VACATION PLANNING USING GEO LOCATION CLUSTERING

1. Abstract

The discovery of places of interest in large cities is a major problem for those who are interested in sightseeing. An interactive and automatic travel route planning service is highly desired to plan a trip that is customized according to user preferences. We propose a project to find the travel route that helps the visitor in order to cover maximum number of places with in a given amount of time.

This project consists of two modules, one for local sightseeing and for global sightseeing. For local sightseeing we will be applying path finding algorithm for given time and for customized locations. For global sightseeing hdbscan is used for data cleaning to detect and remove outliers, then the cleaned data is clustered using k-means(to minimize inter cluster similarity), the path for clustered data is found using path finding algorithm.

Keywords: clustering, hdbscan, spatial data, geo-location

2. Introduction

One of the most important steps for a tourist to plan a trip is travel route planning. Prior to travel to a location which was unknown most of the tourists have doubt how to plan a vacation. Although visitors can take help of travel guide, the process is generally not efficient and the results may not be customized. The main goal of this application is to provide an optimal visit plan for a tourist visiting a location and selecting some places out of given tourism attraction places in a city. The effective implementation of this project faces two problems, how to identify the far away points and how to form the clusters for the selected locations. After addressing the above challenges, we need to find a route map as the user specified.

First Challenge is solved by hierarchical density based spatial clustering of applications with noise(hdbscan), using hdbscan outliers are detected and removed and clusters are formed using k-means clustering algorithm. The outcome of k-means clustering algorithm is passed to path finding algorithm to get route map of geo-clustered data. This ensures the requirement of covering maximum locations within a specified amount of time.

Location recommendation is achieved using the feedback given by the user. The recommendation system is used to filter the location that seeks to predict rating a user would give to a location. The whole idea of a recommendation is to provide a beneficial guide that will not only resolve certain issues, but result in a beneficial outcome [23].

3. Related Work

Spatial data mining problems have been investigated extensively over the past several decades. How to best plan a vacation is really a challenging task, which includes the optimal utilization of the visitor time in planning a vacation. Hamza Bendemra has contributed his idea of planning a vacation for Paris, which was based on the idea of using partition based clustering like k-means or hdbscan to partition the data and schedule a plan for the partitioned data [1] .

4. Methodology

4.1 Data Flow

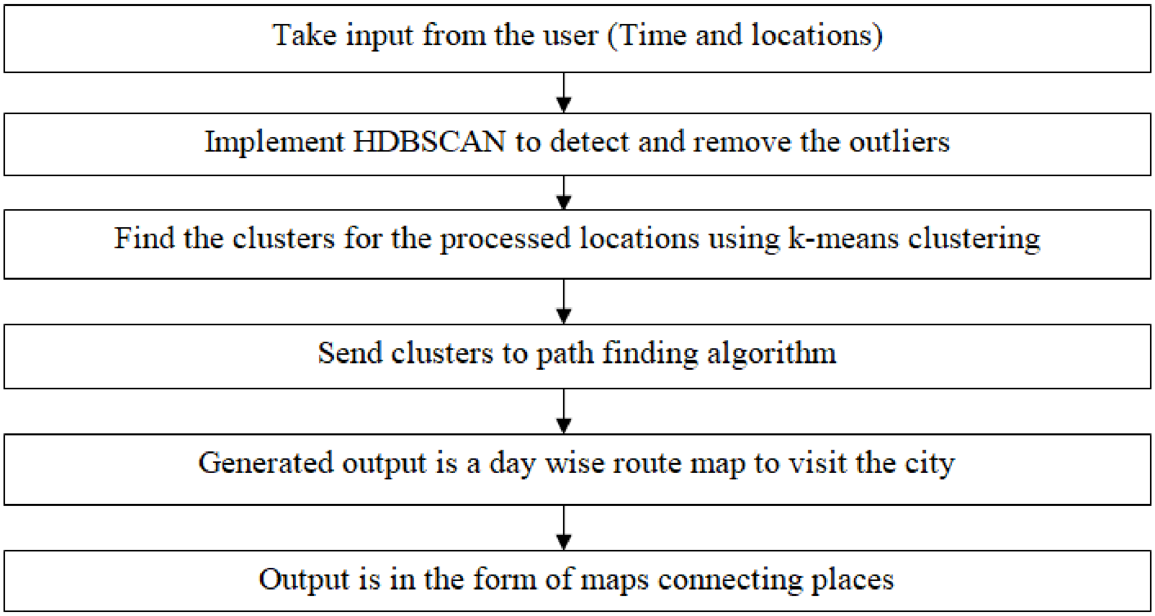
Take input from the user (Time and locations)
Implement HDBSCAN to detect and remove the outliers
Find the clusters for the processed locations using k-means clustering
Send clusters to path finding algorithm
Generated output is a day wise route map to visit the city
Output is in the form of maps connecting places

Figure 1-Data Flow

4.2 Data Cleaning and pre-processing

The existence of geo-locations that are not reachable for user in the specified amount of time is known. This is often referred as noise. Noise or outliers should be removed from the data for the better performance of vacation planning, failing so would result in solution which is not a optimal one.

