

How to cluster nearest unique nodes from different classes using JJCluster in Wisp application?

Journal Title
XX(X):??–??
©The Author(s) 2016
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/

SAGE

Jimut Bahan Pal¹

Abstract

The work of finding the best place according to user preference is tedious task. It needs manual research and lot of intuitive processes to find the best location according to some earlier knowledge about the place. It is mainly about accessing publicly available spatial data, applying a simple algorithm to summarise the data according to given preferences, and visualising the result on a map. We introduced JJCluster to eliminate the rigorous way of researching about a place and visualizing the location in real time. This algorithm successfully finds the heart of a city when used in Wisp application. The main purpose of designing Wisp is to find the best location according to a set of preferences in a map. This has interesting application in finding the perfect location for a trip to unknown place which is nearest to a set of preferences. We also discussed the various optimization algorithms that are pioneer of today's dynamic programming and the need for visualization to find patterns when the data is cluttered. Yet, this general cluster algorithm can be used in other areas where we can explore every possible preference to maximize its utility.

Keywords

JJCluster, Location, User preferences, Visualization, and Wisp

1. Introduction

1.1. Background

To visit an unknown important place we search hotel which is close to a set of preferences and we plan accordingly to explore too many places in a short time. The preferences may be river, museum, shopping mall, lake, cafe, hospital or even movie theatre. There are many possibilities of selecting hotel, but we want the hotel which is only adjoining to every preference. We need the result fast without manually researching about the place. Here, we consider our preferences as a class. Intuition would tell us to search in Google maps to select restaurant. The approach may not be perfect always, and will take a lot of time. Here the main aim of designing Wisp [1, 2] application is to visualise the place according to user preferences in real time, without any botheration of researching manually. We introduced here an algorithm for finding the cluster (the boundary location), named as JJCluster (which stands for Jimut-Jisnoo Cluster). The software module takes a list of places containing corresponding class, name, latitude, longitude, and displays a map for the Cluster formed at real time. This algorithm can be used in a variety of research areas of general clustering, where we need to select unique closest nodes from each class.

The JJCluster marks a boundary across the location which will be the best place for those preferences. Selecting a coordinate inside that boundary may be one of the best places for stay. The other applications of this software may find developed areas according to certain set of preferences. Major factors that boosted the development of this Wisp application are discussed below.

1.2. Availability of data

Cheap storage [3] and sensors have made it possible to collect variety of data in huge quantities. Users generate a large number of data in social networking platforms. The Wisp application used Marker Cluster as its core algorithm until the design of JJCluster. Beilschmidt et al. (2017) [4] proposed Marker Cluster algorithm to cluster huge dataset into a Map as circles. The quantity of data points may lead to visual clutter and most of the browser may not support such data points in a map, which would result in occlusion and loss of information. The shape of circle in Marker Cluster changes dynamically on zooming in or out of the map, which helped in optimising the available resources and visualising any number of points. By using Marker cluster we couldn't determine whether a place have all of the preference classes or some of them are in large quantity, so it may lead to wrong visualisation and selection of hotspots.

Similarly, location based social networks (LBSN) have recently gained popularity to help targeted users [5] to get their needs. The use of local experts, the natives, to review and suggest areas of interest to users with similar behavioural pattern has been the most advanced way of recommending places. Semantic similarities and geographic correlations between users are captured simultaneously by the concept of local experts. This technique is used widely for collecting data for application programming interfaces

¹Department of Computer Science, Ramakrishna Mission Vivekananda Educational and Research Institute, Belur, India.

Corresponding author:

Jimut Bahan Pal, Contact: +918902209098.
Email: jimutbahanpal@yahoo.com

(API's) which helps in designing of Wisp application.

The following section discusses about the related work done by researchers in various fields of data visualisation (both in 2D and 3D), dynamic programming for resource allocations, and designing of recommendation systems to targeted users.

2. Related Works

Visualizations can help human to predict patterns [6] and areas of concern in a dataset, which may have inherently hidden relations. It gives ideas for processing the data further. It may help policy makers [7], volunteers, programme managers, donors and non-governmental organisations to invest money in critical places by getting information from visualizations. In the domain of big data, scientists use statistical and data mining techniques [3] to find the information in which they are interested. When the data is huge scientists use certain partitioning techniques to visualise the data along with clustering methods. Big data analysis provides opportunity to test hypothesis with new data, which have potential to yield new insights to classical research questions across multiple disciplines [8]. Old methods such as surveys are being replaced by sensor data, which needs a lot of pre-processing to find valuable and accurate information that was not available by using old methods. There was constant breach of privacy, when crowd sourced data was collected. Quality measures and pre-processing [9] can help to minimise such privacy risks. This includes basic steps of formatting, cleaning, validation, making the data anonymous before publishing. Recent years have boosted the popularity of location based services, where users generate huge amount of data through social media. Such data can be used for recommending important places [5] for class of users. Geodemographic [10] may help to find social similarities amongst variables within a dataset using cluster analysis and profile matching, which are indicators of social, economic and demographic conditions of neighbourhood. These forms the basis of building crowd sourced platforms for data collection which are necessary for visualising data in large scale.

