of the semantics of the clusters obtained from a reverse geocoder and yellow pages.

Proceedings of the VLDB Endowment, Vol. 3, No. 1

Next, we propose a new model for location ranking that is capable of propagating significance between locations, supports mutual reinforcement between location significance and user authority, and is capable of exploiting factors such as the number of visits to a location and the durations of the visits.

To achieve this, we model the connections between locations and also the connections between locations and users using a two-layered graph with location-location and user-location components. Intuitively, if a location, e.g., a hotel, is connected via trajectories to many other significant locations, e.g., restaurants, the hotel is also considered as significant. Further, locations visited by authoritative users are considered as more significant; and users gain authority by visiting significant locations [23].

A PageRank-like model [18] can capture the location-location interactions, and a HITS-like model [11] can capture the reinforcement between users and locations. However, they cannot capture both the location-location and the user-location components. We propose a new model that is able to accommodate both.

Intuitively, a location visited by a user from a place far away is more likely to be more important than a place visited from a nearby location; and the longer the durations of the stays at a location, the more important the locations is considered to be. Our model accommodates these aspects.

In summary, the paper's contributions are threefold. First, we propose new techniques for extracting semantic locations from GPS data. Second, we propose a new model for assigning significance to the extracted locations. Third, we report on empirical studies with large quantities of GPS data that suggest that the proposed framework is (i) capable of improving the abilities of the OPTICS [1] and K-means clustering algorithms to extract semantic locations and is (ii) capable of significantly outperforming several baseline methods, including rank-by-visits, rank-by-durations, and an extension of an existing approach.

Section 2 states the problem addressed. Section 3 details the proposed location mining framework, including the extraction of stays and their clustering into semantic locations, the construction of the link structures connecting users and locations, and the ranking of locations using random walk models. Section 4 reports on the experimental study. Section 5 concludes and discusses research directions. An appendix contains supplementary matter.

2. PROBLEM STATEMENT

We proceed to cover the data model used in this paper and the problem addressed. Appendix A delves into application scenarios.

2.1 Data Model

While the proposed framework applies to GPS data in general, we assume that the GPS data represents vehicle movements. For specificity, we assume a sampling frequency of one record per second when a vehicle moves.

DEFINITION 1. ***GPS record:*** *A GPS record $\mathcal{G}$ is a five-tuple: $\langle u, t, x, y, s \rangle$, where u is the ID of the user for which $\mathcal{G}$ is recorded, t is the timestamp, x and y are the Euclidean coordinates, and s is the vehicle's instant speed as reported by the GPS device.*

An example is ⟨1715, 2007-07-13 17:20:36, 544456, 6335497, 0⟩ where the coordinates are given in the UTM (Universal Transverse Mercator) coordinate system.

By ordering the records from a user by the timestamp t, we obtain a representation of the user's trajectory.

DEFINITION 2. ***Trajectory:*** *A trajectory $\mathcal{TR}$ of a user is a sequence of GPS records $\mathcal{G}$ for the user that are ordered by the timestamp t of the records, $\mathcal{TR} = \mathcal{G}_1 \rightarrow \cdots \rightarrow \mathcal{G}_i \rightarrow \cdots \rightarrow \mathcal{G}_n$.*

A trajectory describes the movement of a user. The original GPS records can be seen as the set of the trajectories of all the users, denoted as $\mathcal{S}_{\mathcal{TR}}$. From a trajectory, we can compute the sequence of locations visited by the corresponding user as well as how long the user stays in the locations.

We are interested in locations where users have stayed for longer than some predefined duration, as such locations are more likely to be interesting. We extract all such stays from $\mathcal{S}_{\mathcal{TR}}$.

DEFINITION 3. ***Stay point:*** *A stay point $\mathcal{P}$ is a pair $\langle \mathcal{G}, \Delta t \rangle$, where $\mathcal{G}$ is the ID of a GPS record that represents the stay and Δt is the duration of the stay.*

Each time a user stays at a location, we can obtain a corresponding stay point. Thus, a location visited multiple times by one or multiple users obtains multiple stay points. Stay points for the same real-world location almost certainly contain different coordinates, although they are close to each other. The challenge is to derive consolidated locations from a collection of stay points.

Ideally, each consolidated location corresponds to a unique real-world location, e.g., a shopping mall or a tourist attraction.

DEFINITION 4. ***Semantic Location:*** *A semantic location $\mathcal{L}$ is a cluster of stay points and is represented by a four-tuple $\langle x, y, sem, sl \rangle$, where x and y are the centroid of the cluster, sem is the semantics of the location, and sl is a set of stay points associated with the semantic location. We denote the set of all extracted semantic locations as $\mathcal{S}_{\mathcal{L}}$*

For example, semantic location ⟨555122, 6321850, "Aalborg Zoo", $\{864, 1354, 57720, \ldots\}$ ⟩ has centroid $(555122, 6321850)$, "Aalborg Zoo" as the semantics, and $\{864, 1354, 57720, \ldots\}$ as GPS record identifiers that represent stay points.

We are now ready to define the location history of a user.

DEFINITION 5. ***Location History:*** *The location history $\mathcal{H}$ of a user is defined as a sequence $\mathcal{H} = \mathcal{L}_1 \rightarrow \cdots \rightarrow \mathcal{L}_i \rightarrow \cdots \rightarrow \mathcal{L}_n$ of semantic locations $\mathcal{L}$.*

Given a set of GPS trajectories $\mathcal{S}_{\mathcal{TR}}$ we can build the set of all users' location histories, $\mathcal{S}_{\mathcal{H}}$.

2.2 Problem Characterization

We aim to identify and assign significance to semantic locations based on large amounts of GPS records.

Just as web queries issued to a search engine do not have ground truth results, neither do the results for our problem. Rather, we combine a range of indicators of "interestingness" in order to obtain significant locations. Specifically, we consider the following indicators of the significance of a location: (1) the number of visits, (2) the durations of the visits, and (3) the distances users travel to visit locations.

Such indicators may be combined in various ways using location histories. Specifically, various kinds of reinforcement may be applied. Thus, locations that are visited together with significant locations may be assigned increased significance. And users who tend to visit significant locations may be assigned increased authority, which may then be used for assigning increased significance to locations visited by these users.

While a framework that is capable of taking into account many such indicators is likely to have advantages over less capable frameworks, the interpretation of a result by a user is subjective. Different users will have different interests and preferences. We thus do not distinguish among "significant," "hot," "interesting," "attractive," and "popular" as predicates of locations.

By considering location histories selectively, results with different meanings can be obtained. For example, by only considering

GPS data from tourists, or international tourists, it is possible to target interesting locations at specific user groups. This filtering may be done based on demographic data about the users who contribute the data, which may be available to, e.g., rental car and insurance companies. While important, this is orthogonal to the paper's contribution.

We also note that while some trips may carry little meaning, the aggregate results obtained from very large numbers of trips are likely to be robust to such noise and thus yield meaningful results. This is analogous to the use of clicks or links for the assignment of importance to web pages.

Given a set of GPS records, we first pre-process the data to obtain the set of trajectories, $\mathcal{S}_{\mathcal{TR}}$. We address two sub-problems:

1. Development of techniques that are capable of identifying *semantically* meaningful locations $\mathcal{S}_{\mathcal{L}}$ from GPS data. This involves extracting the set of semantic locations $\mathcal{S}_{\mathcal{L}}$ from the GPS trajectories $\mathcal{S}_{\mathcal{TR}}$, and building the location history $\mathcal{H}$ for each user to get the location history set $\mathcal{S}_{\mathcal{H}} = \{\mathcal{H}_1, \ldots, \mathcal{H}_m\}$, where m is the number of users.

2. Development of techniques capable of assigning significance to locations. We use the location histories $\mathcal{S}_{\mathcal{H}}$, to compute the top-$k$ semantic locations ranked according to significance.

3. PROPOSED SOLUTION

Following a solution overview in Section 3.1, we cover the extraction of semantic locations in Section 3.2 and the ranking of locations in Section 3.3.

3.1 Solution Overview

We first extract stay points from the GPS records. We then group stay points such that each group represents a real-world location. We utilize street addresses, semantics, etc. to improve two existing clustering algorithms, OPTICS [1] and K-means [21].

We then form location histories and use these for constructing a two-layered graph that models connections between users and locations, and connections between locations. An example of such a graph is given in Figure 1. The graph has a user layer, where nodes represent users, and a location layer, where nodes represent locations. An edge exists between two locations if a user has traveled between them, and an edge exists between a user and a location if the user has visited the location.

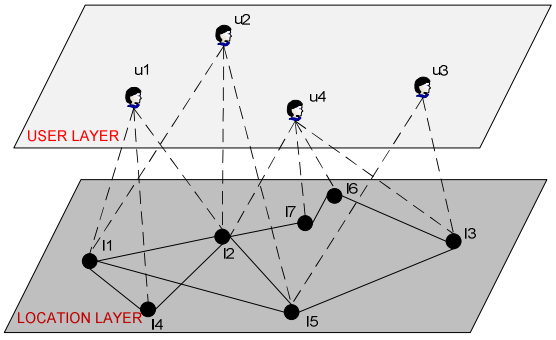


Figure 1: Example two-layered graph of users and locations

As discussed in Section 2.2, we use novel indicators of location significance. Specifically, we propose a new model that enables the propagation of significance among locations and the reinforcement between user authority and location significance using the two-layered graph.

