SWINBURNE UNIVERSITY OF TECHNOLOGY

SCHOOL OF SCIENCE, COMPUTING

AND ENGINEERING TECHNOLOGIES

===***===

MTH30006 – OPTIMISATION

RESEARCH PROJECT

K-medoid Method As An Integer Programming Problem And Application To Facility Location Problems

Name: Duy Khoa Pham

Student ID: 103515617

May - 2023

Table of Contents

1. Abstract

Clustering is a data analysis technique used to identify similar kinds of subgroups in a dataset. Data points (observations) from one subgroup (cluster) are likely similar and different from other subgroups. This paper illustrates the K-Medoids algorithm, one of the most well-known clustering methods. The research gives a deeper understanding of this clustering methodology by reading it through the lens of an integer programming issue, a mathematical framework for complex decision-making. In addition, the project will demonstrate the real-world implementation of the K-Medoids technique by clustering center point of accident across the area of Great Britain. Therefore, its main goal is to support minimizing accident in disaster management, especially for reducing the distances between accident locations and response units.

2. Introduction

2.1 Defining the Problem

Each accident that occurs in a city is associated with a certain city center. To improve the efficiency of ambulance services, it is vital to identify the center point for ambulance callers in these urban areas from which the majority of emergency calls in large urban regions originate. The goal is to discover a group of k points, as shown in Figure 1, where the cumulative total of distances between the points and their respective centers is minimized after assigning all remaining points to their nearest identified point (cluster center).