Researchers built a cyber-infrastructure [11] solution named PolarGlobe, which enables comprehensive collaboration and analysis. Their solution does not require installation of plug-ins, since that is a web based framework. State of the art tools used for measurement of climate change risk, adaptation [12] and vulnerability assessment use web as their common infrastructure. The data driven approach analysis is dominating in research of transportation; with the increase in amounts of data collected from accurate transport system sensors, which was earlier determined with the help of mathematical equations due to scarcity of data. In this way data are collected for gaining insights to improve existing systems.

2.1. Dynamic programming algorithms

Dynamic programming algorithms are useful in different fields of study ranging from telecommunications,

transportations, electrical power systems, water supply, and emergency services which provide useful human resources to certain areas. In designing the optimisation route of network flow there are some constraints, such as the topological characteristics of a networked infrastructure where we choose the best path that is feasible for locomotion. Computer scientists, economist, statisticians, cartographic researchers made significant contributions for finding the optimal path and its solutions for a given set of problem. Optimisations can even help in designing cities by putting areas of critical need such as hospitals [13] in those places which are accessible very quickly. Even allocation of charging stations for electric vehicles [14] are promising solution to tackle climate change impacts for improving air quality and enhancing growth sustainability. Management and location of underground utilities [15] such as depth of cables and pipelines are great concern in many countries, misinformation of which can even result in tragic accidents.

Bellman was one of the earliest scientists who introduced dynamic programming [16, 17]. He worked on dynamic programming problem for finding the shortest path between two points in a graph of interconnected roads and varied density of traffic. This marked the beginning of a new field of optimisation problem known as the dynamic programming problem. Scientists subsequently worked on the problem of optimising certain features in a problem. The most famous problem is Travelling Salesman Problem or the TSP, which is still an extremely interesting topic about whether there exists any algorithm for which can we do better? There is Floyd-Warshall algorithm for finding the shortest path between any two paths in a graph. This algorithm can help to find the distance between any two points of interest in a graph. Scientists can use optimisation when there are interdiction scenarios in a network.

2.2. Cartographic visualizations for pattern recognition

Dodge, Weibel, and Lautenschutz [18] used two phased approach for studying movement pattern in more systematic and comprehensive way. They developed a conceptual framework for movement behaviour of different moving objects, and then classifying it. They devised a toolbox for the study of data mining algorithms for the movement of animal, human, traffic, industry goods, bee etc. which follow a generic spatiotemporal pattern.

Delort [19] examined a technique for visualising large spatial dataset in Web Mapping Systems. The algorithm created a cluster tree, which helped to visualising the cluster in a map without cluttering itself, at any given scale. Though, the algorithm was too slow for its time, it was later improved by many researchers and used widely till date, this is the basis of the idea for designing JJCluster (coincidentally), but here we focus on quality (classes) over quantity. Phan, Xiao, Yeh, Hanrahan, and Winograd [20] made an algorithm using binary hierarchical clustering to formulate flow maps which are used immensely in studying and visualising migration of birds, people, flow of traffic in a network, flow of river etc. Moreover, Stachoň, Šašinka, Čeněk, Angsüsser,

Kubíček, Štěrba, and Bilsková [21] also studied the selection of graphic variables, in a new perspective to use graphics in cartographic visualisations. We used circle markers in the visualisation of this application, along with dynamic legend generators.

Zhu, and Xu [22] reduced the influence of location ambiguity using geotags and geo-cluster to employ semantic features of geo-tagged photographs through clustering. This method measures the semantic correlation which has synonym representation of tags, but not same tags. Delmelle, Zhu, Tang, and Casas [23] proposed a web based open source geospatial toolkit, which are particularly needed in developing countries lacking GIS software. It provides a user friendly and interactive web based framework for analysing the outbreak of vector borne diseases. They analysed the outbreak of dengue for efficient and appropriate allocation of human resources to minimise them. Etten and Jacob [24] et al. provided a package in R for calculating routes and distances in calculating various heterogeneous geographic spaces represented as grids. They used circuit theory and geographical analysis to find out various distances between routes.

2.3. Recommendation Systems to targeted users

D'Ulizia, Ferri, Formica, and Grifoni [25] enabled the relaxation of three kinds of query constraints, namely semantic, structural and topological, by using GeoPQL which translated natural language queries to map visualisations. They used graph theoretic procedures for query approximations. It helped those users who were unaware of exactly what they were looking for due to complexity of finding and locating the data points with naive tags. Cai, Lee, and Lee [26] studied the movement of humans to get better semantic knowledge about a place, through the help of GPS and social media. Dareshiri, Farnaghi, and Sahelgozin [27] improved the existing recommendation capabilities of geo portals. In their approach, the geoportals analyses the user's activity and recommend geospatial resources to users, based on their desires and preferences. Zhao, Qin, Ye, Wang, and Chen [28] identified the centre of cluster in map with the real time traffic data of taxi. These data were further used to study the popularity importance of a place over certain duration of time. The study showed the crowded areas, which were needed by the urban transportation and planning management authorities to reduce congestions.