Our setting differs from the standard web setting in important respects. Specifically, 1) our graph has two levels while the web graph has one, making it resemble the location-location graph, which however, is undirected while the web graph is directed; 2) the web graph is unweighted while we use weights on the location-location edges (to capture the number of trips between locations; 3) our graph is spatial, meaning, e.g., that the willingness of a user to travel a long distance to reach a location indicates high significance; no such distance notion exists on the Internet; and 4) the user-location graph is weighted (to capture the durations of visits).

3.2 Extracting Location Histories

We cover the extraction of stay points in Appendix B and the extraction of semantic locations from stay points next.

We aim at extracting locations from stay points such that the locations are meaningful to users. One idea is to apply a reverse geocoder to the stay points. A reverse geocoder attempts to find the closest addressable location within a certain distance. The result returned for a stay point includes the street address and the coordinates of the street address. When two stay points are mapped to the same addressable location, it is likely that they represent the same semantic location.

However, the reverse geocoding of a stay point involves a call to an external API (in or case, Google Maps), which is very time consuming when there are many stay points. Additionally, even if two stay points are reverse geocoded differently, they may still represent the same semantic location. This suggests that we should exploit other information when extracting semantic locations.

We thus extract the semantic locations in two steps. We first cluster the stay points. We then enhance the clustering by taking onto account a variety of additional information, to be detailed shortly. A final cluster is expected to represent a single, unique semantic location.

For the first step, we use the existing clustering algorithms OPTICS and K-means. For the second step, we repeatedly enhance the results obtained by utilizing the street addresses, the semantics, and the user visit patterns. We denote the resulting method as SEM-CLS (semantics-enhanced clustering). The method involves two steps: split and merge.

In the split step, we sample points from each cluster, reverse geocode them to obtain street addresses, and then obtain their semantics using a yellow pages directory; if the sample points in a cluster have different semantics, we split the cluster since it may contain multiple semantic locations.

In the merge step, we merge the clusters that may represent the same semantic location. In order to determine when to merge two clusters, we represent each cluster by a four-tuple $(\vec{l_u}, t_{as}, t_{ae}, \vec{l_s})$ of features: the user list vector $\vec{l_u}$, which contains the IDs of users who have visited the cluster together with the number of visits by each user; the average stay duration t_{as}, which is the average duration of all the visits; the average entry time (time of a day) t_{ae}, which is the average starting time of all the visits to the cluster; and the semantic list vector $\vec{l_s}$, which contains all the semantics that may apply to the cluster.

The visitors, the stay duration, and the entry time reveal patterns of the users who have visited the location represented by a cluster. If two nearby locations exhibit similar visiting patterns, the two locations are possibly the same. We thus compute the similarity of two clusters based on the four features. First, the vector space model is used to compute the similarity between two visitor lists and also the similarity between two semantic lists. The similarity of the average stay duration is computed as the smaller one divided by the larger one, and the same is done for the average entry time. The final similarity score of two clusters is computed by a simple linear combination of the four scores. Formally, the similarity between two clusters c_1 and c_2 is computed as follows:

$$Sim(c_1, c_2) = \frac{\vec{l_{u1}} \cdot \vec{l_{u2}}}{\|\vec{l_{u1}}\|\|\vec{l_{u2}}\|} + \frac{\vec{l_{s1}} \cdot \vec{l_{s2}}}{\|\vec{l_{s1}}\|\|\vec{l_{s2}}\|} + \frac{min(t_{as1}, t_{as2})}{max(t_{as1}, t_{as2})} + \frac{min(t_{ae1}, t_{ae2})}{max(t_{ae1}, t_{ae2})} \quad (1)$$

3.3 Ranking of Locations

We first cover the construction of the two-layered graph. We then present several baseline location ranking methods in Section 3.3.2, and we present the new location ranking models.

3.3.1 Two-Layered Graph

The two-layered graph consists of two inter-connected sub graphs, a user-location graph and a location-location graph.

DEFINITION 6. ***User-location graph:*** *The user-location graph is a weighted undirected bipartite graph $\mathbb{G}_{UL} = (\mathbb{U}, \mathbb{V}, \mathbb{E}_{UL}, \mathbb{W}_{UL})$, where $\mathbb{U}$ is a set of nodes that represent users, $\mathbb{V}$ is a set of nodes that represent locations, $\mathbb{E}_{UL}$ is a set of edges that represent visits, and the edge weights $\mathbb{W}_{UL}$ describe the numbers of visits to a location by a user.*

Given m users and n locations, we build an $m \times n$ adjacency matrix $\mathbf{M}$ for $\mathbb{G}_{UL}$. Formally, $\mathbf{M} = [v_{ij}]$, $0 \leq i < m$, $0 \leq j < n$, where v_{ij} represents how many times the i^{th} user has visited the j^{th} location.

DEFINITION 7. ***Location-location graph:*** *The location-location graph is a weighted undirected graph $\mathbb{G}_{LL} = (\mathbb{V}, \mathbb{E}_{LL}, \mathbb{W}_{LL})$, where $\mathbb{V}$ is as in the previous definition, $\mathbb{E}_{LL}$ is the set of edges that represent trips, and the weights $\mathbb{W}_{LL}$ of edges describe the numbers of transitions between locations.*

Given n locations, we define an $n \times n$ adjacency matrix $\mathbf{C}$ for $\mathbb{G}_{LL}$. Formally, $\mathbf{C} = [c_{ij}]$, $0 \leq i, j < n$, where c_{ij} represents the times that a user has driven between the i^{th} and the j^{th} location.

Let the location histories of the three users in Figure 1 be $\mathcal{H}_{\mathcal{U}_1} = \{\mathcal{L}_2 \rightarrow \mathcal{L}_1 \rightarrow \mathcal{L}_4, \mathcal{L}_2 \rightarrow \mathcal{L}_4\}$, $\mathcal{H}_{\mathcal{U}_2} = \{\mathcal{L}_5 \rightarrow \mathcal{L}_2, \mathcal{L}_5 \rightarrow \mathcal{L}_1, \mathcal{L}_2 \rightarrow \mathcal{L}_1, \}$, $\mathcal{H}_{\mathcal{U}_3} = \{\mathcal{L}_3 \rightarrow \mathcal{L}_5\}$, and $\mathcal{H}_{\mathcal{U}_4} = \{\mathcal{L}_7 \rightarrow \mathcal{L}_6, \mathcal{L}_3 \rightarrow \mathcal{L}_6, \mathcal{L}_2 \rightarrow \mathcal{L}_7 \rightarrow \mathcal{L}_6\}$. Then:

$$\mathbf{M} = \begin{bmatrix} 1 & 2 & 0 & 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 3 & 2 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 0 & 2 & 0 & 1 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 0 & 2 & 0 \end{bmatrix}$$

We notice that for each user, the number of visits to the user's home may be very large. Because we aim at mining hot public locations and to preserve the users' privacy, locations such as a user's home are removed. We consider a location as a home location if it is visited frequently by the same user at night.

3.3.2 Baseline Methods

We consider three baseline methods, two of which are presented in Appendix C, namely **rank-by-visits** (which is also used as baselines in the work by Zheng et al. [23]), and **rank-by-durations**.

The idea behind the third approach [23], which we cover as a baseline, is to take into account the authority of users instead of treating all users equally. It is thus assumed that hot locations are visited by more authoritative users, and that authoritative users visit more interesting locations. The approach applies the HITS model [11] on $\mathbb{G}_{UL}$ to find authoritative users and interesting locations. Using HITS, the approach computes a hub score and an authority score for each node. The hub scores of users show their travel experience, and the authority scores of locations show the significance of the locations. Additional descriptions of this baseline are given in Appendix C.

Although the approach aims to exploit user authority for location ranking, the model used in the approach limits the performance of the approach. If a location is only visited by one user, but many times, the HITS algorithm used assigns a very high authority score to the location and also a very high hub score to the user. However, a location visited only by a single user or very few users is intuitively not particularly significant. Section 4 offers empirical insight into this problem. Recall also that approach considers raw locations rather than semantic locations.

In the example graph, the final authority vector is {0.1651, 0.2676, 0.0661, 0.0935, 0.1286, 0.1675, 0.1117}, and the ranked list is {2, 6, 1, 5, 7, 4, 3}. The effect of weights can be seen more clearly in the following example: if the user $\mathcal{U}_4$ makes the trip $\mathcal{L}_7 \rightarrow \mathcal{L}_6$ twice, the result becomes very different. The final authority vector becomes {0.0506, 0.1557, 0.0946, 0.0304, 0,0404, 0.3589, 0.2692}, and the ranked list becomes {6, 7, 2, 3, 1, 5, 4}. Notice that $\mathcal{L}_6$ and $\mathcal{L}_7$ have larger authority scores, and the user $\mathcal{U}_4$ also gains a large hub score improperly, simply because this user visits her/his preferred locations more times.

3.3.3 Random Walks on $\mathbb{G}_{UL}$

This model exploits reinforcement between users' travel experiences and locations' significance while avoiding the problem of the last baseline approach. It uses the randomized HITS model, which defines a random walk on $\mathbb{G}_{UL}$.