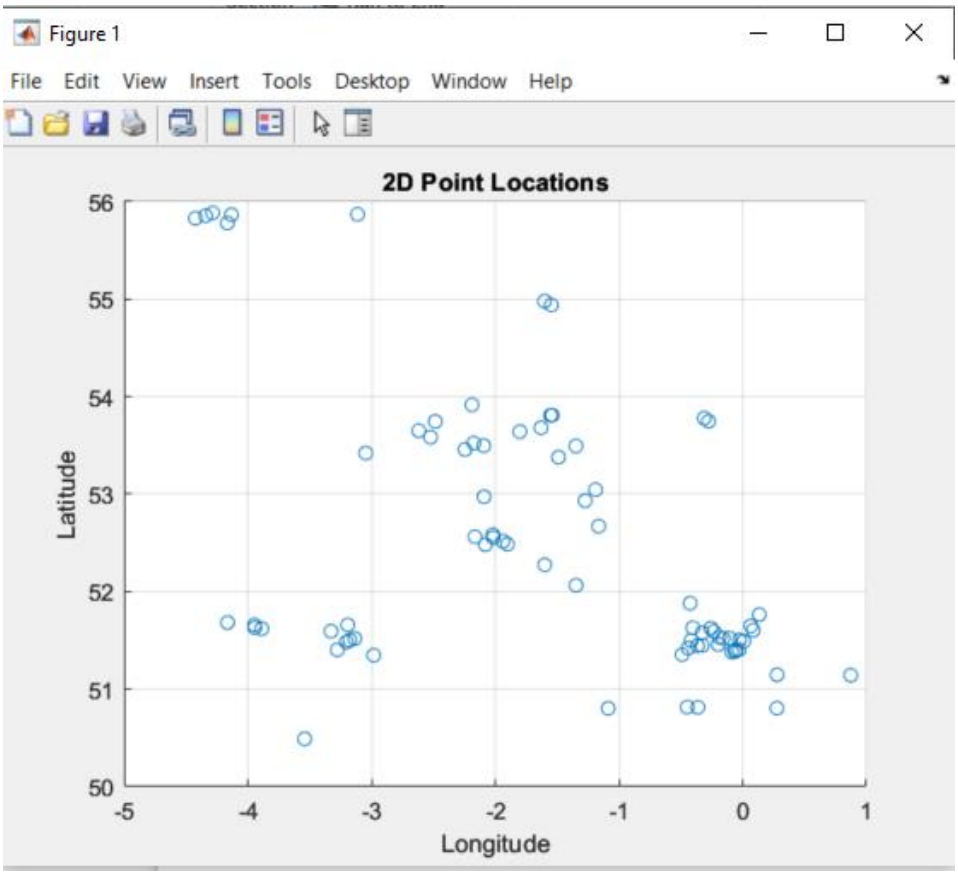

Figure 1. The data point where accident occurred in area of the Great Britain

2.2 K-Medoids Clustering

Our discussion begins with a quick overview of the K-Means algorithm, which is at the heart of unsupervised learning approaches that solve clustering challenges. This algorithm divides N observations into K clusters, with each observation connected to the cluster whose mean is closest, serving as the cluster's prototype. The K-Medoids algorithm is a clustering approach like K-Means. However, it deviates from the K-Means approach by electing an actual data point, instead of computing the centroid, to represent the cluster's center. This characteristic makes it a fitting choice for applications that require recognising a real-world representation of each cluster, such as selecting the best location to send ambulance callers.

2.3 Integer programming problem

Integer Programming is a subset of linear programming in which the decision variables must have integer values. It is a mathematical optimization problem where the objective function and the constraints are linear, and the decision variables are restricted to take integer values.

3. K-Medoid Method Formulated As An Integer Programming Problem

3.1 Mathematical Problem

In the context of location selection for optimizing resource allocation, integer programming is a good approach to handle multi-resource allocation and location challenges. Consider a set of demand points needed to be allocated to set up the emergency service location. These points can be formally represented as a set with n points in total and the distance matrix is:

$$D = \{d_{ij}\},\ i = 1, \dots, n,\ j = 1, \dots, n$$

This matrix is symmetric, and its main diagonal consists of zeros. The goal is to select k points as center points (clusters). The decision variables in this programming model are binary:

$$x_{ij} \in \{0, 1\},\ i = 1, \dots, n,\ j = 1, \dots, n$$

$$y_j \in \{0,1\},\ j = 1, \dots, n$$

Variable y_j is used to represented whether data point j is treated as a cluster point or not. It is set to 1 if data point j is a cluster point, and 0 otherwise.

Variable x_{ij} represents the assignment of point i to point j. It is set to 1 if point i is assigned to point j, and 0 otherwise.

The linear programming can then be written to minimize the sum of the distances between each point and its allocated center, subject to constraints such as each point being assigned to exactly one center and the number of centers being equal to k.

Such approaches are especially important in disaster management, where resources must be deployed wisely to maximize their effects. In the case of natural disasters, for example, it is critical to strategically place relief centers in order to reduce the distance that impacted individuals must travel to obtain these services (Bodaghi & Sukhorukova, 2018).

3.2 Integer Programming Problem Formulation

Linear programming is an excellent strategy for optimizing resource placement in multi-resource allocation and location problems such as disaster management (Bodaghi & Sukhorukova, 2018).

To formulate this mathematically, the objective function of the problem is:

$$Minimise \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij} \quad (1)$$

The constraints of the function subject to:

$$\sum_{j=1}^{n} x_{ij} = 1\ ,\ i = 1, \dots, n \quad (2)$$

$$x_{ij} \leq\ y_j,\ i,\ j = 1, \dots, n \quad (3)$$

$$\sum_{j=1}^{n} y_j = k \quad (4)$$

$$x_{ij},\ y_j\ \in \{0,1\},\ i,\ j = 1, \dots, n \quad (5)$$

3.3 Formulation Explanation

The objective function is aimed to minimize the total distance between each cluster with the points assigned to it. Constraints (2) ensure that each point (point i) is assigned to one cluster. This ensures that resources are allocated to each demand point. Constraints (3) ensure that the point i can only be assigned to the point j if this point (point j) is ensured to be a cluster. Constraint (4) ensures that exactly k points are selected as center point. This could be based on resource limitations or strategic requirements. Constraint (5) demonstrates that the decision variables $x_{ij}\ and\ y_j$ are binary. The formulation (1)-(5) is an integer programming problem.

Furthermore, taking into consideration the variance and covariance in the data may be vital for ensuring that the resource allocation is representative of the total demand, and this element may be accounted for using more advanced or modified linear programming models (Knapp, 1977).