2.4. Visualising geospatial data in 3D

Visualisation in 3D helps to find information, which is almost impossible to find by looking at the data. Three dimensional representations of Geographic information on computers are known as GeOVES [29] or Geospatial Virtual Environment. These environments can engage participants to explore virtual landscapes and knowledge-scapes along with providing different images of reality. Azri, Anton, Ujang, Mioc, and Rahman [30] studied various issues that were needed to be resolved for using geospatial data infrastructure. They optimised space, time accession of data, data model, and spatial access method etc., for utilising and

managing high precision laser data, which could take 63 GB of space in a hard drive. The analysis of 3D visual data with the help of existing 3D software tools such as Autodesk Maya can help to find the related pattern in the data which can be used carefully by researchers to provide solutions to existing problems. The approach contributes to intuitive comprehension of complex geospatial information for decision makers in security agencies as well as for authorities related to urban planning for certain correlation between the urban features.

3. Materials and Methods

3.1. Working of the algorithm

The algorithm is very simple, once we write it with a pen and paper. We consider each point in the map as node of a graph. The aim is to select the shortest distance from each of the individual classes. We describe the general working of the algorithm in Fig. 1,2,3,4 and 5. Firstly, we start by choosing the elements of the first class in the graph as shown in Fig. 1 (the list S is coloured as black in the graph for every step). We choose element 1 from class A and calculate its distance from all the items of the other classes, i.e., from 1 to B's 1, B's 2, C and class D. Then we sum up, i.e., $6+8+12+13 = 39$ and store it in a list. We do the same thing for node 2 of class A; we get the distance $9+7+6+5 = 27$ and add it to the list. Since, there is no other element from the first class, we will choose the minimum from this list i.e., class A's point 2 as shown in Fig. 2. We store this in the list of selected nodes, called S. We can now say that this point in S is nearest to all the other points of every other class. So, the other points of other classes will be nearest to this point. Again, we calculate the same thing for class B's node 1, but here from point 2 of class A (which is the only element present in the list S), we store it in an empty list. The value that we got is 9. We do this for the second element of class B, and store 7 to the list. Now, we select the nearest point from the entire list S, i.e., node 2 from class B. We store this to the list S as shown in Fig. 3. We will now find the distance of the next class's elements from each element of list S, add them up and select the minimum distance among the classes. Since, there is only one element present in the remaining dataset, so we store 1 from class C, after calculating the distance $5+8 = 13$, to list S as shown in Fig. 4. Now S has 2 from class A, 2 from class B, and 1 from class C. We continue this process for the next class, by calculating node 1 of class D from every element of S. We calculate this distance as $6+6+3 = 15$. So, the final list S have the points 2 from class A, 2 from class B, 1 from class C, 1 from class D as shown in Fig. 5. We draw the lines to separate it from other points and visualise the cluster in a map later in this paper for real world data.

If there is no member of a class in an area, then that particular class is not considered and the data is skipped. All the classes (preferences) are treated equally in importance by the algorithm.

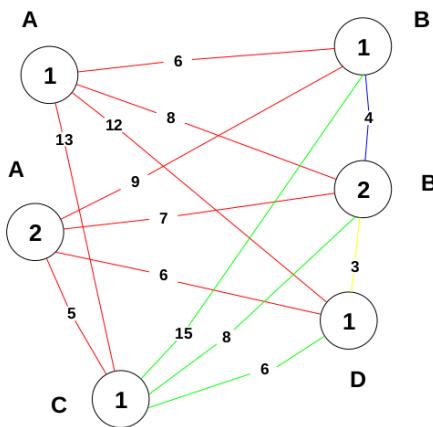


Figure 1. The original graph in the beginning.

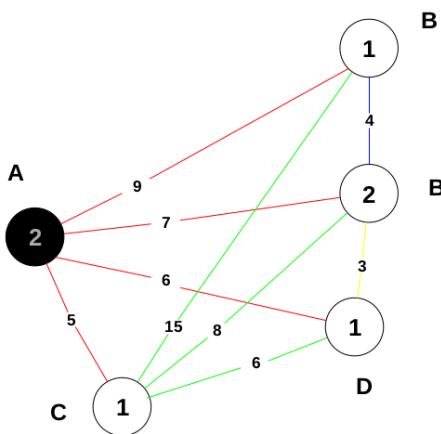


Figure 2. The nearest point from the first class is selected.

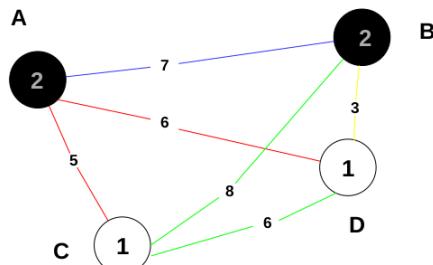


Figure 3. The closest point from the second class to the first class is selected.

3.2. Collecting data for Wisp application

The algorithm for the JJCluster is implemented using Python3. It uses Folium [31, 32] library to visualise geospatial data. Social media is generating large amount of data, after careful analysis of the data, it can be used for mapping and understanding of the evolving human landscape [33]. These data provide ambient geospatial information such as momentary social hotspots according to majority preferences. These provide a better understanding

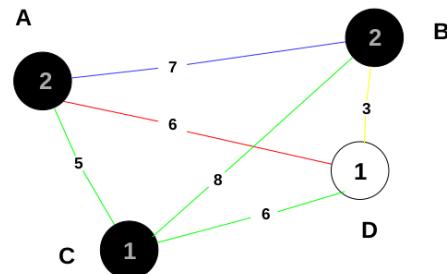


Figure 4. The third point (which is closest since it is only present once) from the third class to the first and second chosen points of their classes is selected.

