Frequent Trajectory Mining on GPS Data

Norma Saiph Savage
Computer Science
UCSB
USA
saiph@cs.ucsb.edu

Shoji Nishimura
NEC Corporation
Japan
s-nishimura@bk.jp.nec.com

Norma Elva Chavez
Computer Engineering
UNAM
Mexico
normaelva@gmail.com

Xifeng Yan
Computer Science
UCSB
USA
xyan@cs.ucsb.edu

ABSTRACT

In this paper we propose a new algorithm for finding the frequent routes that a user has in his daily routine, in our method we build a grid in which we map each of the GPS data points that belong to a certain sequence. (We consider that each sequence conforms a route) we then carry out an interpolation procedure that has a probabilistic basis and find a more precise description of the user’s trajectory. For each trajectory we find the edges that were crossed, with the crossed edges we create a histogram in which the bins denote the crossed edges and the frequency value the number of times that edge was crossed for a certain user. We then select the K most frequent edges and combine them to create a list of the most frequent paths that a user has. We compared our results with the algorithm that was proposed in Adaptive learning of semantic locations and routes [6] to find frequent routes of a user, and found that our implementation on the contrary of [6] can discriminate directions, ie routes that go from A to B and routes that go from B to A are taken as different. Furthermore our implementation also permits the analysis of subsections of the routes, something that to our knowledge had not been carried out in previous related work.

Keywords
Frequent user paths, GPS tracking.

1. INTRODUCTION

The advances in positioning techniques like GPS and Wi-Fi have allowed for users to collect a sequence of time-stamped locations representing their daily trajectories. For many these trajectories represent personal interests and inclinations. In the work done in GeoLife: Managing and Understanding Your Past Life over Maps, by sharing personal trajectories online, users also shared life experiences. Trajectories allow us to comprehend users and locations, they give insight as to how people move about in their city. In this paper, we are interested in finding a user’s frequent trajectories. We believe this information could be highly beneficial as it could enhance mobile advertisement, for example a certain restaurant may wish to give discounts to people that passed regularly by their business. It could also improve public transportation systems, because if it is observed that many users shared a frequent path, a new bus route could be created to accommodate them.

We propose a novel algorithm that by taking into account spatial and temporal attributes can mine frequent trajectory patterns in a probabilistic manner.

From the very beginning we were reluctant in utilizing a coded map of the transportation network to match the user’s GPS data and with it identify the routes which were taken, mainly because methods that use map-matching algorithms have shown in occasions to completely ignore the correlation between subsequent points and are not very efficient in finding personalized daily routes [3]. Furthermore because a map will always be a simplified representation of the real traffic network, there can exist several missing roads. In
Mining GPS Data to Augment Road Models [4] this problem was addressed, they sought to refine an existing map by using GPS traces. These methods focus on creating much more accurate maps, but it is important to note that this does not assure a perfect map matching. It is actually much more difficult to determine the correct linkage in a high scale network [6]. We were therefore more interested in the work that had built road maps without any prior road map. For example in Mining GPS Traces for Map Refinement [5], from a blank map the centerlines are constructed though clustering and then the arrangement of lines and intersections are determined. In the recent work done in Detecting Road Intersections from GPS Traces [1], intersections are found through a detector trained on ground truth data. These techniques aided us for developing our algorithm, but we deferred from them a bit, because we were interested in not only building a road map but also in detecting the frequent trajectories of a user.

The paper is presented as follows, in the next section we explain our approach and our method for finding frequent trajectory patterns, we describe the data set we used for experimentation and present our results. We then briefly explain the method proposed in [6], we show the results of using their algorithm over our data set and then give a comparison between these two algorithms. Finally, we present the conclusions of this research as well as our future work.