4.2.1 HDBSCAN

To tackle the problem of noise from the geo-locations we have used Hierarchal density based spatial clustering of applications with noise (hdbscan) to detect outliers (the points that are far from the remaining locations). The outliers can be removed from the actual plan [2][3].

HDBSCAN computes clusters based on the reachability of the selected places. The two parameters that are accepted by hdbscan were the minimum points that are required to classify for a cluster and the distance between two points that are supposed to be in same cluster [20].

4.2.2 K-means

In general, unsupervised learning methods are useful for datasets without labels and when we actually could not predict the particular outcome. Output of hdbscan is passed to k-means algorithm, we will get the day wise plan for the visitor to visit the city [7][9]. K-means algorithm computes clusters based on the minimum mean square distance from the centre [11][14]. In this algorithm we are going to cluster the places based on the number of days.

4.3 Optimistic plan

4.3.1 Initialization

In this algorithm, the execution will start from the source. We will calculate the time taken to move from source to destination. If the above calculated time is less than the available time, then we find the new source by finding minimum of visit time and travelling time combined. In this process we find the new source and we mark the previous source as visited. The algorithm will stop after failing the first mentioned condition.

4.3.2 Algorithm

- Mark all the points as not visited, take input from user the available time.
- Identify the source and the destination, note visit time (vt) for all the points on the plane.
- Calculate time taken to move from source to destination ,distance will be in kilometers and assume the speed to be 30kmph
- While the time taken is less than the available time, then do step 5 to 10
- Find the updated visit times (uvt) to all the points on the plane as
  $uvt(A_i) = \{vt(A_i) + [(\text{distance between source , } A_i)/speed]*60 \}$ (1)
- Find the place with minimum uvt.
- Mark the source as visited.
- Make the place with minimum uvt as new source.
- Update the available time as
  Available time = available time – $uvt(A_i)$ (2)
- Calculate time taken to move from source to destination and goto step 4.

4.3.3 Working

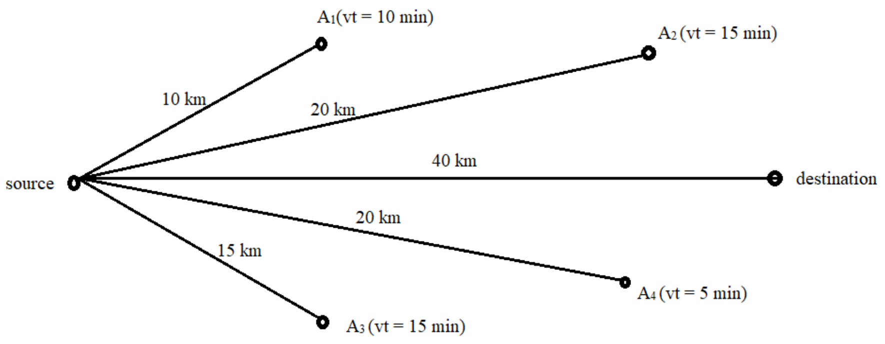

Figure 1- Initial map showing places with visit times and distances.

Here, let us consider the map (Figure 1) as the initial map to go for the path finder algorithm. Consider the available time =200 minutes.