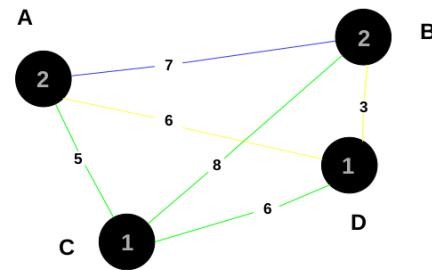


Figure 5. The fourth point from the fourth class (which is closest since it is only present once) from all the selected classes is chosen.

of those features of a place, which cannot be estimated easily.

The data is collected through Foursquare [34; 35] API, which is a social location service allowing users to explore world around them. It is like a general social media platform, where users can connect to their friends having foursquare account by downloading the app in mobile devices. The foursquare API has restful services, which helps the developers to create certain applications based on the data as shown in Fig. 6. The availability of foursquare is 24 x 7, and it is maintained by rich community of users. There are 150,000 developers currently building location-aware experiences with foursquare technology and data [49]. We have chosen foursquare API over other API in this application because we can focus on a certain location and get the data in brief.

The cleaned JSON dump is shown in Fig. 7. This data is an instance of a panda's data frame, fed to the JJCluster module for further processing and visualisation. This is an instance of pre-processed data for the location Tokyo. The data is divided into class, name, latitude and longitude. This forms the raw data for generic clustering module. Any data following this structure can be passed as input to the JJCluster module.

The user interface here is self-intuitive, passing the necessary data as inputs to the console for visualisation. The user gives the radius, location, preferences as input as shown in Fig. 8. The location's latitude and longitude is looked up from a remote server using Geocoder API. The

4. Results and Discussions

4.1. Visualisations generated in Wisp

Wisp is a small application of the JJCluster algorithm. It takes foursquare secret, foursquare ID, name of the location, number of preferences, preference list, and the type of map as input and displays the map by a custom made HTTP server on real time as shown in Fig. 6. The source code of the Wisp application is uploaded in GitHub. The quality and correctness of the visualisation will depend on the data. Since, foursquare is community maintained and trusted by developers, it should provide accurate information about a place. There are certain places which are not listed by the foursquare API, or the number of reviews is too low. Such places are usually under developed and such organization's owner probably doesn't list themselves in the foursquare API platform. The porting of this algorithm with Google maps data may lead to more accurate results, the Google maps enlists almost all the places present in any given location. Since there is an easy premium integration of Google services, this clustering algorithm could be used by Google map data to give highly accurate map results. The wisp application is free alongside with the regular data that is fetched from foursquare API. There is a premium package available in foursquare and might cost money, but using premium Google services to get pin point location of places will be a better option for getting accurate results from this clustering algorithm. The maps that are obtained by using foursquare free services and are implemented using Python3 are shown in Fig. 9, Fig. 10, Fig. 11, and Fig. 12. The preferences are plotted in a legend for each map on the left side, which forms a unique class. Each of the data points in the map are treated as individual nodes in JJCluster. The minimum nodes are selected and are successfully plotted in the folium map of the countries.

One of the many visualisation generated by Wisp application is shown in Fig. 9, targeting Tokyo, Japan with 15 preferences within a radius of 9 km, which forms the biggest size polygon among all the maps. Since, the number of preferences are high, the zonal boundary of the area increases because all the places may not be present in a small span of area. We can see the map of Kolkata, India in Fig. 10. There are eight preferences i.e., restaurant, gym, park, ice cream, movie theatre, hospital, river and books that can be treated as unique set of classes in a map. This is visualised in a browser. We can see details of each data points with the help of pop-ups. The total amount of data points shows the popularity of the foursquare location in that place. The whole map is in a radius of 10 km, but a certain part of the map is shown. We can also determine the high market value of property such as houses and flats in such places if it is close to many preferences. Similarly, in Fig. 11, we see the small part of Russia that is clustered and zoned (among all the bigger part of the data points) according to the algorithm. We studied a more diverse domain of classes in Fig. 11, for New York. We may find different shapes and sizes of zonal polygons in a city for different preferences. It may even happen that the data are updated in foursquare API and the shapes of the polygon changes over time. Since this is a community

maintained API, it helps users to get the most recent data according to their needs.