This model is insensitive to small perturbations and thus is more stable than what is obtained by using the original HITS model. The randomized HITS model uses a normalized version of matrix $\mathbf{M}$. Bharat and Henzinger [5] show that mutually reinforcing relationships between nodes may assign improper weights to links; the normalization solves this problem.

We define the column vector w_{user} as the hub vector and the column vector w_{loc} as the authority vector. Formally, the randomized HITS model applied to $\mathbb{G}_{UL}$ can be described as follows:

$$w_{loc}^{k+1} = (\epsilon \mathbf{M^T_{row}} + (1-\epsilon)\mathbf{E_1}) \cdot w_{user}^{k}$$
$$w_{user}^{k+1} = (\epsilon \mathbf{M_{col}} + (1-\epsilon)\mathbf{E_2}) \cdot w_{loc}^{k+1}, \quad (2)$$

where k is the number of iterations, $\mathbf{M_{col}}$ is the column stochastic matrix of $\mathbf{M}$ ($\mathbf{M_{col}}$ is computed by normalizing each column in $\mathbf{M}$), $\mathbf{M_{row}}$ is the row stochastic matrix of $\mathbf{M}$ ($\mathbf{M_{row}}$ is computed by normalizing each row in $\mathbf{M}$), $\mathbf{E_1}$ is a matrix with all elements equal to $\frac{1}{m}$, $\mathbf{E_2}$ is a matrix with all elements equal to $\frac{1}{n}$, and ϵ is the "teleport probability," which represents the probability of a random surfer teleporting from a location node to a user node (resp. from a user node to a location node) instead of following the links in $\mathbb{G}_{UL}$.

We initialize w_{user} as $(\frac{1}{m}, ..., \frac{1}{m})$ and w_{loc} as $(\frac{1}{n}, ..., \frac{1}{n})$, and we then use the power iteration algorithm to calculate the vectors w_{user} and w_{loc}. The vector w_{loc} is used to rank all the locations.

Assume $\epsilon = 0.85$ and consider the example graph. If user $\mathcal{U}_4$ makes the trip $\mathcal{L}_7 \rightarrow \mathcal{L}_6$ twice, the final stable authority vector becomes {0.1374, 0.2148, 0.1003, 0.1003, 0.1390, 0.1730, 0.1351}, and the ranked list is {2, 6, 5, 1, 7, 3, 4}. Compared with the results of the original HITS [23], $\mathcal{L}_6$ and $\mathcal{L}_7$ have lower rankings because the weights are normalized.

3.3.4 Random Walks on $\mathbb{G}_{LL}$

Unlike the previous method, this method exploits the location-location links: the significance of one location is affected positively by the significance of the locations connecting to it.

Inspired by the PageRank algorithm [18] that is developed for

web link graphs, we perform random walks on $\mathbb{G}_{LL}$ to rank locations. Our model differs from the model used in the PageRank algorithm in two respects: 1) weights are associated with our links, while the web link graph has no link weights and 2) the location-location graph is undirected while the web link graph is directed.

We are aware that weights are also explored in other applications of the random walk model (e.g., [4]) and that random walks have been applied to undirected graphs (e.g., for document summarization [17]). However, no previous work performs a random walk on a spatial graph like the location-location graph.

Using a column vector w_{loc}, the random walk model applied to $\mathbb{G}_{LL}$ can be described as follows:

$$w_{loc}^{k+1} = (\alpha \mathbf{C}_{\mathbf{row}}^{\mathbf{T}} + (1-\alpha)\mathbf{E}) \cdot w_{loc}^{k}, \tag{3}$$

where $\mathbf{C_{row}}$ is the row stochastic matrix of $\mathbf{C}$ defined in Section 3.3.1 ($\mathbf{C_{row}}$ is computed by normalizing each row in $\mathbf{C}$), $\mathbf{E}$ is a matrix with all elements set to $\frac{1}{n}$, and α is the well known "damping factor" (set to 0.85, following the PageRank algorithm [18]).

We initialize w_{loc} as a uniformly distributed vector $(\frac{1}{n}, ..., \frac{1}{n})$ and then apply the power iteration algorithm until the vector w_{loc} converges to a stable state.

This model takes into account the relations between locations, but ignores the effects of different users. It thus cannot distinguish between visits by more versus less authoritative users.

3.3.5 Unified Link Analysis Framework

We propose a new unified probabilistic model that takes into account both the links between users and locations and the links between locations, as captured by the two-layered graph. No existing approach is able to accommodate both aspects: the approach based on randomized HITS model (Section 3.3.3) cannot model the relations between locations, while the approach based on PageRank-like model (Section 3.3.4) cannot model the mutual reinforcement between users and locations. Additionally, the proposed unified model is able to incorporate the durations of visits and the distances between locations.

The unified model uses the structure of the two-layered graph to build a Markov chain. The proposed unified model thus can be characterized as random walks on the Markov chain built on both $\mathbb{G}_{UL}$ and $\mathbb{G}_{LL}$ in which the states are nodes. We first present the unified model and then explain the process of random walks.

We define three transition probabilities for the Markov chain: $p(\mathcal{U}_k|\mathcal{L}_i)$, the transition probability to a user node $\mathcal{U}_k$ from a location node $\mathcal{L}_i$; $p(\mathcal{L}_i|\mathcal{U}_k)$, the transition probability to a location node $\mathcal{L}_i$ from node $\mathcal{U}_k$; and $p(\mathcal{L}_i|\mathcal{L}_j,\mathcal{U}_k)$, the transition probability to location node $\mathcal{L}_i$ from $\mathcal{L}_j$ for user $\mathcal{U}_k$.

Given m users and n locations, we capture the transition probabilities (i.e., $p(\mathcal{L}_i|\mathcal{U}_k)$) from user nodes to location nodes in an $(m \times n)$ matrix $\mathbf{N}_{UL}$, and, similarly, the transition probabilities (i.e., $p(\mathcal{U}_k|\mathcal{L}_i)$) from location nodes to user nodes in an $(n \times m)$ matrix $\mathbf{N_{LU}}$. We capture the stationary distributions of the random walks for users and locations by two column vectors w_{user} and w_{loc}, respectively, to rank the significance of locations. The unified model, denoted as **Unified**, can then be described as:

$$\begin{aligned}
&w_{loc}^{k+1} = \mathbf{N_{LU}} \cdot w_{user}^{k}, \quad w_{user}^{k+1} = \mathbf{N}_{UL} \cdot w_{loc}^{k+1} \\
&p(\mathcal{U}_k|\mathcal{L}_i) = \epsilon \frac{Num(\mathcal{U}_k,\mathcal{L}_i)}{Num(\mathcal{L}_i)} + (1-\epsilon)\frac{1}{m} \\
&p(\mathcal{L}_i|\mathcal{U}_k) = \sum_{j=1}^{n} p(\mathcal{L}_j|\mathcal{U}_k) p(\mathcal{L}_i|\mathcal{L}_j,\mathcal{U}_k) \\
&p(\mathcal{L}_i|\mathcal{L}_j,\mathcal{U}_k) = \alpha \frac{Num(\mathcal{L}_i,\mathcal{L}_j,\mathcal{U}_k)}{Num(\mathcal{L}_j,\mathcal{U}_k)} + (1-\alpha)\frac{1}{n},
\end{aligned} \tag{4}$$

where $Num(\mathcal{U}_k,\mathcal{L}_i)$ is the number of visits to $\mathcal{L}_i$ by $\mathcal{U}_k$ and $Num(\mathcal{L}_i)$ is the total number of visits to $\mathcal{L}_i$; $p(\mathcal{L}_i|\mathcal{L}_j,\mathcal{U}_k)$ is the probability of $\mathcal{L}_i$ being visited by $\mathcal{U}_k$ from $\mathcal{L}_j$; and $Num(\mathcal{L}_i,\mathcal{L}_j,\mathcal{U}_k)$ is the number of trips between locations $\mathcal{L}_j$ and $\mathcal{L}_i$ by $\mathcal{U}_k$, and $Num(\mathcal{L}_j,\mathcal{U}_k)$ is the total number of visits by $\mathcal{U}_k$ to $\mathcal{L}_j$. Parameters ϵ and α are the "teleport probability" in the random walks, and they control the effect of the constant probability $1/m$ and $1/n$, respectively. In turn, these two are used to smooth $p(\mathcal{U}_k|\mathcal{L}_i)$ and $p(\mathcal{L}_i|\mathcal{L}_j,\mathcal{U}_k)$, respectively.

As mentioned, the unified model can be interpreted as a random walk model on both $\mathbb{G}_{UL}$ and $\mathbb{G}_{LL}$. Thus, at location node $\mathcal{L}_i$, the random walk proceeds to user node $\mathcal{U}_k$ with the probability $p(\mathcal{U}_k|\mathcal{L}_i)$. Intuitively, the transition probability to a user node $\mathcal{U}_k$ reflects the significance of the user to the location based on the user's previous behavior, i.e., the frequency of the user visiting $\mathcal{L}_i$. At $\mathcal{L}_i$ the random walk can either follow a link in the graph $\mathbb{G}_{UL}$, or it can teleport to a random user node with probability $1-\epsilon$. Similarly, at user node $\mathcal{U}_k$ the random walk proceeds to location node $\mathcal{L}_i$ with probability $p(\mathcal{L}_i|\mathcal{U}_k)$.

