Analyzing Travel Patterns for Scheduling in a Dynamic Environment

Sonia Khetarpaul[1], S.K. Gupta[1], and L. Venkata Subramaniam[2]

[1] Department of Computer Science and Engg., Indian Institute of Technology, Delhi, India
{sonia,skg}@cse.iitd.ac.in
[2] IBM Research Lab., Delhi, India
lvsubram@in.ibm.com

Abstract. Scheduling a meeting is a difficult task for people who have overbooked calendars and many constraints. This activity becomes further complex when the meeting is to be scheduled between parties who are situated in geographically distant locations of a city and have varying traveling patterns. We extend the work of previous authors in this domain by incorporating some real life constraints (varying travel patterns, flexible meeting point and considering road network distance). We also generalize the problem by considering variable number of users. The previous work does not consider these dimensions. The search space for optimal meeting point is reduced by considering convex hull of the set of users locations. It can be further pruned by considering other factors, e.g., direction of movement of users. Experiments are performed on a real-world dataset and show that our method is effective in stated conditions.

Keywords: Spatio-temporal data mining, geographical Locations, GPS logs, Meeting Points.

1 Introduction

Recent advances in wireless communication and positioning devices like Global Positioning Systems (GPS) have generated significant interest in the field of analyzing and mining patterns present in spatio-temporal data. The pervasiveness of location-acquisition technologies (GPS, GSM networks, etc.) has enabled convenient logging of location and movement histories of individuals. The increasing availability of large amounts of spatio-temporal data pertaining to the movement of users has given rise to a variety of applications and also the opportunity to discover travel patterns. Managing and understanding the collected location data are two important issues for these applications.

The amount of data generated by such GPS devices is large. For example, most GPS devices collect location information for a user every 2 to 5 seconds [8]. This means that for a single user between 17000 to 44000 data points are generated in a single day. Aggregated over tens of users over several days the data size grows exponentially [8,9]. This is extremely rich data and a lot of useful analysis can be performed on this data, potentially giving rise to a variety of application.

A. Cuzzocrea et al. (Eds.): CD-ARES 2013, LNCS 8127, pp. 304–318, 2013.

The objectives of this paper is to analyze historical data, determine spatio-temporal relationships among users and predict their behavior efficiently and to determine the optimal meeting point for n users on a road network.

Different applications ranging from location-based services to computer games require optimal meeting point (OMP) query as a basic operation. For example, an educational institute may issue this query to decide the location for a institute bus to pick up the students, so that the students can make the least effort to get to the pickup point. This is also true for numerous other scenarios such as an organization that wants to find a place for its members to hold a conference. This can also be helpful for deciding common meeting points, for social networking site users, having common interests. In strategy games, a computer player may need this query as part of the artificial intelligence program, to decide the appropriate routes.

We introduce two measures to evaluate the processing cost i.e. the minimum-sum-center and the direction of movement. These measures operate over the spatio-temporal domain of each moving objects by applying a network distance to all objects tracked. Each measure induces a spatio-temporal relation that minimizes or maximizes a property over the underlying network graph for the given measure and the given set of moving users. We develop query processing algorithms for computing the value of these measures and to determine spatio-temporal relations and the point on the road network that yields the optimal value of relation's value from the predictive graph of moving objects. Finally, we demonstrate how object movement histories and projected movement trajectories can be used to determine the optimal meeting point.

1.1 Problem Statement

The problem statement can be stated in terms of input and output as follows:

- **Input :** Given GPS logs of n users, proposed meeting time, road network
- **Output:** An Optimal Meeting Point Location(Latitude,Longitude) for given n users

We assume that users have travelling patterns which can be discovered from logs of their travel history. These logs are generated by standard GPS devices. We assume that these devices generates traces at the same rate and in same format(same granularity) or GPS traces can be transformed to a fixed format. We define the optimal meeting point for group of people as:

Let $\{u_1, u_2, ..., u_n\}$ be a set of n users. The users have a definite travel pattern(may be periodical) which is hidden in their logs. Let $u_i(t)$ denotes the location of user i at time instant t. It is noted that if all of these n users wish to meet at some fixed time, and the meeting point is desired as OMP, then the sum of distances would be the total cost of meeting(for all the users). It is proposed to minimize this cost.

Optimal meeting point is defined to be $arg\ min_{OMP \in N}[\Sigma d_N(u_i(t), OMP)]$, where $d_N(x, y)$ is the shortest distance between two point x and y on the road network N.

Informally, we define the **optimal meeting point** as a point on the road network where the sum of distances travelled by all the users is minimum. In Figure 1 total distances travelled by all users is minimum at point OMP, and is 23 K.M.