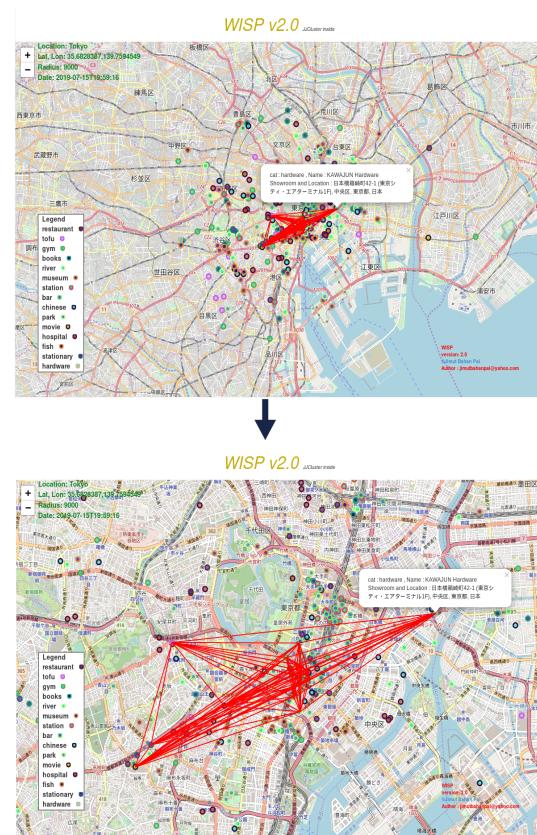


Figure 9. Open Street map of Tokyo with 15 preferences in a radius of 9 Km (above: Zoomed out, below: Zoomed in).

4.2. Optimisation matrix of the visualisations

The names of the places are treated as unique nodes, minimum distance is found by selecting the nodes from different classes and adding them to the selected class's list S. We optimised the distance by applying the JJCluster algorithm. We define certain terms related to optimization matrix.

D: The distance of the considered node to all the nodes in the list S.

T: The total distance of all the nodes in a particular class to all the nodes in the list S.

k: $\frac{T}{D}$, the optimization factor.

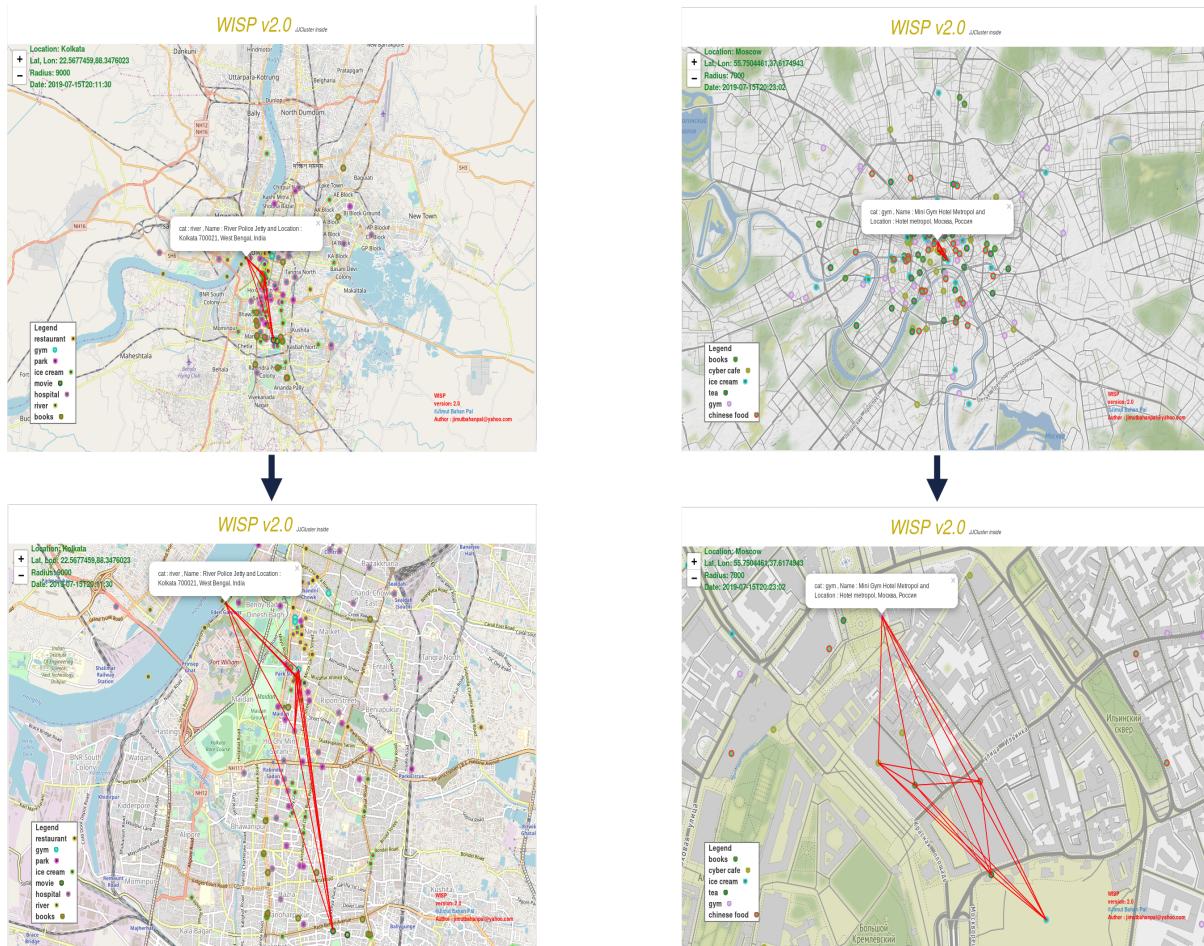
The optimisation matrix for Tokyo is shown in Table 1. Here we name the list S by its class as there is a constraint of place in the table. We note that it contains the name of those points, which are represented aside for its corresponding class. As per the algorithm, the list S takes the minimum of the selected points and hence, the size of list increases after each iteration.