To compute $p(\mathcal{L}_i|\mathcal{U}_k)$, we perform random walks on $\mathbb{G}_{LL}$ for each user node $\mathcal{U}_k$. The transition probability of the random walks on $\mathbb{G}_{LL}$ from location $\mathcal{L}_j$ to location $\mathcal{L}_i$ for user $\mathcal{U}_k$ reflects the trips by the user between the two locations, and the probability is computed by $p(\mathcal{L}_i|\mathcal{L}_j,\mathcal{U}_k)$ in Equation 4. From the stationary distributions of the random walks, we get $p(\mathcal{L}_i|\mathcal{U}_k)$ for each location $\mathcal{L}_i$.

The unified model exploits both user-location and location-location reinforcement. In fact, the models in Sections 3.3.3 and 3.3.4 are special cases of the unified model.

Reduction to the Model in Section 3.3.3. If we disregard the location-location reinforcement, i.e., the locations are independent so that $p(\mathcal{L}_i|\mathcal{L}_j,\mathcal{U}_k) = p(\mathcal{L}_i|\mathcal{U}_k)$, we compute the conditional probabilities in Equation 4 as follows:

$$\begin{aligned}
p(\mathcal{U}_k|\mathcal{L}_i) &= \epsilon \frac{Num(\mathcal{U}_k,\mathcal{L}_i)}{Num(\mathcal{L}_i)} + (1-\epsilon)\frac{1}{m} \\
p(\mathcal{L}_i|\mathcal{U}_k) &= \epsilon \frac{Num(\mathcal{U}_k,\mathcal{L}_i)}{Num(\mathcal{U}_k)} + (1-\epsilon)\frac{1}{n},
\end{aligned}$$

where and $Num(\mathcal{U}_k)$ is the total number of visits by $\mathcal{U}_k$. In Appendix F, we show that with these definitions, Equation 4 can be reduced to Equation 2.

Reduction to the Model in Section 3.3.4. If we disregard the user-location reinforcement, i.e., we treat all the users equally, we can rewrite the probability $p(\mathcal{L}_i|\mathcal{L}_j,\mathcal{U}_k)$ as follows:

$$p(\mathcal{L}_j|\mathcal{L}_i) = \alpha \frac{Num(\mathcal{L}_j,\mathcal{L}_i)}{Num(\mathcal{L}_i)} + (1-\alpha)\frac{1}{n},$$

where $Num(\mathcal{L}_j,\mathcal{L}_i)$ is the number of trips between $\mathcal{L}_i$ and $\mathcal{L}_j$. In Appendix F, we show that in this case, Equation 4 can be reduced to Equation 3.

We proceed to present an algorithm that implements the unified model. This algorithm uses the power iteration method to compute the stationary distribution of vector w_{loc} that ranks locations. Although the ranking is done offline, we show that it is possible to reduce the unified model to a simplified model that enables a more efficient algorithm.

THEOREM 1. *Let $\mathbf{P}$ be the $n \times n$ (n is the number of locations) transition matrix of the Markov chain on $\mathbb{G}_{UL}$ and $\mathbb{G}_{LL}$. An element p_{ij} of the matrix represents the transition probability between two locations, and it is computed as:*

$$p_{ij} = p(\mathcal{L}_j|\mathcal{L}_i) = \sum_{k=1}^{m} p(\mathcal{U}_k|\mathcal{L}_i) p(\mathcal{L}_j|\mathcal{L}_i,\mathcal{U}_k) \tag{5}$$

The unified model in Equation 4 can be reduced to the following:

$$w_{loc}^{k+1} = \mathbf{P^T} \cdot w_{loc}^{k} \quad (6)$$

PROOF. *See Appendix D.*

An algorithm that utilizes Equation 6 is more efficient than one that uses Equation 4. The pseudocode of the proposed algorithm is in Appendix E.

Based on the unified model, we proceed to present an extended model, denoted as **ST-Unified**, that is able to incorporate stay durations and distances between locations.

First, each stay has a duration Δt, and the longer the stay at a location, the more significant the location is assumed to be. To account for durations in the unified model in Equation 5, we extend the conditional probabilities $p(\mathcal{U}_k|\mathcal{L}_i)$ that so far only considered the numbers of visits:

$$p'(\mathcal{U}_k|\mathcal{L}_i) = \epsilon(\varepsilon\frac{Num(\mathcal{U}_k,\mathcal{L}_i)}{Num(\mathcal{L}_i)} + (1-\varepsilon)\frac{Dur(\mathcal{U}_k,\mathcal{L}_i)}{Dur(\mathcal{L}_i)}) + \frac{1-\epsilon}{m}$$

where $Dur(\mathcal{U}_k,\mathcal{L}_i)$ is the duration at $\mathcal{L}_i$ of $\mathcal{U}_k$, $Dur(\mathcal{L}_i)$ is the total duration of stays at $\mathcal{L}_i$, and ε controls the effect of the durations.

Second, given several equally attractive locations, e.g., convenience stores, a user is expected to prefer to travel to the nearest one. Thus, the longer the distance a user is willing to travel to reach a location, the more significant the location is considered to be. To account for this, we extend the definition of $p(\mathcal{L}_j|\mathcal{L}_i,\mathcal{U}_k)$, the probability of user $\mathcal{U}_k$ traveling from location $\mathcal{L}_i$ to location $\mathcal{L}_j$. If there are trips between $\mathcal{L}_j$ and $\mathcal{L}_i$ in the location history of $\mathcal{U}_k$, the definition becomes:

$$p'(\mathcal{L}_j|\mathcal{L}_i,\mathcal{U}_k) = \alpha(\eta\frac{Num(\mathcal{L}_j,\mathcal{L}_i,\mathcal{U}_k)}{Num(\mathcal{L}_i,\mathcal{U}_k)} + (1-\eta)\frac{Dist(\mathcal{L}_j,\mathcal{L}_i)}{\sum_h Dist(\mathcal{L}_i,\mathcal{L}_h)}) + \frac{1-\alpha}{n},$$

where $Dist(\mathcal{L}_j,\mathcal{L}_i)$ is the distance between $\mathcal{L}_j$ and $\mathcal{L}_i$. The larger this distance is, the larger the conditional probability becomes. Parameter η controls the effect of distances.

In **ST-Unified**, the random walks on $\mathbb{G}_{UL}$ consider two factors: the stay duration and the reinforcement between visitors and locations. The random walks on $\mathbb{G}_{LL}$ consider another two factors: the distances and the connection between locations.

For implementation, the transition matrix of **ST-Unified** is defined as $\mathbf{P'}$, and its element p'_{ij} is computed as:

$$p'_{ij} = \sum_{k=1}^{m} p'(\mathcal{U}_k|\mathcal{L}_k)p'(\mathcal{L}_j|\mathcal{L}_i,\mathcal{U}_k)$$

Column vector w_{loc} can then be computed as:

$$w_{loc}^{k+1} = \mathbf{P'^T} \cdot w_{loc}^{k} \quad (7)$$

THEOREM 2. *Both **Unified** and **ST-Unified** converge utilizing power iteration.*

PROOF. *See Appendix D.*

4. EXPERIMENTAL EVALUATION

We cover the experimental settings and then the results semantic location extraction and ranking in turn.

4.1 Settings and Evaluation Methods

We use GPS data collected from 119 cars driven by young drivers during the period 01/01/2007–31/03/2008. The data set contains in excess of 0.1 billion GPS records.

Following the approach in Appendix B, we obtain 159,062 stays from the data set. After data cleaning, we obtain 76,139 stay points. Details on stay extraction are given in Appendix G.1, and Figure 2 shows distribution of a sample of the stay points.

We also use three subsets of the whole dataset for evaluating the proposed methods. DATA1: Data restricted to the town of Nørresundby; this dataset contains 352 locations. DATA2: 1,508 locations that were visited by at least 5 users. DATA3: Using the data from DATA2, if a location is visited by a user more than 5 times, the number of visits to the location by the user is counted as 5. We create these subsets in order to obtain datasets with different properties.

All experiments are conducted on a computer with 2.3 GHz CPU and 2 GB main memory; our proposals are implemented using C#.

4.1.1 Extraction of Semantic Locations

Algorithms: We evaluate the performance of the enhanced method SEM-CLS (Section 3.2), comparing with the OPTICS and K-means algorithms. For OPTICS [1], we use software provided by its authors, and we use WEKA[1] for the K-means method. We elaborate on how to obtain the ground truth for semantic locations in Appendix G.3.

Metrics: We use three metrics, namely entropy, purity [21], and normalized mutual information (NMI) [15] that are used widely to evaluate the performance of clustering algorithms when a ground truth exists. The smaller the entropy, the better a clustering method performs. For the other two measures, the larger, the better.