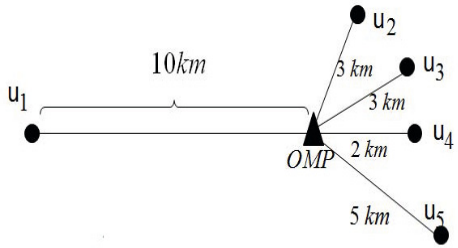

Fig. 1. Optimal Meeting Place

We also consider predicted directions of motion of users at time t and at time $t + \delta t$ to optimize the total distance covered by each user. It is considered by examining the consecutive hulls of of location points.

The rest of this paper is organized as follows. In the following subsection we describe related work and also put the work of this paper in the context of related work. Later in section 2, we introduce algorithms, describing the underlying idea to determine the optimal meeting point. In Section 3, we report the results of our experiments on real-world dataset to show the feasibility of our algorithms for the OMP queries. Finally, we conclude our paper in Section 4.

1.2 Related Work

There has been a lot of prior work on using spatio-temporal data to track movement history. There has also been work around integrating multiple users' information to learn patterns and understand a geographical region. GeoLife is a location-based social-networking service on Microsoft Virtual Earth. GeoLife enables users to share travel experiences using GPS trajectories [3, 6, 8, 9, 14]. It finds the top most interesting locations, classical travel sequences in a given geospatial region. In geolife to find out interesting locations in a given geospatial region HITS model (Hypertext induced topic model search) is introduced. The paper [7] is based on Hybrid Prediction Model, which estimates an object's future locations based on its pattern information as well as existing motion functions using the object's recent movements. Other systems like and CityVoyager [10] are designed to recommend shops and restaurants by analyzing multiple users' real-world location history. Mobile tourist guide systems [6, 10–12] typically recommend locations and sometimes provide navigation information based on a user's real-time location. In contrast, our approach is based on assumption that users are moving not stationary, and follows a regular routine during weekdays. Our approach to determine users location points is based on simple statistics that applied on users historical GPS traces.

OMP problem is well studied in different forms in Euclidean space. In Euclidean space optimal meeting point is basically the Geometric median of location point set. When the Euclidean distance is adopted as the metric of distance, the OMP query is called the Weber problem [17], and the OMP is called the geometric median of the query point. Like various nearest neighbor queries [5, 13, 15, 16], the OMP query is also fundamental in spatial databases.

In our previous work [1] we have determined common meeting point from spatio-temporal graph analysis for two users in the Euclidean space. In this paper we determine the optimal meeting point from the trajectory analysis for n users on road network. This brings in several factors in the analysis, making the system flexible and more realistic to use, while adding complexity to analysis.

On the other hand, the OMP query is not well explored in terms of road networks, where the network distance is adopted as the distance metric. However, compared with the Weber problem, this is a more realistic scenario for location-based services. Recently, [4] proposed a solution to this problem by checking all the split points on the road network. It is proved in [4] that an OMP must exist among the split points, which leads to an algorithm that checks the split point of each query point in Q on each edge in the road network G = (V, E), and picks the split point with the smallest sum of network distances as the OMP. As a result, the search space is $|Q|.|E|$, which is huge. Although [4] includes a pruning technique to skip some split points that are guaranteed not to be an OMP, the search space after pruning is still very large. Therefore, a novel road network partitioning scheme is proposed in [4] to further prune the search space, based on the property that the OMP is strictly confined within the partition where all the objects in the query set Q are located. After that [2] proves that an OMP must exist either on vertex or on query points, there is no need to check all the split points. So search space is reduced to $|Q| + |V|$. To further reduce the search space two phase convex-hull-based search space pruning techniques are proposed in [2]. In contrast our approach considers that user are moving not stationary, so we are predicting user locations, the directions in which they are moving and pruning the search space based on their locations before and after the meeting. Our approach considers both spatial and temporal aspect of data. Our approach is defined in following section.

2 Determining Optimal Meeting Points for Multiple Users

GPS and other positioning devices generate location information every few seconds (often at the interval of two to five seconds). An individual carrying such a device potentially generates thousands of GPS points everyday. It is important to be able to aggregate all the data from multiple users and predict their location points at the given time. In this paper we will apply statistical operation to generate spatio-temporal location point prediction for each user at given time. After predicting the location points for the user optimal meeting point is determined.

We will now define the notation and also describe the problem that we are solving. We assume that there are n users whose time stamped GPS logs are available to us.

GPS Users: We have GPS logs of n users $U = \{u_1, u_2, ..., u_n\}$.