2. METHODS

For our algorithm we consider that we have sequences of GPS data, which are sampled at different positions. From these GPS data sequences, we intend to find the frequent paths that a user takes on a daily basis. This is not a simple task to solve since even when one is in a fixed location, it is very difficult to obtain two same readings from the GPS, therefore although two GPS sequences could come from the same source, that in both cases traveled through the same route, the two sequences will rarely present the exact same values, this is because GPS data tends to be very fuzzy, there frequently exists a measurement error, furthermore GPS readings are not available under all conditions. This provokes that we may have data points that were recorded in the log, that are too far apart from each other and it is thus a bit ambiguous to know the true path that the user followed. To solve this problem, we propose to create a grid in which GPS data points are drawn in. This grid resembles a bit the work done by Hägerstrand, in which he mapped the physical area that existed around a person to a two-dimensional plane, where the person’s current location and destination are represented as zero-dimensional points and time is represented by the vertical axis. The interval of the edges of the grid is a design parameter which can be modified freely (In our work, this value was calculated from the average speed that the user presented). The larger the edge interval, the more points that are allowed to fall into a cell and the similarity between these points is increased. Once the grid has been constructed we map each GPS sequence into it, for each two points of a sequence, we calculate the possible trajectories that could exist between them. Figure 1 presents a visualization of this idea. Since evidently an infinite number of different trajectories can take place between two points, we limit our search to an ellipse area, in which the major semi-axis of the ellipse is constructed from the straight line that is drawn between these two points. The minor semi-axis of the ellipse is constructed by obtaining the midpoint of the major semi-axis and creating a vertical line that passes through this point and has a height $h$ (This $h$ value is another design parameter which we can modify to obtain diverse results). Once we have created the ellipse we select the number of possible trajectories that are confined inside of it. This is the third design parameter of our algorithm. All of these trajectories have the characteristic, that they have the same starting and ending points and pass through a point that is confined to the minor semi-axis. Figure 2 we can observe how we confine trajectories to an ellipse. For each of these calculated trajectories, we obtain through which edges they crossed, each edge has an assigned crossing probability, this crossing probability is calculated as follows: we consider that a cell has $X$ percentage of chance to be occupied, and has $n$ candidate crossings, we then assign an $X/n$ crossing probability to each edge of the cell. For each generated sequence, we have the edges that belong to it, as well as the probability that each of its edges hold. Each one of those probabilities contributes to the overall support that a certain trajectory holds. The trajectory with the greatest support is the one that is selected. In summary, in this step, between two read GPS points we construct a series of intermediate points and
with this obtain more insight of the movements that might have been carried out in the moments in which the GPS reading was not present. This is an interpolation process which we believe helps to overcome the problems that are presented by occlusion of the GPS readings. We then obtain a series of possible trajectories that could have existed for one GPS sequence. We select the trajectory which has the highest support.

The next step is to compute the histogram of frequent edges. For each sequence, we check the edges that it holds, if one of the edges is an edge that had never been crossed before, then a new bin that represents that edge is created. On the contrary when we have an edge that has previously been observed in the bin that represents that edge we increment its frequency by one. At the end, we have a histogram whose bins represent the crossed edges, and the frequency the number of times an edge has been crossed. We then obtain the $K$ edges with the highest frequency, the $K$ value is a design parameter. With these $K$ edges, we create combinations of sequences that contain all of these edges, these sequences are what we denominate a user’s frequent trajectories. Since we wished to compare our algorithm with what the state of the art has done in this area, in the following, we will explain the nature of the algorithm that was proposed by [6]. They developed a minimax criterion to compare two routes. They suppose two routes of GPS sequences are given by:

$$R_1 = g_1^1; g_1^2; g_1^3; \ldots; g_1^n$$
$$R_2 = g_2^1; g_2^2; g_2^3; \ldots; g_2^n$$

The best match of $R_1$ in terms of $R_2$ is:

$$R_1 = \hat{g}_1^1; \hat{g}_2^1; \hat{g}_3^1; \ldots; \hat{g}_n^1$$

where $\hat{g}_1^1 = \arg\min_{g_1^j} ||g_1^j - \hat{g}_1^1||$ Similarly $R_2 = \hat{g}_1^2; \hat{g}_2^2; \hat{g}_3^2; \ldots; \hat{g}_n^2$ is the best match $R_2$ in terms of $R_1$ where $\hat{g}_2^2 = \arg\min_{g_2^j} ||g_2^j - \hat{g}_2^2||$. They regard two routes as the same if: $\max ||g_1^j - \hat{g}_1^1|| < d_1$ and $\max ||g_2^j - \hat{g}_2^2|| < d_2$. These were the largest sampling intervals of $R_1$ and $R_2$ respectively.

3. DATA SETS

We tried our algorithm over two different data sets. One of them was a personal GPS logging system, that was taken by a member’s car navigation system. This data set had about 42 tracks, where each track has a sequence of GPS data recollected from one car drive. The other data set in which we tried our algorithm on, was the CRAWDAD data set [2], this data set recollected data from over 523 cabs over 1 month in San Fransisco. It had over 10 million GPS points. Before running our algorithm over these data sets, we first prepared and pruned the data. In the case of the cab tracks, it was necessary to identify when a starting and when an ending point of a particular route began. For this, we went over the tracks of each driver and for each track we tried to identify points in which the cab was parked. We assumed that when the cab was parked, it was because a route had just ended. We considered that when the cab started moving from this parking state, it was because a new route was beginning. With this methodology, for each taxi driver track we found a series of routes, that had their starting and ending points well defined. In Figure 3 a visual representation of this is presented. To save computation time when running the algorithm of [6], we grouped routes that had similar starting and ending points.