4.1.2 Location Ranking

Ranking models: We evaluate the approaches presented in Section 3.3: the randomized HITS on $\mathbb{G}_{UL}$ (**U-L**), the random walks on $\mathbb{G}_{LL}$ (**L-L**), the unified model on the two-layered graph (**Unified**), and the unified model taking into account the durations and distances (**ST-Unified**). We compare with the three baseline approaches [23], i.e., rank-by-visits, rank-by-durations, and the HITS-based approach.

Ground truth for ranking: We compute the top-50 locations for each ranking method, combine them, and subject them to expert annotators in order to assess their significance. The locations are shown on a map, and their associated semantics are given to the annotators, who do not know which methods produced which locations.

Four individuals familiar with Nordjylland perform the annotation, applying a label from Table 1 to each location. For each location, we use the average score from all the annotators.

Score	Specification
2	Very interesting to most people in general and recommended
1	An OK location to most people in general
0	Neutral to most people in general
-1	Not interesting to most people in general and not recommended
-2	I have no idea of what it is

Table 1: Annotation specification

Metrics: To capture the performance of our location significance ranking approaches, we apply three popular ranking performance metrics, namely Mean Average Precision (MAP), Precision@n, and nDCG (normalized Discounted Cumulative Gain). Precision@n is the fraction of the top-n locations retrieved that are significant. When computing MAP and Precision@n, only the locations with score above 1.5 are considered as significant. nDCG computes the relative-to-the-ideal performance [9].

[1]http://www.cs.waikato.ac.nz/ml/weka/

Parameter selection: We use the parameters ϵ in the random walk model on $\mathbb{G}_{UL}$ and α in the random walk model on $\mathbb{G}_{LL}$. Both are set to 0.85, thus following the PageRank [18] and the randomized HITS [16] algorithms. In **ST-Unified**, ε and η are set to 0.3.

4.2 Experimental Results

4.2.1 Extracting Semantic Locations

To evaluate the grouping of stay points into locations by the different clustering methods, we adjust the parameters of each method to obtain a number of clusters (and thus locations) that is similar to the number of ground truth locations.

We set k to 7,100 for K-means, and the initial k mean points are selected randomly. For OPTICS, we set ϵ to 17 meters and minPts to 2 (a cluster must have at least two stay points). In the SEM-CLS method, threshold th is set to 3. As a result, K-means returns 7,056 clusters (a bit below k due to empty clusters), and OPTICS returns 7,088 clusters.

We apply SEM-CLS to enhance the results of K-means and OPTICS. We vary the number of sampling points for each cluster from 2 to 5 for SEM-CLS. The results are shown in Tables 2 and 3. The "# Samples" column gives the number of sampling points in each cluster to be reverse geocoded, and the number of generated clusters (locations) is given in the last column.

# Samples	Purity	Entropy	NMI	# Clusters
K-means	0.8630	0.4770	0.5909	7056
2	0.8743	0.4438	0.6096	6013
3	0.8852	0.4195	0.6236	6304
4	0.8990	0.3867	0.6311	6510
5	0.9096	0.3598	0.6375	6631

Table 2: K-means versus SEM-CLS on K-means with different numbers of sampling points

# Samples	Purity	Entropy	NMI	# Clusters
OPTICS	0.8699	0.4567	0.6526	7088
2	0.8828	0.4292	0.6682	6055
3	0.8979	0.3847	0.6703	6358
4	0.9129	0.3573	0.6724	6555
5	0.9259	0.3139	0.6747	6680

Table 3: OPTICS versus SEM-CLS on OPTICS using different numbers of sampling points

We can see that OPTICS performs better than K-means, and we see that SEM-CLS, when applied to both clustering algorithms, is able to improve on them in terms of all three metrics. This suggests that the use of the semantic and visit patterns for the splitting and merging of clusters in SEM-CLS is effective.

We also observe that SEM-CLS improves as more points are sampled and reverse geocoded. However, calling an external API to perform reverse geocoding consumes substantial time. Hence, there is a tradeoff between the effectiveness and the efficiency.

Efficiency. K-means takes 12 minutes to finish, and OPTICS takes 4 minutes. The enhancement method SEM-CLS takes less than 1 second. Additional details are available in Appendix G.4.

4.2.2 Ranking the Significance of Locations

Table 4 depicts the MAP, Precision@n, and nDCG@n results of the different ranking models on the whole dataset. Each column corresponds to an approach. **ST-Unified** performs the best in terms of all metrics, and **Unified** performs better than **U-L** and **L-L**, and these all perform better than the three baseline methods.

U-L exploits the mutual reinforcement between the authority of users and the significance of locations. **L-L** uses information on the number of visits, which is also used by the rank-by-visits method. It outperforms rank-by-visits because it propagates the significance between locations.

Unified improves on **U-L** and **L-L** because it combines the two graphs $\mathbb{G}_{UL}$ and $\mathbb{G}_{LL}$ while the latter two only consider one subgraph. **ST-Unified** performs better than **Unified** because it exploits the combined graph and also the distance between locations and stay durations.

L-L performs worse than the other proposed models. The reason may be that it does not capture the authority of users. The model is also significantly affected by the locations visited many times by very few users—such locations have large in-degree weights.

The HITS-based approach [23] on $\mathbb{G}_{UL}$ performs worse than the other two baselines. Another study [23] shows that it performs better than rank-by-visits. This may be because the mutual reinforcement between users and locations can assign unsuitable weights to links according to the analysis of the HITS algorithm by Bharat and Henzinger [5], as discussed in Sections 3.3.2–3.3.3. For example, we find a location visited by only one user, but more than 200 times. As a result, the user has a very high hub score of 0.968, and the top-10 locations returned by the HITS-based approach are all locations visited by this user. However, none of them are significant locations. The method rank-by-durations outperforms rank-by-visits slightly by incorporating visit durations.

The top-10 results of the four proposed ranking models are given in Appendix G.5.

Efficiency. The last row of Table 4 gives the runtimes of the different methods. **Unified** and **ST-Unified** use power iteration. When using Equation 6, the runtime is about 4 seconds, while it is 130 seconds wuen using Equation 4. Recall that location ranking is done offline and not at query time. Pre-computed location rankings are utilized to answer top-k queries as described in Appendix A.

Results on other datasets. Tables 5–7 shows the results of the HITS-based approach [23] and the proposed methods. We see that **ST-Unified** consistently performs the best and that **Unified** performs better than **U-L** and **L-L**.

	HITS	U-L	L-L	Unified	ST-Unified
MAP	0.5424	0.7231	0.6908	0.7664	**0.8090**
P@20	0.5	0.7	0.6	0.75	**0.8**
nDCG@20	0.8588	0.9269	0.8861	0.9302	**0.9545**

Table 5: Ranking results using different ranking models on DATA1

	HITS	U-L	L-L	Unified	ST-Unified
MAP	0.3982	0.6952	0.6691	0.7816	**0.8202**
P@20	0.45	0.7	0.6	0.75	**0.80**
nDCG@20	0.8092	0.9290	0.9166	0.9456	**0.9501**

Table 6: Ranking results using different ranking models on DATA2

	HITS	U-L	L-L	Unified	ST-Unified
MAP	0.7008	0.7579	0.6647	0.7982	**0.8321**
P@20	0.6	0.7	0.6	0.75	**0.8**
nDCG@20	0.9202	0.9534	0.9132	0.9552	**0.9658**

Table 7: Ranking results using different ranking models on DATA3

The results in Table 5 on DATA1 that concerns a small region are similar to the results on the whole data, meaning that the region size does not affect the results.

As discussed in Section 3.3.2 and in the coverage of the results reported in Table 4, a possible problem of the HITS-based approach is that locations visited many times by very few users are ranked too highly. To avoid this potential problem, DATA2 does not contain locations visited by very few users. Table 6 shows that DATA2 does

	Rank-by-visits	Rank-by-durations	HITS [23]	U-L	L-L	Unified	ST-Unified
MAP	0.2020	0.2126	0.062	0.3748	0.3020	0.4060	**0.4274**
P@20	0.45	0.45	0.1	0.75	0.6	0.9	**0.95**
P@50	0.36	0.38	0.12	0.68	0.52	0.74	**0.76**
nDCG@20	0.8261	0.8324	0.4555	0.9411	0.9031	0.9678	**0.9897**
nDCG@50	0.7678	0.7747	0.4380	0.9226	0.8827	0.9402	**0.9717**
Runtime(ms)	103	107	1536	2209	3540	4234	4318

Table 4: Ranking results using different ranking models on the whole dataset

help the HITS-based approach, although the other models remain better.

In DATA2, every location has at least 5 visitors, but a user may still visit a location many times. For example, we find that in DATA2, one location is visited by one user 101 times while being visited very few times by other users. Thus both the user and the locations visited by the user obtain very high ranking scores in the HITS-based approach.

To eliminate such effects, we set for each user who visits a location more than 5 times the number of visits to 5, thus obtaining DATA3. Table 7 shows that the performance of the HITS-based approach is then close to that of the U-L model that is based on the randomized HITS algorithm, although **ST-Unified** still performs the best.

Parameter Study. ST-Unified uses four parameters: ϵ, α, ε, and η. The best performance is obtained when ϵ and α are in the range 0.7–0.9 and ε and η are around 0.3. Additional details are available in Appendix G.6.