GPS Point: A GPS point g_i is a four field tuple, $\langle x, y, d, t\rangle$ where x, y are geographic coordinates (Latitude, Longitude respectively) and d, t is the timestamp (date, time respectively) represents a user's location at any point of time.

The pair (Longitude, Latitude) represents the position of the user at a particular point of time. For the purposes of standardization, we assume that each value in the pair is given in six decimal places. A value in decimal degrees to 6 decimal places is accurate to 0.111 meter at the equator [20].

GPS Trajectory: A GPS trajectory Tr_i is a sequence of GPS traces ordered by timestamp $Tr_i = (g_{i_1}, g_{i_2} ....., g_{i_n})$ of the user i.

We divide our region of interest into a grid C of $m * n$ cells where m is number of unique δLat and n is number of unique δLng.

Cell: A cell c_i is a rectangular element of grid C dividing the region of interest. They are sequenced major rowwise and minor columnwise.

Road Network Distance: Road network distance is the shortest length of path between two cells on road network. This distance is obtained using function $d_N(c_i, c_j)$, it returns the length of path between two cells of grid C, c_i and c_j on road network. This distance function $d_N(c_i, c_j)$ can be realized through google maps.

Location Point: A location point l_i represents the location of the i^{th} user on grid C at a point of time.

We determine the location points of all individual users at given point of time. Location point of a user is a predicted geographical location where the user is, at given meeting time. This prediction is done by statistical analysis of their past GPS logs.

2.1 Location Point Determination

Most of us generally follow a specific travel pattern during working days. To determine the location point for each individual user, at a given point of time, we analyze their past GPS logs. By applying statistical operations on their past GPS trajectories, we are able to predict their locations at a given point of time. For location points analysis w.r.t. to time and space, the whole geographical space is divided into grid, where each cell $c(l * w)$ represents a small geographical region and is assigned a number. Twenty four hours in a day are divided into small time periods of length δt. The log records are mapped on to this grid. For each user, his/her location cell number, after every δt interval of time is identified from his/her log records. User locations of many days at different time intervals are summarized to generate his/her spatio-temporal graph.

We determine the maximum and minimum value of latitude and longitude($Ar = (maxlat - minlat) * (maxlng - minlng)$ gives total area of city), which define our domain of interest. $\delta lat * \delta lng$ form the area of a single cell within the grid. They are sequenced major rowwise and minor columnwise, from 1 to K, where K maximum number of cells. Location of each user is predicted in terms of a cell number. Mapping of user's GPS location into cell number and cell number into GPS location is done using following conversions.

Mapping given location (Lat_i, Lng_i) to cell number:
$Cellno = ((Lng_i - minLng)/\delta Lng) *$ No. of unique $\delta clng + ((Lat_i - minLat)/\delta Lat)$

Mapping given cell number to location (Lat_i, Lng_i):
$Cellno = lng_{ind} *$ No. of unique $\delta lng + lat_{ind}$

Where lng_{ind} is quotient and lat_{ind} is remainder when $cellno/$ (No of unique δlng)
$Lat_i = minLat + lat_{ind} * \delta Lat$
$Lng_i = minLng + lng_{ind} * \delta Lng$

Using these mapping we are able to plot user historical traces onto grid after every δt time interval. After plotting the user traces on the grid we apply statistical mode operation (defined below) to determine the user location at given time t.

Temporal Mode of User Location Points
The temporal mode of a set of data points is the value in the set that occurs most often during a specified time interval. The mode of i^{th} user's historical data points $Mode(U_i)$ is the value that occur more frequent within a specified the time. For applying the mode operation, users longitude and latitude values are mapped onto the grid so that it points to the defined geographical region. The mode operation is applied for each user.

Let U_i be the set of GPS points of user i at time δt and l_i be the most frequently visited location point of user for given time δt.

$l_1 = Mode(U_1)$
$l_2 = Mode(U_2)$
$l_n = Mode(U_n)$
$L = \{l_1, l_2, l_3, ..., l_n\}$

For a set of users, we can determine the cell numbers(location points) in which they are expected to be at any point of time. Let the set of location points of n users is denoted with $L = \{l_1, l_2, ..., l_n\}$ at time t.

By applying these statistics we determine the cell where an individual user is mostly present at a given point of time. We discard the cells which are visited very few number of times in long period of GPS traces.

A baseline algorithm to solve the meeting point problem is defined below.

2.2 Baseline Algorithm

For a given set of users, let L be union of set of location points. The baseline algorithm considers all the cells $|C|$ within the grid as the probable candidates for an optimal meeting point. The baseline algorithm evaluates the sum of distances from the location points of each user to each cell and the OMP is the cell with the minimum value of the sum.