The optimisation matrix corresponds to actual data fetched from the foursquare API, and scaled in the map

Sl. No.	Name	Class	List S	Latitude	Longitude	Distance (D)	Total Distance (T)	Optimization Factor (k)
1	Coffee and Books	books	[books]	55.754403	37.622445	354.560790	10618.142223	29.947311
2	Bosco Cafe	cyber cafe	[books, cyber cafe]	55.754878	37.620674	0.122773	61.572129	501.513267
3	Ice Cave	ice cream	[books, cyber cafe, ice cream]	55.751477	37.628756	1.143593	146.870961	128.429429
4	Moscow Tea	tea	[books, cyber cafe, ice cream, tea]	55.752457	37.626123	0.948188	195.201380	205.867904
5	Mini Gym Hotel Metropol	gym	[books, cyber cafe, ice cream, tea, gym]	55.758079	37.620857	2.369243	393.457266	166.068773
6	Royal Chinese Restaurant	chinese food	[books, cyber cafe, ice cream, tea, gym, chinese food]	55.754480	37.625580	1.620614	284.816591	175.746076

Table 3. Optimisation matrix obtained for Moscow using JJCluster

Sl. No.	Name	Class	List S	Latitude	Longitude	Distance (D)	Total Distance (T)	Optimization Factor (k)
1	Museum of Chinese in America	chinese	[chinese]	40.719361	-73.999086	448.537714	15335.590138	34.190191
2	Soho Thai	thai	[chinese, thai]	40.720117	-73.999504	0.091115	70.532027	774.101656
3	Kaigo Coffee Room	coffee	[chinese, thai, coffee]	40.718633	-74.000367	0.315488	105.182689	333.396911
4	Indigo Books HQ Design Studio	books	[chinese, thai, coffee, books]	40.719422	-73.997885	0.485303	223.256302	460.035156
5	Tribeca Soho Animal Hospital	hospital	[chinese, thai, coffee, books, hospital]	40.720578	-74.004869	2.000487	586.780255	293.318670
6	Columbus Park	park	[chinese, thai, coffee, books, hospital, park]	40.715603	-73.999884	2.417840	360.109107	148.938377
7	Onieal's Grand Street Bar and Restaurant	bar	[chinese, thai, coffee, books, hospital, park, bar]	40.719593	-73.997966	1.553863	196.666538	126.566175

Table 4. Optimisation matrix for New York using JJCluster**Figure 10.** Open Street map of **Kolkata** with 8 preferences in a radius of 9 Km (above: Zoomed out, below: Zoomed in)**Figure 11.** Stamen Terrain map of **Moscow** with 6 preferences in a radius of 7 Km (above: Zoomed out, below: Zoomed in).

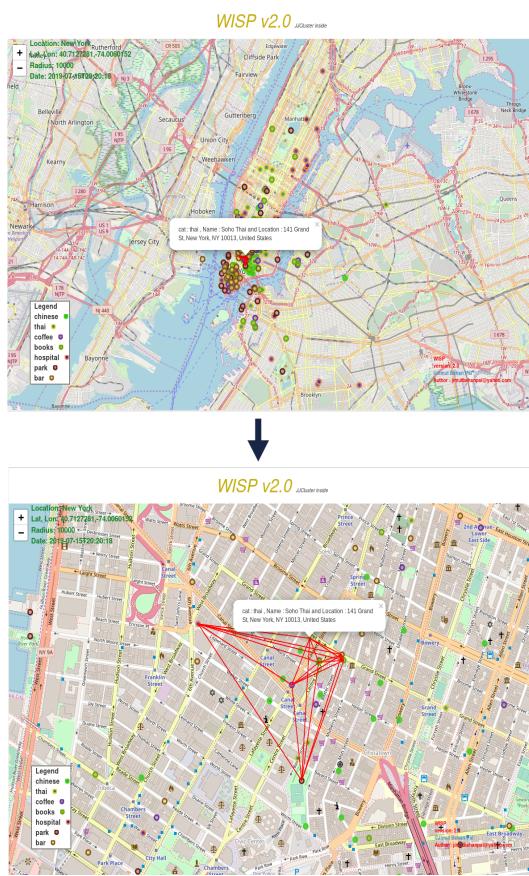


Figure 12. Open Street map of New York with 7 preferences in a radius of 10 Km (above: Zoomed out, below: Zoomed in).

applications.

The main objective for designing this Wisp application is to eliminate incorrect manual search. This clustering algorithm merely needs data in a particular format, so hacking and tweaking it for suitable uses are encouraged, since this is an open sourced project.