5. CONCLUSIONS AND FUTURE WORK

Motivated by the proliferation of GPS data, we propose a framework that encompasses new techniques for extracting semantically meaningful geographical locations from such data and for the ranking of these locations according to their significance.

We model the relationships between locations and also the relationships between locations and users with a two-layered graph. The ranking model we propose takes into account significance propagation among locations; mutual reinforcement between location significance and user authority; and aspects such as the number of visits to a location, the durations of the visits, and the distances between locations.

An empirical study demonstrates that our proposals are capable of extracting semantic locations and of performing better rankings than several baseline methods and previous work.

Several promising directions for future work exist. First, it is of interest to study the processing of geo-context aware queries (discussed in Appendix A) based on the hot semantic locations. Second, it is of interest to mine "hot" semantic patterns from GPS trajectories.

Acknowledgments

The authors thank the anonymous reviewers for their insightful comments. The research was conducted when the authors were employed at Aalborg University, Denmark. C. S. Jensen is an Adjunct Professor at University of Agder, Norway.

6. REFERENCES

[1] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In *Proc. SIGMOD*, pp. 49–60, 1999.

[2] D. Ashbrook and T. Starner. Learning significant locations and predicting user movement with GPS. In *Proc. ISWC*, pp. 101–108, 2002.

[3] D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.

[4] A. Balmin, V. Hristidis, and Y. Papakonstantinou. Objectrank: Authority-based keyword search in databases. In *Proc. VLDB*, pp. 564–575, 2004.

[5] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proc. SIGIR*, pp. 104–111, 1998.

[6] G. Cong, C. S. Jensen, and D. Wu. Efficient retrieval of the top-k most relevant spatial web objects. *PVLDB*, 2(1):337–348, 2009.

[7] C. H. Q. Ding, X. He, P. Husbands, H. Zha, and H. D. Simon. Pagerank: Hits and a unified framework for link analysis. In *Proc. SDM*, pp. 249–253, 2003.

[8] R. Hariharan and K. Toyama. Project lachesis: parsing and modeling location histories. In *Proc. Geographic Information Science*, pp. 106–124. 2004.

[9] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM TOIS*, 20(4):422–446, 2002.

[10] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting places from traces of locations. *Mobile Computing and Communications Review*, 9(3):58–68, 2005.

[11] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *JACM*, 46(5):604–632, 1999.

[12] A. Langville and C. Meyer. Deeper inside PageRank. *Internet Mathematics*, 1(3):335–380, 2004.

[13] L. Liao, D. J. Patterson, D. Fox, and H. Kautz. Building personal maps from GPS data. *Annals of the New York Academy of Sciences*, 1093:249–265, 2006.

[14] J. Liu, O. Wolfson, and H. Yin. Extracting semantic location from outdoor positioning systems. In *Proc. MDM*, p. 73, 2006.

[15] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[16] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. In *Proc. SIGIR*, pp. 258–266, 2001.

[17] J. Otterbacher, G. Erkan, and D. R. Radev. Using random walks for question-focused sentence retrieval. In *Proc. HLT/EMNLP*, pp. 915–922, 2005.

[18] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. TR 1999-66, Stanford InfoLab, 1999.

[19] M.-H. Park, J.-H. Hong, and S.-B. Cho. Location-based recommendation system using Bayesian user's preference model in mobile devices. In *Proc. UIC*, pp. 1130–1139, 2007.

[20] F. Schmid and K.-F. Richter. Extracting places from location data streams. In *Proc. UbiGIS 2006*, 2006.

[21] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co., 2005.

[22] K. Yatani, K. Tamura, K. Hiroki, M. Sugimoto, and H. Hashizume. Toss-it: Intuitive information transfer techniques for mobile devices using toss and swing actions. *IEICE Transactions*, 89-D(1):150–157, 2006.

[23] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from GPS trajectories. In *Proc. WWW*, pp. 791–800, 2009.

[24] C. Zhou, N. Bhatnagar, S. Shekhar, and L. G. Terveen. Mining personally important places from GPS tracks. In *Proc. ICDE Workshops*, pp. 517–526, 2007.

APPENDIX

A. APPLICATION SCENARIOS

The extracted top-k hot semantic locations can be used by a recommendation system to fulfill the following two types of recommendation queries.

The first kind of query is the location-aware top-k query. It returns a ranked list of places according to a user's location. For example, a user staying in a hotel would like to find the top-10 significant locations that are close to the hotel.

Formally, a location-aware top-k query is represented by $\mathcal{Q} = \{k, \iota\}$, where k is the number of locations requested, and $\iota = (x, y)$ is the query location. The query retrieves the top-k hot locations with regard to the query location ι.

Clearly, the popularity of locations should be taken into account to answer such queries. In addition, the interestingness of a location to a user is affected by the distance from the user's location (the query location) to the semantic location. A user is probably willing to visit a less interesting nearby location instead of slightly more interesting, but further away location.

Hence, to answer this kind of recommendation query, we take into consideration both the inherent significance of a location and its distance to the query location.

A location-aware top-k query is similar to a web query. For a web query, web pages are ranked by a combination of their inherent, relative importance (e.g., as computed by PageRank), and their relevance to the query. Here the inherent importance corresponds to the inherent significance of locations, and the relevance corresponds to the distance between locations and the query location.

In addition, the extracted hot locations can also be combined with the location-aware top-k keyword query [6], which takes into account both the distances of locations to the query point and the relevance of the textual descriptions of locations to the query keywords.

The second kind of query is the context-aware recommendation query. Recall that a GPS record $\mathcal{G}$ contains user information (the user ID u), temporal information (the timestamp t), and spatial information (the coordinates x and y). The concept of a multidimensional view of the data from multidimensional databases and data warehousing can be incorporated into hot-location recommendation. In other words, we can take into account a user dimension, a time dimension, and a spatial dimension when providing context-aware recommendations. Example queries include "find the top-30 locations visited by persons between the age 20 and 30," "find the top-10 locations in May, 2007," and "find the top-20 locations in the city center."

The context-aware recommendation query takes parameters $\langle k, us, ts, rs\rangle$. It retrieves the top-$k$ locations from the GPS data that satisfy the user-dimension predicate us, the time-dimension predicate ts, and the spatial-dimension predicate rs. Obviously, if we do not impose any context constraints, the context-aware query reduces to the problem of ranking semantic locations as defined in Section 2.2.

On the user dimension different groups of users have various interests, and the hot locations from different user groups reveal the group's preferences. On the time dimension the significance of locations may vary with the time of day since locations have different opening hours (e.g. bars are usually open at night). Another example is that the significance of locations may vary across the four seasons (e.g. outdoor tourist sites may not be visited in winter as frequently as in summer). On the spatial dimension, the significance of a location may vary with respect to different regions. If a location is mostly visited sequentially from other locations within the same region, this indicates that the location will have high local importance, but it may not be that interesting in the global view. For example, a community kiosk may display the property.

Note that the two kinds of queries are not orthogonal and can be combined. For example, a query could be "find the top-10 locations in summer near my hotel." Similar to a location-aware query, to process this query we still need to compute the popularity of locations and the influence of locations to the query location. However, the popularity of locations will be computed by a context-aware recommend query, i.e., we use only the GPS trajectories generated in summer to extract and rank hot locations.

B. EXTRACTING STAY POINTS

When a car is turned off, the GPS-enabled device attached to the car also stops recording. We utilize this feature to extract stay points. If the duration Δt between two consecutive records from a user is larger than a threshold t_{th}, the two records represent a stay. The record that is the end of the previous trip is denoted $\mathcal{G}_{end}$ and the record that is the start of the next is denoted $\mathcal{G}_{start}$.

When a car is started, the car's GPS device needs sometime to start working. Hence the ending record best captures the location of the stay, and we define a stay as $\mathcal{P} = (\mathcal{G}_{end}, \Delta t)$.

The value of the threshold t_{th} affects the number of extracted stay points. We studied the effect of t_{th} on number of stay points using different for on our data, and we also reverse geocode the generated stay points and measured the number of distinct street addresses. We found it useful to use a threshold of 10 minutes, which yields some 76,000 stay points and more than 7,000 street addresses for our data. This value is also used in previous work [3].

Substantial data cleaning of the raw GPS data is also needed to obtain a robust approach. A number of checks were carried out, and the approach was generally to repair or discard data when problems were identified. The specifics are left out for brevity.

C. DETAILS ON BASELINES

We consider three baseline methods:

Rank-by-visits. This method assumes that the more a location is visited, the more significant it is. This method ranks the locations according to the number of visits by all the users. Ties are broken by the sum of durations.

Rank-by-durations. This method is based on the rank-by-visits method. It takes into account the durations of visits: It ranks the locations according to the sum of durations of all visits to a location. Ties are broken by the number of visits.

Zheng et al. [23]. Formally, the approach utilizes the matrix $\mathbf{M}$ from Section 3.3.1 to compute an authority score column vector $\mathbf{a}$ for the locations, and a hub score column vector $\mathbf{h}$ for the users.

$$\mathbf{a^{k+1}} = \mathbf{M}^T \cdot \mathbf{h^k}, \qquad \mathbf{h^{k+1}} = \mathbf{M} \cdot \mathbf{a^{k+1}} \tag{8}$$

The locations are ranked by the authority score vector $\mathbf{a}$.