The approach is presented in Algorithm 1 where function d_N computes shortest distance between a location point l_i and cell c_j on road network using any standard procedure (in our case Google Maps API).

The baseline algorithm has very high computational complexity $O(nK)$, where n number of users and K is the number of cells, as it considers the entire grid as the search space. It is required to prune the space to overcome this problem. We propose the use of a two level convex hull pruning to reduce the search space, and there by improving the efficiency.

Algorithm 1. BaseLine Algorithm(L,C)

Data: Location points $L = \{l_1, l_2, \ldots, l_n\}$ of n users at given time T_i on the Grid containing $|C|$ cells
Result: Optimal Meeting Place- A cell on the Grid $OMP(Lat, Lng)$
begin
  $OMP \leftarrow NULL$
  $mincost \leftarrow +\infty$
  foreach $c_i \in C$ **do**
    $sum \leftarrow 0$
    foreach $l_j \in L$ **do**
      $sum \leftarrow sum + d_N(c_i, l_j)$
    $cost \leftarrow sum$
    if $cost < minCost$ **then**
      $mincost \leftarrow cost$
      $OMP \leftarrow c_i$
  $Return(OMP)$

So baseline approach is to consider all the cells on the grid as a search space to determine the optimal meeting point. In our approach we are using two level convex hull pruning to reduce the search space and to improve the efficiency of search.

2.3 Convex Hull Based Pruning

The convex hull $H(L)$ of a set L is the intersection of all convex sets of which L is a subset. It is also the union of all straight lines joining all pairs of points in L [19].

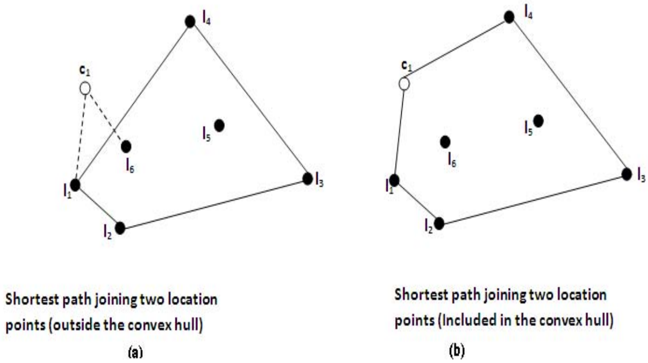

Fig. 2. Counter example

It can be observed that given a set of location points L, a minimum distance point from all location points of set L, i.e., $argmin_{x'}[\Sigma d_E(l_i, x')]$ always lies inside the convex hull H(L), where function $d_E(x, y)$ returns Euclidean distance between points x and y. It can be deduced from the property of a convex object that its centroid lies within the object [19]. But, as shown in figure 2(a), it may not be always true for road network.

To ensure this property for road network, we calculate the shortest route between every two location points using function $shortRoute(x, y)$, all the points that lies among the routes are merged with location points set as described in figure 2(b). After that we take the convex hull of this set.

According to the baseline algorithm, OMP must exist among one of cells in the Grid. It is not necessary to check all the cells in the grid. The search space can further be pruned. We check only those cells that are in the smallest partitioned grid enclosing all cells of the user's location points. We define a convex hull based pruning technique in Algorithm 2, where $convexHull(L)$ computes the convex hull of the point set L using Andrew's Monotone Chain algorithm [18] and takes $O(|P|log|P|)$ time where P is the number of points. All the cells those lie within the hull, we collect them into set P. Now to determine the OMP we check only those that are belong to set P.

Algorithm 2. ConvexHullPruning(LocationPoints,Cells)

Data: Location points $L = \{l_1, l_2, \ldots, l_n\}$ of n users at given time T_i on the Grid consists of cells C
Result: Set of Grid Cells P lying inside the Convex Hull
begin
  $OCells \leftarrow 0$
  foreach $l_i \in L$ **do**
    foreach $l_k \in L$ **do**
      $OCells \leftarrow shortRoute(l_i, l_k)$
  $L_n \leftarrow L \cup OCells$
  $P \leftarrow 0$
  $H \leftarrow ConvexHull(L_n)$
  foreach $c_i \in C$ **do**
    if $c_i \in H$ **then**
      $P \leftarrow P \cup c_i$
    else
      Discard c_i
  $Return(P)$

The search space is significantly reduced using convex hull based pruning. It is further possible to trace the direction of movement from the user data. We can further reduce the search space by considering the direction of movement of the Users'.