5. Conclusions and Future Scope

Creation of real time visualisation application is successful using JJCluster algorithm. We studied an application domain of the JJCluster algorithm for selecting the best places according to preferences in a map using Wisp application. The main objective of designing this JJCluster algorithm is to optimise and select the data points in such a way that it minimises the total distance of every nearest nodes from different classes. This algorithm can be used by researchers of different fields like statisticians, economists, cartographers, general scientists who are interested in finding certain hotspots according to classes. This may also be used to deploy drone swarms in those locations, where critical action is needed, which is determined by the algorithm, since this minimizes the total distance. Applications for this algorithm may further attract interdisciplinary researchers to evolve this algorithm according to their needs. It may happen that an oceanographer is studying about a place and there are too many classes of objects in his research. The objects with unique classes may be scattered over a wide

place, but using this algorithm he may be able to select and find those places, which have all of the unique classes in the shortest span of area of the map. This will help to reduce fuel, time, resources and focus on areas of concern, having all the resources, which he might want to study.

Wisp is built on a map based system, it is open sourced, and scientists can use this module by passing necessary formatted data to give visualisations in real time. The real fuel here is data, if the data is accurate, it will give good results, but it may happen that certain places doesn't have the data, in that case use of premium Google services may dominate than using foursquare free services for wisp application. The visualisation is done in real time and saves users from manually searching a place. Since foursquare is community maintained API, it may happen that certain labelling of data is wrong, as the users have freedom to add and manipulate data. We may use Google API's in future for its popularity and correctness. In conclusion, we just need to pass the necessary formatted data to get the visualisations in real time with the help of Wisp application. In this method we can cluster nearest unique nodes from different classes using JJCluster algorithm in Wisp application.

Acknowledgements

I am thankful to my father, Dr. Jadab Kumar Pal, Deputy Chief Executive, Indian Statistical Institute, Kolkata, for constant motivation and support for completion of this project. I dedicate this clustering algorithm to my brother Jisnoo Dev Pal. I also acknowledge Prof Shalabh Agarwal, former Head of the department of Computer Science, St. Xavier's College, Kolkata for his support. Finally, I am thankful to all my well-wishers who supported me in every part of this project.

References

- Pal, J.B. (2018). Wisp source code. Github. Source code in Python 3, <https://github.com/Jimut123/wisp> available on web. DOI: <https://zenodo.org/record/3254836#.XSBQFVmtbt1>.
- Pal, J.B. (2019). Wisp project web page. Github. <https://jimut123.github.io/projects/wisp.html> available on web.
- Sjöbergh J., Tanaka Y. (2014). Geospatial Digital Dashboard for Exploratory Visual Analytics. In: Kawtrakul A., Laurent D., Spyros N., Tanaka Y. (eds) Information Search, Integration, and Personalization. ISIP 2013. Communications in Computer and Information Science, vol 421. Springer, Cham.
- Beilschmidt C., Fober T., Mattig M., Seeger B. (2017). Quality Measures for Visual Point Clustering in Geospatial Mapping. In: Brosset D., Claramunt C., Li X., Wang T. (eds) Web and Wireless Geographical Information Systems. W2GIS 2017. Lecture Notes in Computer Science, vol 10181. Springer, Cham
- Wang, R., Hu, TL. & Chen, G. J. Comput. (2018). Sci. Technol. 33: 190. <https://doi.org/10.1007/s11390-017-1766-3>
- Wang, S. & Eick, C.F. Geoinformatica (2014). 18: 569. <https://doi.org/10.1007/s10707-013-0190-2>
- Lilford R., Kyobutungi C., Ndugwa R., Sartori, J., Watson, S.I., Siluzas, R., Kuffer, M., Hofer, T., Albuquerque, J.P.D., and Ezeh, A. (2019). Because space matters: conceptual framework to