Before start of the algorithm,

Place	Visit time(min)	uvt(min)	visited
source	0	0	0
A_1	10	0	0
A_2	15	0	0
A_3	15	0	0
A_4	5	0	0
destination	0	0	0

Table 1-Iteration 0

- Calculate the time taken for source to destination (in the above case it is 80 minutes).
- The calculated time is less than the available time. So, go for iteration 1.
- Calculate the updated visit times (uvt) for every place ($A_{i \text{ (where } i = 1 \text{ to } 4)}$) using Eq.(12).
- Choose the place with minimum uvt from the below table as a new source (in this case A_1) if it is not visited, mark the previous source (Source) as visited
- Calculate the remaining time .The uvt for places will be as follows

Place	Visit time(min)	uvt(min)	visited
source	0	0	1
A_1	10	30	0
A_2	15	55	0
A_3	15	45	0
A_4	5	45	0
destination	0	80	0

Table 2-After iteration 1

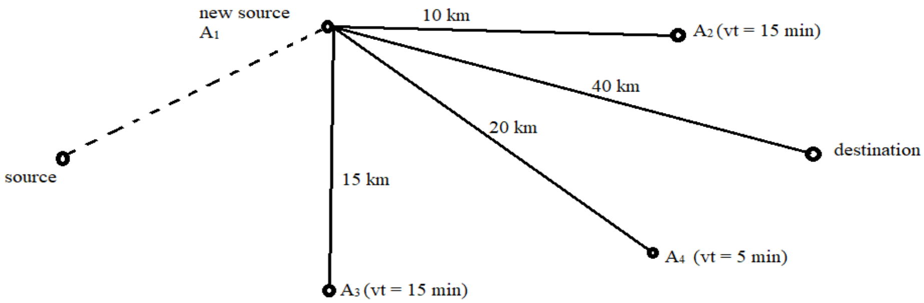

Figure 2- Map showing places with visit times and distances after iteration 1.

Do the above procedure for the new source and destination (Figure 2)