2.4 Direction of Movement Based Pruning

Let $L(t) = \{l_1, l_2, \ldots, l_n\}$ be the set of location points of users at time t and let P be the set of cells of the grid lying within the convex hull of $L(t)$. Similarly, let $L'(t+\delta t) = \{l'_1, l'_2, \ldots, l'_n\}$ be the set of location points at time $t + \delta t$ and P' be the set of cells of the grid lying within the convex hull of $L'(t + \delta t)$. By analyzing the two consecutive convex hulls, it is found that two cases are possible.

Case 1: The two convex hulls are intersect or overlap.

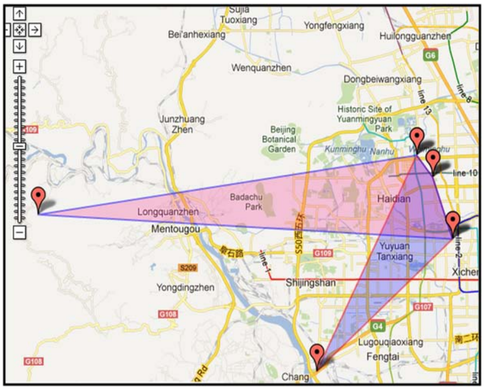

Fig. 3. Intersecting Hulls

In this case, the meeting point among the cells is assumed to lie inside the intersection/overlapped region. Figure 3 depicts the intersecting and figure 4 depicts the overlapped convex hulls of four users. It may be noted that this case also covers the case if a convex hull is completely contained in the other convex hull shown in figure 5. In this case, we assume that the meeting point would lie in the smaller convex hull. We take meeting point inside the intersection because it reduces the sum of total distance travelled by users before and after the meeting.

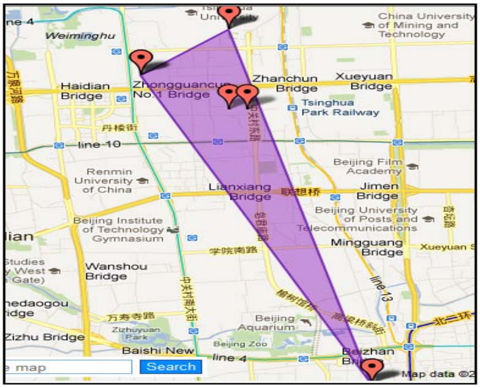

Fig. 4. Overlapped Hulls

Case 2: The two convex hulls are disjoint of each other (i.e. they have zero intersection). In this case, we assume that the meeting point would lie in the first convex hull. We take meeting point inside the intersection because it reduces the sum of total distance travelled by users to reach the meeting point.

These two levels of convex hull pruning prune the search space to P'', a considerably reduced set. A complete algorithm 4 is developed to determine optimal meeting point in such a scenario. Function Map(OMP,OMPLoc) converts cell designated as OMP into (Latitude,Longitude) pair represented by OMPLoc.

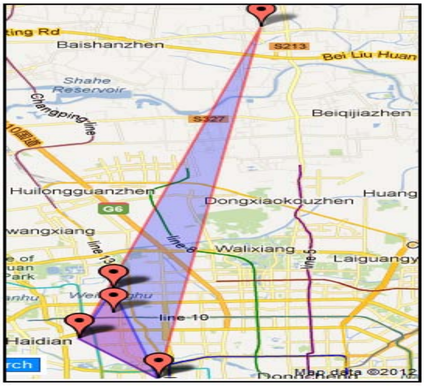

Fig. 5. Inner Hull

Algorithm 3. DirectionPruning(LocationPoints,Cells)

Data: Set of Grid Cells P at given time T_i and P' at time $T_i + \delta t$
Result: Set of Grid Cells P'' selected after pruning
begin
  $P'' \leftarrow P \cap P'$
  if $P'' isNull$ **then**
    $P'' \leftarrow P$
  $Return(P'')$

3 Experiments

In this Section, we first present details about the GPS dataset used [6, 8, 9]. Next we present the results of applying statistical operations on the temporal aspect of GPS data for predicting user location points at a given time. Then, as stated, two level of pruning are used to determine OMP.

We compare the results of our approach with two other approaches in terms of total distance travelled by users before and after the meeting and number of cell searched. First approach is when we do not consider the direction of movement of users and second approach is when we consider all location points before and after the meeting i.e. L and L' simultaneously.