- help distinguish slum from non-slum urban areas, BMJ Global Health 4:e001267.
- Tranos, E., Mack, E. (2019). A new opportunity for transport geography?, Journal of Transport Geography, Volume 76, Pages 232-234, ISSN 0966-6923,<https://doi.org/10.1016/j.jtrangeo.2018.08.003> available on web.
- Leao, S., Lieske, S., Conrow, L., Doig, J., Mann, V., and Pettit, C. (2019). Building a National-Longitudinal Geospatial Bicycling Data Collection from Crowdsourcing, Urban Science, vol. 1, pp. 23-23, <https://doi.org/10.3390/urbansci1030023>.
- Singleton, A., Longley, P. (2009). Geodemographics Visualisation and Social Networks in Applied Geography. Applied Geography 29(3) 89-298 <https://doi.org/10.1016/j.apgeog.2008.10.006>.
- Wang, W. Li, and Wang, F. (2017). Web-scale multidimensional visualization of geospatial big data to support climate sciences. Informatics, 4(3), 17.
- Kingston, R., Cavan, G., Butlin, T., Gill, S. and Lindley, S. (2014). Web-GIS tools for climate change adaptation planning in cities. In Handbook of Climate Change Adaptation: Climate Change Adaptation Technologies. Springer Nature , Berlin Heidelberg.
- Zhang, W., Cao, K., Liu, S., and Huang, B. (2016). A multi-objective optimization approach for health-care facility location-allocation problems in highly developed cities such as Hong Kong. Computers, Environment and Urban Systems, 59, 220-230
- Dong,G., Ma, J., Wei, R., and Haycox , J. (2019). Electric Vehicle Charging Point Placement Optimization by Exploiting Spatial Statistics and Maximal Coverage Location Models. Transportation Research Part D: Transport and Environment, 67, 77-88.
- Zlatanova, S., Döner, F. and Oosterom, P.V. (2011). Management and visualization of utility networks for local authorities: a 3D approach: In: Janssen, Macintosh, Scholl, Tamburis, Wimmer, Bruijn & Tan (Eds.), Electronic Government and Electronic Participation, Joint Proceedings of Ongoing Research and Projects of IFIP EGOV and ePart, Schriftenreihe Informatik 37, Trauner Verlag, pp. 459-474
- Bellman, R. (1956). On a Routing Problem, The Rand Corporation.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L. , and Stein C. (2009). Introduction to Algorithms, Third Edition (3rd ed.). The MIT Press.
- Dodge, S., Weibel, R. and Lautenschutz, A-K. (2008). Towards a Taxonomy of Movement Patterns. Journal of Information Visualization, Vol. 7, pages 240–252.
- Delort, J-Y. (2010). Hierarchical Cluster Visualization in Web Mapping Systems, April 26-30, Raleigh, NC, USA
- Phan, D., Xiao,L., Yeh, R., Hanrahan,P., and Winograd, T. (2005). Flow Map Layout. In Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization (INFOVIS '05). IEEE Computer Society, Washington, DC, USA, 29-. DOI: <https://doi.org/10.1109/INFOVIS.2005.13>
- Stachoň, Z., Šašinka, Č., Čeněk, J., Angsüsser, S., Kubíček, P., Štěrba, Z., and Bilíková, M. (2018). Effect of Size, Shape and Map Background in Cartographic Visualization: Experimental Study on Czech and Chinese Populations. ISPRS Int. J. Geo-Inf.7, 427.
- Zhu, Z. & Xu, C. (2017). Multimedia Systems 23: 53. <https://doi.org/10.1007/s00530-014-0426-5>
- Delmelle, E.M., Zhu, H., Tang, W., and Casas, I. (2014). A web-based geospatial toolkit for the monitoring of dengue fever, Applied Geography, Volume 52, Pages 144-152, ISSN 0143-6228, <https://doi.org/10.1016/j.apgeog.2014.05.007>.
- Etten, V., and Jacob., R. (2017). Package gdistance: Distances and Routes on Geographical Grids. Journal of Statistical Software, [S.l.], v. 76, Issue 13, p. 1 – 21. ISSN 1548-7660. Available at: <https://www.jstatsoft.org/v076/i13>. Date accessed: 05 july 2019. doi:<http://dx.doi.org/10.18637/jss.v076.i13>.
- D'Ulizia, A., Ferri, F., Formica, A., and Grifoni, P. (2009). Approximating geographical queries. Journal of computer science and technology 24(6): 1109–1124.
- Cai, G., Lee, K., and Lee, I. (2018). Mining semantic trajectory patterns from geo-tagged data. Journal of computer science and technology 33(4): 849–862. DOI <http://dx.doi.org/10.1007/s11390-018-1860-1>
- Dareshiri, S., Farnaghi, M. and Sahelgozin, M. (2017). A recommender geoportal for geospatial resource discovery and recommendation, Journal of Spatial Science, DOI: <http://dx.doi.org/10.1080/14498596.2017.1397559>.
- Zhao, P., Qin, K., Ye, X., Wang, Y. and Chen,Y. (2017). A trajectory clustering approach based on decision graph and data field for detecting hotspots, International Journal of Geographical Information Science, 31:6, 1101-1127, DOI: <http://dx.doi.org/10.1080/13658816.2016.1213845>.
- Germannis, T., Pettit, C.J., and Cartwright, W. (2004). Building a Three-dimensional Geospatial Virtual Environment on Computer Gaming Technology. Journal of Spatial Science. vol. 49, pp. 89-95, <http://dx.doi.org/10.1080/14498596.2004.9635008>
- Azri, S., Anton, F., Ujang, U., Mioc, D., and Rahman, A.A. (2015). Crisp Clustering Algorithm for 3D Geospatial Vector Data Quantization. In: Breunig M., Al-Door M., Butwiłowski E., Kuper P., Benner J., Haefele K. (eds) 3D Geoinformation Science. Lecture Notes in Geoinformation and Cartography. Springer, Cham.
- Story, R. (2014). Github.<https://github.com/python-visualization/folium> source code of the folium API.
- Santarcangelo, J., Aklson, A., Ahuja, R., Aghabozorgi, S., and Lin, P. (2018). IBM Data Science Professional Certificate, Coursera, <https://www.coursera.org/specializations/ibm-data-science-professional-certificate> available on the web.
- Stefanidis, T., Crooks, A.T. and Radzikowski, J. (2013). Harvesting Ambient Geospatial Information from Social Media Feeds, GeoJournal 78, (2): 319-338.
- Foursquare Developers, Foursquare website,Create Magical Real-World Moments for Your Users. <https://developer.foursquare.com/> available on web.
- Myer, T. (2010). Introduction to the Foursquare API. available on web <https://www.ibm.com/developerworks/library/os-foursquare/index.html>
- Harvesine Formula. (2018). Wikipedia article, https://en.wikipedia.org/wiki/Haversine_formula.