5. Results

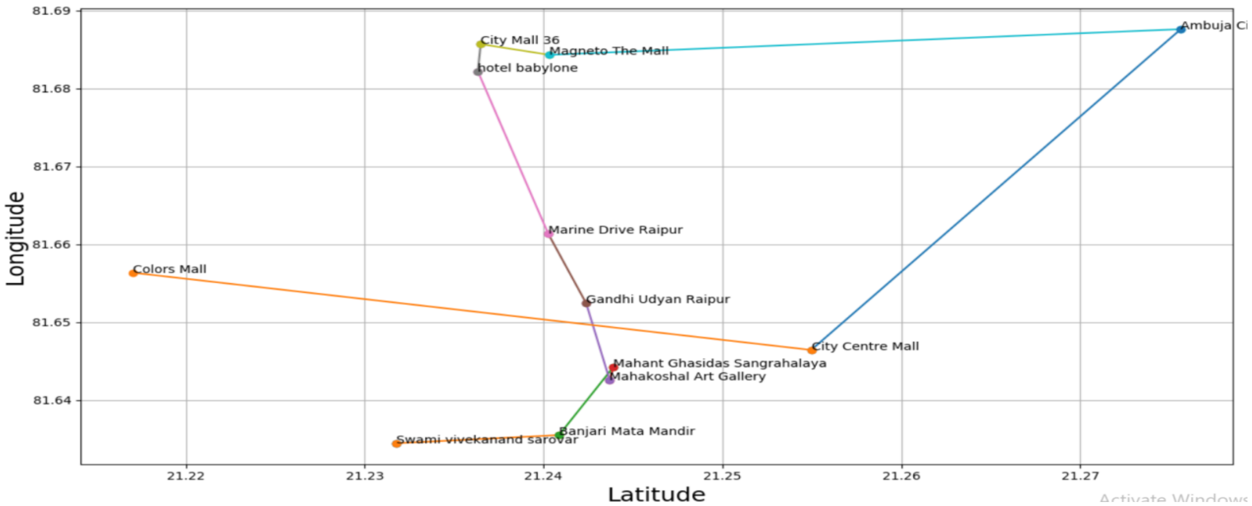

Figure 3- Day 1 visit plan

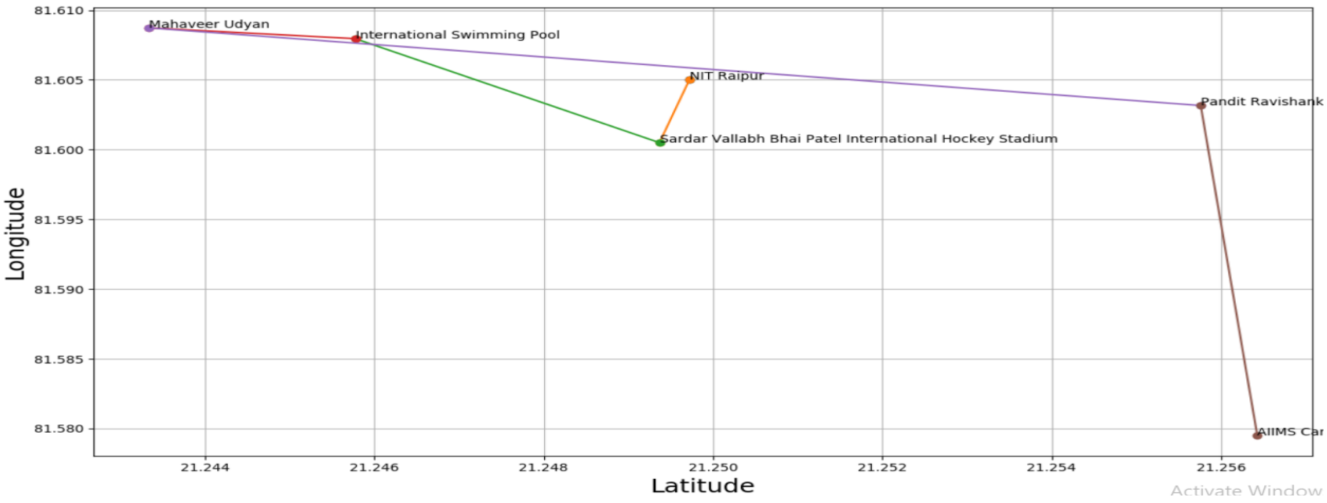

Figure 4- Day 2 visit plan

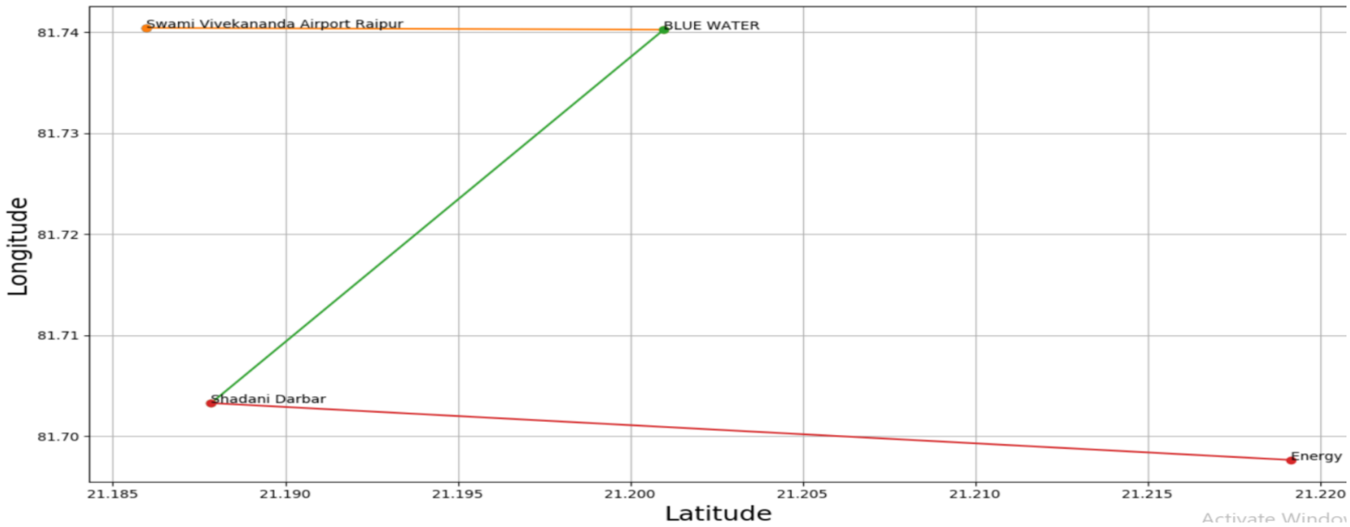

Figure 5- Day 3 visit plan

6. Discussion on Results

As previously discussed in section 4.2.1 the output of this process is a map consisting of locations that should be traversed sequentially according to path in order to cover maximum number of locations in specified amount of time. Fig 3 is the route plan for visitor on first day of the trip. Starting at the Swami Vivekananda sarovar, ends at Colors mall. Here the source is taken as nearest location in the neighbourhood of destination from the previous cluster and the next location on the route is the nearest location from the present location.

Visitor has selected for a three day plan so Fig 4 and Fig 5 are the route plans for day 2 and day 3 where the visitor starts at NIT Raipur and end at AIIMS Raipur in day2 plan. Whereas visitor starts at Energy park and end his vacation at Airport for Day 3. The above mentioned vacation plan is optimal in the sense that the visitor covers the maximum number of places with in the specified time which meets the requirement of the project.