3.1 GPS Trajectory Dataset

The GPS trajectory dataset [6, 8, 9] is a repository of real life data collected by Microsoft Research. The data was collected by 165 users in a period of over two years (from April 2007 to August 2009). A GPS trajectory of this dataset is represented by a sequence of time-stamped points, each of which contains the information of latitude, longitude, height, speed and heading direction, etc. These trajectories were recorded by different GPS loggers or GPS-phones, and have a variety of sampling rates.

Algorithm 4. OMP Algorithm(L,L',C)

Data: Location points $L = \{l_1, l_2, \ldots, l_n\}$ and $L' = \{l'_1, l'_2, \ldots, l'_n\}$ of n users at given time T_i and $T_i + \delta T_i$ on the Grid G containing $|C|$ cells
Result: Optimal Meeting Place- A cell on the Grid $OMPLoc(Lat, Lng)$
begin
  $P \leftarrow ConvexHullPruning(L, C)$
  $P' \leftarrow ConvexHullPruning(L', C)$
  $P'' \leftarrow DirectionPruning(P, P')$
  $mincost \leftarrow +\infty$
  foreach $c_i \in P''$ **do**
    $sum \leftarrow 0$
    foreach $l_j \in P''$ **do**
      $sum \leftarrow sum + d_N(c_i, l_j)$
    $cost \leftarrow sum$
    if $cost < minCost$ **then**
      $mincost \leftarrow cost$
      $OMP \leftarrow c_i$
  $Map(OMP, OMPLoc)$
  $Return(OMPLoc)$

95 percent of the trajectories are logged in a dense representation, e.g., every $2 \sim 5$ seconds or every $5 \sim 10$ meters per point, while a few of them do not have such a high density being constrained by the devices. This dataset recorded a broad range of users' outdoor movements at different time intervals over a day, including not only life routines like going home and going to work but also some entertainment and sports activities, such as shopping, sightseeing, dining, hiking, and cycling etc. In the data collection program, a portion of users carried a GPS logger for more than two years, while some of them may have carried a logger for few weeks.

We worked on 126 users from the above dataset and worked on their GPS traces. This subset consists of a total of 68612 days data with 5,832,020 GPS points. The total area covered by the GPS logs exceeded 3,880,951 Sq. kilometers. The majority of the data was created in Beijing, China. A large part also came from Hangzhou. We have partitioned the total area covered by users into grid. Following subsections explain the grid formation for further processing.

3.2 Determining Grid

The major portion of this dataset belongs to Beijing, China. A grid is defined over this area, which is approximately 38972.068 Sq. kilometers, with minLat=38.0, minLng =115.0 and maxLat=40.0, maxLng=117.0. This is further divided into small cells of (222 meter * 219 meter) area with δLat=0.002 and δLng=0.025. Thus we have a total of 1,80,00,00 cells. These cells are sequenced major rowwise, minor columnwise and identified by unique numbers from 1 to 1800K. Users GPS points are mapped into these cells and their location points are predicted.

After partitioning the city into the grid, the aim is to compute the user locations at a given point of time and map them onto grid.

3.3 Predicting User Location Points

Recent one month historical data of a user is analyzed to predict his/her location at a given time. For example if we have to predict the user location for 18 April at 10 am. then we extract his last month data points for the time interval 9:45 to 10:15 am. Data points are mapped on to grid and temporal mode operation is applied to determine the cell with maximum frequency of data points. This cell is marked as location point for the user at 10 am.

This experiment is performed for hundreds times and prediction is made for days for which data is already available. It is observed that 73% of times predicted cell is same as actual location of user. For all prediction made a **mean square error of 0.238** was determined. Figure 6 shows a mean square error detected for 104 prediction made.

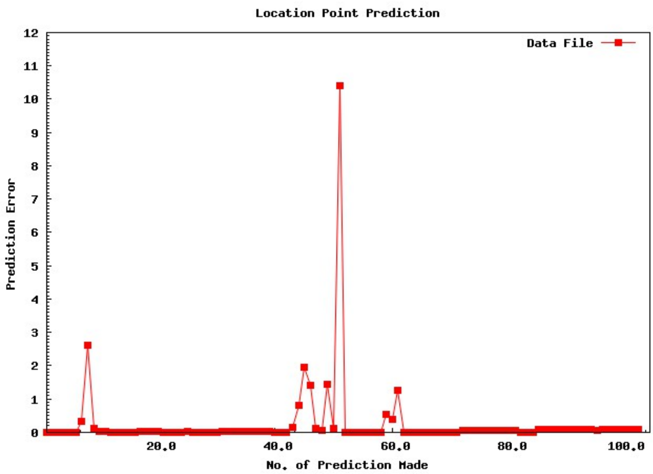

Fig. 6. Prediction Error

After computing users locations our aim is to determine a meeting point for them on road network.