4. RESULTS

We utilized our personal GPS data set to verify that our algorithm functioned adequately. We established the ground truth as to what the most frequent paths were manually and compared the output of our algorithm with it. We found that it was better not to select the K edges with highest frequency but rather to recollect all of the edges that were above a certain frequency. We noticed that with our data set the best threshold value was 10. We observed that our system was capable of identifying all of the frequent paths we had manually selected, and also provided significant information as to what sub-paths were the most traveled. We carried out this same type of testing over the algorithm proposed by [6] and found that it did not identify all of the frequent routes that we expected to see. A clear problem we observed, is that unique routes cannot be distinguished in terms of their direction i.e., the route from place A to place B is found to be the same as the route from place B to place A. Another problem that we see with this method is that it is only capable of detecting when two trajectories are similar, but we cannot identify the common subsections that the two trajectories present. Their algorithm considers the information of a route as a whole and does not permit for an analysis of the subsections of the route. Subsection analysis can be very important for giving local traffic reports or for local mobile advertisement applications, where shop owners might wish to target users that pass frequently through the street of their business. It could also be utilized by
public transportation systems that wished to create routes which more users benefited from, perhaps a great number of persons passed daily through "Main Street" and their final destination diverted by simply a block or two. Therefore if a bus route that passed through Main Street were to be opened, many of these users could get off at Main Street and simply walk the block to arrive to their desired destination. In all of these cases having an analyze of the whole route might not be as useful, as being able to analyze subsections.

We also tested our algorithm over the CRAWDAD data set. We present here in Figures 4 and 5 the frequent routes that we found for the first taxi driver in our data set, these routes are built from the most frequent crossed edges, we threshold these frequent crossed edges by the "support" they present. We also tested the algorithm of [6] over [2]. Due to space constraints we do not present the visualization of the output of their algorithm.

5. CONCLUSIONS

In this report we presented a new algorithm through which we can identify using a probabilist support mechanism the frequent routes that a person takes on a daily basis. The novelty of our approach is that by creating a histogram of frequent edges we permit for the analysis of subsections of the frequent route. It is to our best knowledge, that in past work, this was something that had not be done. In the near future, we would intend to improve the computational time and strategy to create the combinations of crossing edges. The computation time of our algorithm depends not only on the size of our data set but also on grid size. When we obtain routes that have high detail, the grid size tends to be very small. Therefore, when adjacent measured points are located within a great distance between each other, our algorithm generates many interpolating points, the number of interpolating points is $O(L^2)$ where L is the distance that exists between those two measured points. We need therefore to introduce some cut out mechanisms for this case. One possible method is to clean abnormal data before running our algorithm. Sometimes the data set contains measured points which are located far away from the rest of the measured points. For example, the data set of taxi trajectories in San Francisco contains measured points that are actually located in the middle of the Pacific Ocean! This kind of measurement error is highly likely to occur, we therefore need to eliminate such data points.

Another possible solution is to introduce a hierarchical interpolation mechanisms. GPS devices measure a current location by a constant time interval. However they sometimes fail to measure or give location information because their satellite are occluded or they are turned off. For example, in the data set of taxi trajectories, it records location information approximately every minute but in occasions presents readings that have a time interval over 10 minutes! For these cases, we believe that an adequate approach might be to roughly interpolate a few points between the measured points and then recursively interpolate points between the previously interpolated points and the measured points. We are also considering in extending this approach to mobile phones, and making the adequate modifications to create an online algorithm that given a sequence of GPS data points from a Mobile phone can find the frequent routes of a user. Overall we created an novel algorithm for identifying the frequent routes of a user. We tested our algorithm over two fairly large data sets and found that our algorithm provided a feasible solution for finding these frequent routes. Additionally our algorithm was capable of providing an analysis about the sub routes of a route, which can be important information for a varied number of applications.

6. ACKNOWLEDGMENTS

This research was sponsored in part by the Army Research Laboratory under cooperative agreement W911NF-09-2-0053 (NS-CTA). The views and conclusions contained herein 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 notice herein. Special thanks to CONACYT for providing funding for Saiph’s studies.

7. REFERENCES