7. Conclusion

In this paper we have proposed one new algorithm for path finding, also the application developed will be giving an greedy plan. It will cover the maximum number of places which were given by he visitor within the given amount of time. The results show that the performance of the application was competitive and path proposed by the map when traversed with all the possible combinations of the locations is the best path in the sense that it covers the maximum locations with the utilization of given time. Detection of noise location that cannot be travelled by the visitor and removal of noise adds to performance of path finding algorithm.

8. References

[1] Basic idea available on: https://towardsdatascience.com/using-unsupervised-learning-to-plan-a-paris-vacation-geo-location-clustering-d0337b4210de

[2] Martin Ester, Hans-Peter Kriegel, Joerg Sander, Xiaowei Xu (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Institute for Computer Science, University of Munich. Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining.

[3] Campello, Ricardo JGB, Davoud Moulavi, Arthur Zimek, and Jörg Sander. "A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies." Data Mining and Knowledge Discovery 27, no. 3 (2013): 344-371.

[4] R. Campello, D. Moulavi, and J. Sander, *Density-Based Clustering Based on Hierarchical Density Estimates*

[5] McInnes L, Healy J. *Accelerated Hierarchical Density Based Clustering*
In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, pp 33-42. 2017
[pdf] <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8215642>

[6] R.C.; Hennig, C. (2015). "Recovering the number of clusters in data sets with noise features using feature rescaling factors". Information Sciences. 324: 126–145.:aRXiV1602.06989. doi:10.1016/j.ins.2015.06.039

[7] Alon Vinnikov and Shai Shalev-Shwartz (2014). "K-means Recovers ICA Filters when Independent Components are Sparse" (PDF). Proc. Of Int'l Conf. Machine Learning (ICML 2014).

[8] D., Manning, Christopher (2008). Introduction to information retrieval. Raghavan, Prabhakar., Schütze, Hinrich. New York: Cambridge University Press. ISBN 978-0521865715. OCLC 190786122

[9] Mirkes, E.M. "K-means and k-medoids applet". Retrieved 2 January2016.

[10] Bradley, Paul S.; Fayyad, Usama M. (1998). "Refining Initial Points for k-Means Clustering". Proceedings of the Fifteenth International Conference on Machine Learning

[11] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, S. Vassilvitskii "Scalable K-means++" 2012 Proceedings of the VLDB Endowment.

[12] Zhang, J.; Zhu, M.; Papadias, D.; Tao, Y.; Lee, D.L. Location-based spatial queries. In Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, CA, USA, 10–12 June 2003.

[13] Zaïane, O.R.; Foss, A.; Lee, C.H.; Wang, W. On data clustering analysis: Scalability, constraints, and validation. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Taipei, Taiwan, 6–8 May 2002.

[14] Ragesh Jaiswal and Nitin Garg. Analysis of k-means++ for separable data. In Proceedings of the 16th International Workshop on Randomization and Computation, pp. 591?602, 2012

[15] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In Proceedings of the 18th annual ACM-SIAM symposium on Discrete Algorithms (SODA?07), pp. 1027-1035, 2007.

[16] Andrea Vattani. k-means requires exponentially many iterations even in the plane. In Proc. of the 25th ACM Symp. on Computational Geometry (SoCG), pages 324-332, 2009

[17] Arthur, David, Bodo Manthey, and H. Roglin. ”k-Means has polynomial smoothed complexity.” Foundations of Computer Science, 2009. FOCS’09. 50th Annual IEEE Symposium on. IEEE, 2009.

[18] Agarwal, Manu, Ragesh Jaiswal, and Arindam Pal.”k-means++ under Approximation Stability.” Theory and Applications of Models of Computation: 84.

[19] N. Dalvi, R. Kumar, A. Machanavajjhala, and V. Rastogi. Sampling hidden objects using nearest-neighbor oracles. In SIGKDD, 2011.

[20] C. Xia, W. Hsu, M. L. Lee, and B. C. Ooi. Border: efficient computation of boundary points. Knowledge and Data Engineering, IEEE Transactions on, 18(3):289–303, 2006.

[21] G. Karypis, E.-H. Han, and V. Kumar. Chameleon: Hierarchical clustering using dynamic modeling. Computer, 1999.

[22] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In KDD, 1996.

[23] Rancesco Ricci and Lior Rokach and Bracha Shapira, Introduction to Recommender Systems Handbook, Recommender Systems Handbook, Springer, 2011, pp. 1-35