3.4 Determining Optimal Meeting Place

To check the efficiency and accuracy of our approach, we conducted experiments to determine optimal meeting point in three different ways, described below.

OMP by Considering set L(No Directions). In this case, we determine the meeting point, for a given meeting time, by considering only set L. For example, if the meeting time is at 10 am, then search space is pruned by considering users' location at 10 am only. A convex hull of current location points L is calculated and the optimal meeting point is determined within this convex hull.

OMP by Considering Set L and L' Simultaneously. In this case, we determine the meeting point, for a given meeting time, by considering both set L and L' at the same time. For example, if meeting time is at 10 am and meeting duration is one hour, then we prune the search space by using both sets of user locations at 10 am and at 11 am simultaneously. Convex hull of all location points is calculated and the optimal meeting point is determined within this convex hull.

OMP by Considering Direction of Movement of Users with Time. In this case, we consider the direction of movement of users to determine the meeting point. For example, if meeting time is 10 am and meeting duration is one hour, then we prune the search space by considering users' location points at 10 am and at 11 am. We first determine the convex hull of locations points at 10 am and then we determine the convex hull of user locations at 11 am. An optimal meeting point is determined within the intersecting/overlapped area of two convex hull.

A table 1 summarizes the average distance travelled by all users and average number of cells to be searched in each of these three cases is given below.

Table 1. Distance Travel and Cells to be searched in 3 different Cases

Without Directions		All Location Points		With Directions	
$Avg.Distance_1$	$Avg.Cells_1$	$Avg.Distance_2$	$Avg.Cells_2$	$Avg.Distance_3$	$Avg.Cells_3$
35.032	454.925	31.00505	407.375	31.94225	74.125

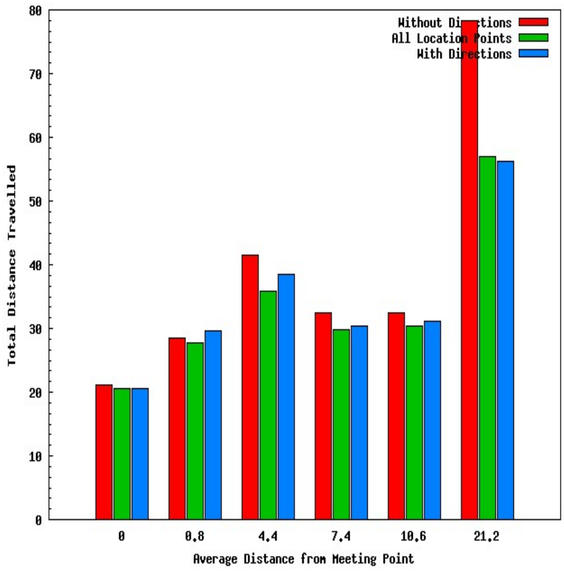

Fig. 7. Distance Travelled by users

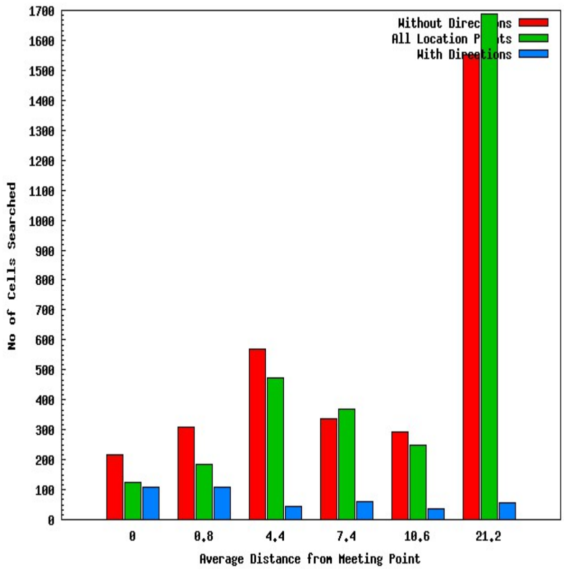

Fig. 8. Search Space reduction

Experiment is performed 50 times for different number of users and Optimal meeting point is determined in each of these three cases. Figure 7 shows the graph of total distance travelled by users in each of these three cases. The X-axis shows the average distance travelled by users. Graph shows that users have to travel much more distance in case 1, when we do not consider direction of movement of users. In case 2 and case 3 these distances are nearly same.

Figure 8 shows the graph of number of cells to be searched to determine the optimal meeting point in each of these three cases. Graph shows that more number of cells are required to be search in case 1 and case 2 as compare to case 3. It shows that the search space is significantly reduced if we apply the two-level convex hull pruning.