D. PROOFS

D.1 Proof of Theorem 1

PROOF. Let $p(\mathcal{U}_k)$ denote the kth element in w_{user}, and $p(\mathcal{L}_i)$ denote the ith element in w_{loc}. From Equation 4, we know that:

$$\begin{aligned} p(\mathcal{L}_i) &= \sum_{k=1}^{m} p(\mathcal{U}_k) p(\mathcal{L}_i|\mathcal{U}_k) \\ &= \sum_{k=1}^{m} p(\mathcal{U}_k) \sum_{j=1}^{n} p(\mathcal{L}_j|\mathcal{U}_k) p(\mathcal{L}_i|\mathcal{L}_j, \mathcal{U}_k) \end{aligned}$$

According to the bayesian theorem, we can get:

$$p(\mathcal{L}_i) = \sum_{k=1}^{m}\sum_{j=1}^{n} p(\mathcal{U}_k)p(\mathcal{L}_j|\mathcal{U}_k)p(\mathcal{L}_i|\mathcal{L}_j,\mathcal{U}_k)$$
$$= \sum_{k=1}^{m}\sum_{j=1}^{n} p(\mathcal{L}_j)p(\mathcal{U}_k|\mathcal{L}_j)p(\mathcal{L}_i|\mathcal{L}_j,\mathcal{U}_k)$$
$$= \sum_{j=1}^{n} p(\mathcal{L}_j)\sum_{k=1}^{m} p(\mathcal{U}_k|\mathcal{L}_j)p(\mathcal{L}_i|\mathcal{L}_j,\mathcal{U}_k)$$

Since $p(\mathcal{L}_i) = \sum_{j=1}^{n} p(\mathcal{L}_j)p(\mathcal{L}_i|\mathcal{L}_j)$, we can obtain:

$$p(\mathcal{L}_i|\mathcal{L}_j) = \sum_{k=1}^{m} p(\mathcal{U}_k|\mathcal{L}_j)p(\mathcal{L}_i|\mathcal{L}_j,\mathcal{U}_k)$$

After changing the subscript, we have:

$$p(\mathcal{L}_j|\mathcal{L}_i) = \sum_{k=1}^{m} p(\mathcal{U}_k|\mathcal{L}_i)p(\mathcal{L}_j|\mathcal{L}_i,\mathcal{U}_k)$$

This completes the proof. □

D.2 Proof of Theorem 2

PROOF. Because we add a constant probability to both $p(\mathcal{U}_k|\mathcal{L}_i)$ and $p(\mathcal{L}_j|\mathcal{L}_i,\mathcal{U}_k)$, it is assured that all the elements in $\mathbf{P}$ and $\mathbf{P}'$ is larger than zero. Hence both matrices are **irreducible**. All elements in the two matrices are transition probabilities and are thus **positive**. In matrix $\mathbf{P}$, it can be shown that:

$$\sum_{j=1}^{n} p_{ij} = \sum_{j=1}^{n}\sum_{k=1}^{m} p(\mathcal{U}_k|\mathcal{L}_i)p(\mathcal{L}_j|\mathcal{L}_i,\mathcal{U}_k)$$
$$= \sum_{k=1}^{m} p(\mathcal{U}_k|\mathcal{L}_i)\sum_{j=1}^{n} p(\mathcal{L}_j|\mathcal{L}_i,\mathcal{U}_k)$$
$$= \sum_{k=1}^{m} p(\mathcal{U}_k|\mathcal{L}_i)\cdot 1 = 1$$

Therefore $\mathbf{P}$ is row **stochastic**. Similarly we can also prove that $\mathbf{P}'$ is row **stochastic**.

In summary, $\mathbf{P}$ and $\mathbf{P}'$ are **irreducible**, **positive** and row **stochastic** matrices. According to the Markov chain theorem, both the Unified and ST-Unified model will converge using the power iteration algorithm. □

E. ALGORITHM OF THE UNIFIED MODEL

The algorithm of the unified model is shown in Algorithm 1.

Algorithm 1 UnifiedModel ($\mathcal{S}_\mathcal{H}$, m, n)

Input: $\mathcal{S}_\mathcal{H}$, location histories of all users, m, the number of users, n, the number of locations
Output: w_{loc}, the column vector containing ranking scores of all locations

```
M[m][n] ← NewMatrix()
N[m][n][n] ← NewMatrix()
P[n][n] ← NewMatrix()
initialize all the elements in w_loc as 1/n
initialize all the elements in M[m][n] and N[m][n][n] as zero
for each user's location history H_k in S_H do
    if the ith location is visited by the kth user then
        M[k][i] ← M[k][i] + 1
        if the trip is from the jth location then
            N[k][j][i] ← N[k][j][i] + 1
BuildMatrix(P, M, N)
do power iteration on P until w_loc reaches the stationary state.
return w_loc
```

F. REDUCTION OF THE UNIFIED MODEL

The unified model can be reduced to each of the two models introduced in Sections 3.3.3 and 3.3.4.

Reduction to random walk on $\mathbb{G}_{UL}$: If we disregard the location-location reinforcement, we have $p(\mathcal{L}_i|\mathcal{L}_j,\mathcal{U}_k) = p(\mathcal{L}_i|\mathcal{U}_k)$. We can estimate the element $p(\mathcal{U}_k|\mathcal{L}_i)$ in $\mathbf{N}_{UL}$ and the element $p(\mathcal{L}_i|\mathcal{U}_k)$ in $\mathbf{N_{LU}}$ as:

$$p(\mathcal{U}_k|\mathcal{L}_i) = \epsilon\frac{Num(\mathcal{U}_k,\mathcal{L}_i)}{Num(\mathcal{L}_i)} + (1-\epsilon)\frac{1}{m}$$
$$p(\mathcal{L}_i|\mathcal{U}_k) = \epsilon\frac{Num(\mathcal{U}_k,\mathcal{L}_i)}{Num(\mathcal{U}_k)} + (1-\epsilon)\frac{1}{n}$$

Under this estimation, we can see that:

$$\mathbf{N_{LU}} = \epsilon\mathbf{M}^{\mathbf{T}}_{\mathbf{row}} + (1-\epsilon)\mathbf{E_1}$$
$$\mathbf{N}_{UL} = \epsilon\mathbf{M}_{\mathbf{col}} + (1-\epsilon)\mathbf{E_2}$$

Therefore, Equation 4 is exactly the same as Equation 2.

Reduction to random walk on $\mathbb{G}_{LL}$: If we disregard the user-location reinforcement, we treat all the users' trajectories as one pseudo "user." Now we have:

$$p(\mathcal{L}_i|\mathcal{L}_j,\mathcal{U}_k) = p(\mathcal{L}_i|\mathcal{L}_j) = \alpha\frac{Num(\mathcal{L}_j,\mathcal{L}_i)}{Num(\mathcal{L}_i)} + (1-\alpha)\frac{1}{n}$$

Notice that $p(\mathcal{L}_i|\mathcal{L}_j)$ is the element of $\mathbf{P}$. Hence

$$\mathbf{P} = \alpha\mathbf{C}_{\mathbf{row}} + (1-\alpha)\mathbf{E}$$

We can see that Equation 6 is equal to Equation 3. Since we have proved that Equation 6 is simplified from Equation 4, Equation 4 can be reduced to Equation 3.

G. ADDITIONAL EXPERIMENTAL SETTINGS AND RESULTS

G.1 Extracting stays

As explained in Appendix B, we mark two consecutive GPS records as the starts and end of a stay if the time duration between them exceeds Δt=10 minutes. This yields 159,062 stays in our data set. After data cleaning, we obtain 76,139 stay points. These points are located in the region $56^\circ \sim 58^\circ$ North, $9^\circ \sim 11^\circ$ East, which is the region of Nordjylland in Denmark. Figure 2 depicts the distribution of a sample of the stay points.

G.2 Reverse Geocoding and Obtaining Semantics

The Google Maps API is invoked for reverse geocoding. Given a latitude and a longitude, Google Maps returns an addressable location that is nearest to the query position. The returned information contains the street address and the coordinates. An online yellow and white pages directory in Denmark (http://www.degulesider.dk/) is used for finding the semantic of a given street address. The yellow pages service returns a list of semantic locations that are near the query street address and one of which may match exactly the given query street address.

The coordinates used in the raw GPS data are in the UTM (Universal Transverse Mercator coordinate system) format. To perform reverse geocoding, we convert the data from the UTM format to

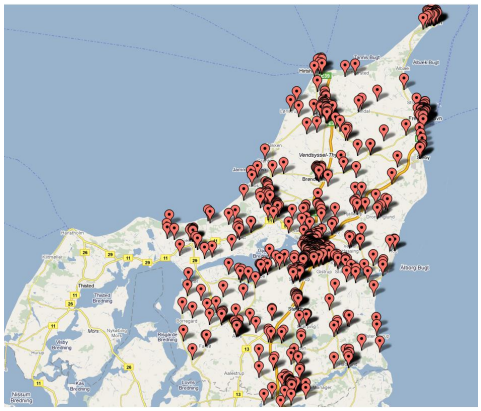


Figure 2: Distribution of stays in our GPS data

latitude and longitude. The method introduced by Salkosuo [2]. is used to perform the conversion. The conversion uses WGS (World Geodetic System) 84 specification.

G.3 Ground Truth of Semantic Locations

We build a ground truth of semantic locations to evaluate the clustering methods. We first reverse geocode each stay point using the Google Maps API. Given a stay point $\mathcal{P}$ with coordinate (x, y), Google Maps API will return the street address of the point together with the coordinate (x', y') of this street address. We call the returned coordinate the reverse-coordinate to distinguish it from the coordinate of the stay point. Note that a street address may correspond to multiple reverse-coordinates, particularly when the returned street address does not contain a street number. We assume that a semantic location corresponds to one or several distinct reverse-coordinates.

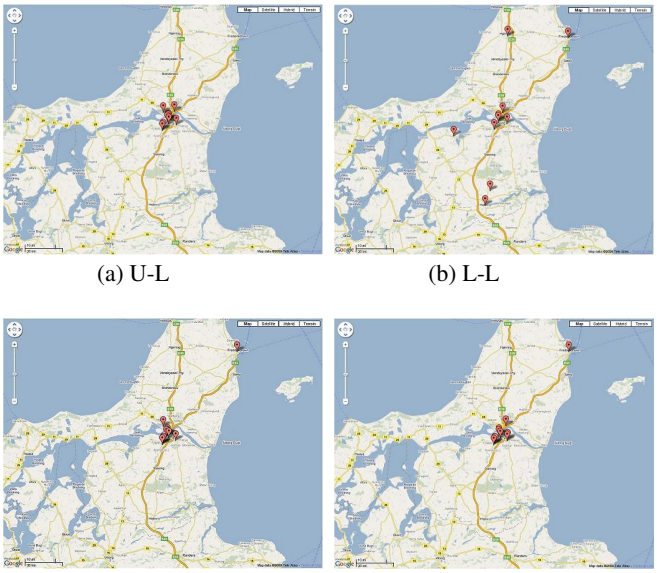
We group stay points such that each group has a distinct reverse-coordinate. We then check whether some groups should be merged to represent a semantic location: We compute the distance of all pairs of groups using their reverse-coordinates; If the distance between two groups is less than 100 meters, we ask annotators to check whether they are the same semantic location and should be merged. Finally, we obtain 7,082 semantic locations.

G.4 Efficiency of Semantic Location Extraction

K-means takes 12 minutes to finish, and OPTICS takes 4 minutes. The enhancement method SEM-CLS takes 390 ms when the number of sampling points is 2, and 420 ms when the number of sampling points is 5 if we assume that all the points have been reverse geocoded by calling external API beforehand. The runtime of reverse geocoding depends on the stability of network, and the availability and workload of external API service. Sometimes we have to call the external API service several times to get the result. Hence, it does not really make sense to report the runtime of the external API service.

G.5 Top-10 Results of Four Ranking Models

Figure 3 shows the top-10 results of the four ranking models proposed in Section 3.3. The harbor location is detected by the L-L model (the top right corner). It does not have a large number of visitors, but has a relative large number of visits. The hospital location (in the middle) is detected by the U-L model. This is because many users have been there, but not too many times. The two unified models detect both locations. The top 10 semantic locations returned by the four methods are listed in Table 8.



(a) U-L (b) L-L

(c) Unified (d) ST-Unified

Figure 3: Top-10 results of the four ranking models

	U-L	L-L	Unified	ST-Unified
1	Bilka	Bilka	Bilka	Bilka
2	Føtex	Stadium	Church	Church
3	Church	Harbor	Stadium	Stadium
4	Cinema	Føtex	Føtex	Føtex
5	Kommune	unknown	Hospital	Zoo
6	Hospital	Church	Cinema	Hospital
7	Stadium	Cinema	Zoo	Cinema
8	Train station	Doense Dybfrost	Harbor	Kommune
9	unknown	Gas station	Bauhaus	Harbor
10	Bauhaus	unknown	unknown	Bauhaus

Table 8: Top-10 results of the four models

Bilka and Føtex are big supermarkets in Aalborg. Bauhaus and Doense Dybfrost are two companies, and their annotation scores are smaller than 1.5.

G.6 Parameter Study for the Ranking Models

There are four parameters in the ST-Unified model, i.e., ϵ, α, ε and η. ST-Unified performs the best when ϵ and α are in the range 0.7–0.9 and ε and η are around the value 0.3.

Parameters ϵ and α are used to control the effects of the constant probability (to teleport to other nodes). Parameter ε is to control the importance of the stay duration at a location, and parameter η controls the importance of the distance between two locations. They affect the effectiveness of the ST-Unified model.

Figure 4 shows the effect of varying ϵ. As we can see, the best ranking performance occurs when the value of ϵ is in the range 0.7–0.9. By setting ϵ to zero, we treat all the users equally, and thus ST-Unified works like the L-L model; by setting ϵ to 1, we disregard the teleport probability in the random walk on $\mathbb{G}_{UL}$. At both extremes, the performance becomes worse.

[2]http://www.ibm.com/developerworks/java/library/j-coordconvert/index.html

Figure 5 shows the effect of varying α, and we can see that the performance is the best when α is in the range 0.7–0.9. When we set α to zero, all the transition probabilities become the same, and ST-Unified works like the U-L model; and when setting α to 1, we disregard the teleport probability in the random walk on $\mathbb{G}_{LL}$.

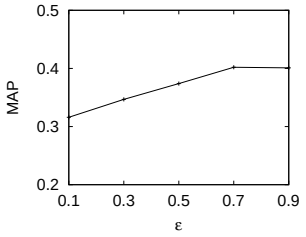


Figure 4: MAP by varying ϵ

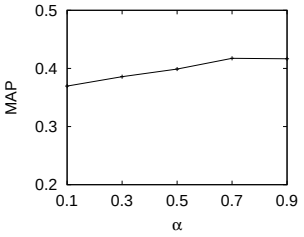


Figure 5: MAP by varying α

Figure 6 and Figure 7 shows the effect of varying η and varying ε, respectively. We can see that only if we combine the consideration of the number of visits, the durations and the distances will we achieve the best results.

When η and ε are set as zero, ST-Unified will become Unified. When ε is set to a large value, the duration plays a more important role than the visiting times in estimating the conditional probability $p(\mathcal{L}_i|\mathcal{U}_k)$. We can see that the performance is slightly worse. When η is set at a large value, the distance is more important for estimating $p(\mathcal{L}_j|\mathcal{L}_i,\mathcal{U}_k)$. This leads to very poor performance because the distance alone is not a good indicator of the transition probability of a random surfer between two locations.

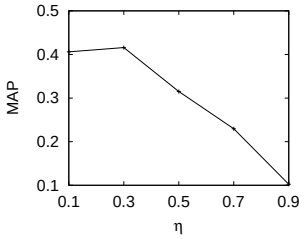


Figure 6: MAP by varying η

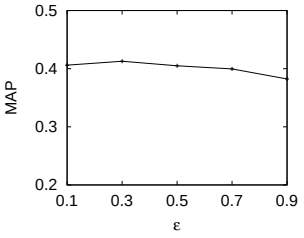


Figure 7: MAP by varying ε

H. ADDITIONAL RELATED WORK

Extracting locations from GPS trajectories: There has been a host of studies on extracting locations from single users' GPS trajectories [3, 8, 10, 13, 20, 24]. There is also recent work [23] on extracting locations from multiple users' trajectories. Some works [8, 10, 20] do not consider the re-occurrence of GPS readings at the same location, and thus they do not need to cluster the GPS points into locations; other works make use of clustering to group GPS records to get locations [3, 23, 24]. However, no works consider the semantics of locations when extracting locations.

Liu et al. [14] consider the extraction of semantic locations, but the study is based on a single user's data and does not cluster similar points. In contrast, our work handles multiple users' trajectories, and a semantics enhanced clustering method is used for extracting semantic locations.

Mining important locations: Zhou et al. [24] mine personal important locations from a single user's trajectories. The importance is a personal view and is "defined" by the user, such as the *home* or *work office* of the user. They do not rank locations, but only classify locations as important or not. In contrast, we rank locations based on location histories of multiple users.

Zheng et al. [23] mine interesting locations and travel sequences from GPS data, and a HITS-based inference model is used for ranking locations. Our work differs from that work in that 1) we consider semantics and use semantic to enhance the location extraction process; 2) we propose new models for ranking locations, which are capable of better exploiting the features of GPS trajectories.

Several locations recommender systems [19, 22] can recommend locations to users based on real-world data. Our proposed techniques can also serve as a location recommender system.

PageRank and HITS. These are two popular link based ranking algorithms that were originally developed for web link analysis. Both methods rank pages according to their importance and authority, estimated by the importance of pages pointing to them. They can be understood as as a Markov chain in which the states are pages and the transition probabilities are determined by the links between pages; the PageRank problem actually amounts to solving an old problem (computing the stationary vector of a Markov chain) in the context of web links [12].

We note that Ding et al. [7] attempt to find a framework to unify the PageRank and HITS models. The unification is to establish a connection between the two models such that it becomes possible to use a simple count of the number of inlinks to a webpage as an approximation of its PageRank. However, the unified model works on a single-layered web link graph, not the two-layered graph in our problem.