4 Conclusions

Scheduling a meeting is a difficult task for people who have overbooked calendars and many constraints. The complexity increases when the meeting is to be scheduled between parties who are situated in geographically distant locations of a city and have varying travel patterns.

In this paper, we investigated the problem of *identifying a common meeting point for a group of users who have temporal and spatial locality constraints that vary over time*. We solved the above problem for a number of users on road network by using the GPS traces of the users.

We begin by mining historical GPS traces of individual user and mapped their locations onto predefined grid and predicted their location points for the given meeting time. We then applied two levels of convex hull based pruning to reduce the search space and determine the optimal meeting point. We have used predicted future direction of movement of the users to reduce total distance travelled by users before and after the meeting.

The method was evaluated on a large real-world GPS trace dataset and showed the effectiveness of our proposed method in identifying a common meeting point for an arbitrary number of users on the road network.

References

1. Khetarpaul, S., Gupta, S.K., Subramaniam, L.V., Nambiar, U.: Mining GPS traces to recommend common meeting points. In: Proceedings of IDEAS 2012, pp. 181–186 (2012)
2. Yan, D., Zhao, Z., Ng, W.: Efficient Algorithms for Finding Optimal Meeting Point on Road Networks. In: Proceedings of the VLDB Endowment, pp. 968–979 (2011)
3. Khetarpaul, S., Chauhan, R., Gupta, S.K., Subramaniam, L.V., Nambiar, U.: Mining GPS Data to Determine Interesting Locations. In: Proceedings of IIWeb 2011, WWW 2011 (2011)
4. Xu, Z., Jacobsen, H.-A.: Processing Proximity Relations in Road Networks. In: Proceedings of SIGMOD, pp. 243–254 (2010)
5. Chen, Z., Shen, H.T., Zhou, X., Yu, J.X.: Monitoring Path Nearest Neighbor in Road Networks. In: Proceedings of SIGMOD, pp. 591–602 (2009)
6. Zheng, Y., Zhang, L., Xie, X., Ma, W.: Mining correlation between location using human location. In: Proceedings of ACM GIS 2009, pp. 625–636 (November 2009)
7. Jeung, H., Liu, Q., Shen, H.T., Zhou, X.: A Hybrid Prediction Model for Moving Objects. In: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering (ICDE 2008), pp. 70–79. IEEE Computer Society, Washington, DC (2008)
8. Zheng, Y., Zhang, L., Xie, X., Ma, W.Y.: Geolife: Managing and understanding your past life over maps. In: Proceedings of MDM (April 2008)
9. Zheng, Y., Zhang, L., Xie, X., Ma, W.-Y.: Understanding mobility based on gps data. In: Proceedings of Ubicomp, pp. 312–321 (September 2008)
10. Simon, R., Frohlich, P.: A mobile application framework for the geospatial web. In: Proceedings of WWW, pp. 381–390 (May 2007)
11. Beeharee, A., Steed, A.: Exploiting real world knowledge in ubiquitous applications 11(6), 429–437 (2007)
12. Park, M., Hong, J., Cho, S.: Location-based recommendation system using bayesian user's preference model in mobile device. In: Proceeding of UIC, pp. 1130–1139 (July 2007)
13. Mouratidis, K., Yiu, M.L., Papadias, D., Mamoulis, N.: Continuous Nearest Neighbor Monitoring in Road Networks. In: Proceedings of VLDB, pp. 43–54 (2006)
14. Horozov, T., Narasimhan, N., Vasudevan, V.: Using location for personalized poi recommendations in mobile environments. In: Proceedings of SAINT, pp. 124–129 (January 2006)
15. Cho, H., Chung, C.: An Efficient and Scalable Approach to CNN Queries in a Road Network. In: Proceedings of VLDB, pp. 865–876 (2005)
16. Yiu, M.L., Mamoulis, N., Papadias, D.: Aggregate Nearest Neighbor Queries in Road Networks. IEEE Trans. on Knowl. and Data Eng. 17(6), 820–833 (2005)
17. Cooper, L.: An Extension of the Generalized Weber Problem. Journal of Regional Science 8(2), 181–197 (1968)
18. Preparata, F.P., Shamos, M.I.: Computational Geometry:An Introduction. Springer (1985)
19. Convex Hull and properties (March 28, 2013), http://en.wikipedia.org/wiki/Convex_set, http://en.wikipedia.org/wiki/Centroid
20. Accuracy versus decimal places at the equator (March 28, 2013), http://en.wikipedia.org/wiki/Decimal_degrees