4. Application in defining center location in Great Britain

```
-0.31099,53.7802
-3.11255,55.8706
-0.088533,51.3806
-0.326795,51.4476
-3.94653,51.6594
-0.447304,50.8153
-0.416988,51.4987
-1.53935,53.8108
-0.275438,53.7478
-2.08991,52.9739
-4.28391,55.8863
-3.94304,51.6317
-1.34684,53.4937
-2.07919,52.482
0.278042,51.1482
-2.98169,51.348
-2.16283,52.5623
-0.189638,51.535
-1.16379,52.6721
-3.88619,51.6198
-1.93748,52.5172
-0.32634,51.578
-1.18939,53.0464
-0.403184,51.6297
-2.17313,53.5239
-0.425243,51.8809
-1.54619,54.9408
-2.61786,53.6521
-1.60199,54.9812
-0.105094,51.5223
-1.59985,52.2746
-3.53969,50.4902
-1.55228,53.8087
0.084291,51.6036
-0.062879,51.3858
0.87316,51.143
-3.27546,51.404
-3.32742,51.5945
-0.23427,51.5926
```

Figure 2. Sample text data points in the area of the Great Britain

```
% Read points from the file
% dlmread is a function used to read a space or tab separated data
file.
data = dlmread('urbanGB.txt');
figure;
scatter(data(:,1), data(:,2));
xlabel('Longitude');
ylabel('Latitude');
title('2D Point Locations');
grid on;

% Number of Data Points
n = size(data, 1);

% Number of Medoids
k = 4;

% Construct Distance Matrix. Install pdist
D = pdist2(data, data);

% Objective Function Coefficients
f = [D(:); zeros(n, 1)]; % (1)

% Equality Constraints for Sum(x_ij) = 1 for each i
Aeq = [kron(eye(n), ones(1, n)), zeros(n, n)]; % (2)
beq = ones(n, 1);

% Inequality Constraints for x_ij <= y_j
A = [kron(eye(n), -eye(n)), kron(ones(n, 1), eye(n))]; % (3)
b = zeros(n * n, 1);

% Additional Equality Constraints for Sum(y_j) = k
Aeq = [Aeq; [zeros(1, n^2), ones(1, n)]]; % (4)
beq = [beq; k];

% Bounds
lb = zeros(n * n + n, 1); % (5)
ub = ones(n * n + n, 1); % (5)

% Integer Constraints
intcon = 1:(n * n + n);

% Solve the Integer Linear Program
options = optimoptions('intlinprog', 'Display', 'off'); % Hide solver
output
sol = intlinprog(f, intcon, A, b, Aeq, beq, lb, ub, options);
```

Figure 3. MATLAB implementation of K-Medoid Algorithm

4.1 Instruction for running the code

The script is run on MATLAB code. To smoothly implemented, the users need to install the **Statistics and Machine Learning Toolbox** from Add-Ons in MATLAB. In addition, a space or tab-separated data file named 'urbanGB.txt', which contains the geographical data points. Each row should represent a data point with two columns (longitude and latitude). Place this file in the current MATLAB working directory.

4.2 Implementation explanation

To begin with, the script (Figure 3) generates a scatter plot to visualize the initial distribution of points (Figure 2). On the x-axis, we have the longitude, and on the y-axis, we have the latitude. The script proceeds to define the number of desired center locations, denoted as k. In this case, k is set to 4, indicating that we are looking for four center locations. The function “pdist2” is used to compute Euclidean distances between each pair of observations in the data. Here it compute the distance between one point to another and store it in matrix D(77x77) which is shown below:

```
% Construct Distance Matrix. Install pdist
D = pdist2(data, data);
```

Figure 4. "pdist2" function for calculating Euclidean Distance

To find the optimal solution, the Integer Linear Programming is used. The code constructs an objective function with the aim of minimizing the total distance between the points and their respective center locations. Constraints are included to ensure that each point is assigned to one and only one center and that exactly k centers are selected.

However, if it is possible that this method might not be able to find the feasible solution, the code would uses an alternative buit-in MATLAB method called k-medois algorithm which is implemented through Partitioning Around Medoids (PAM) technique. According to George Seif's article on Towards Data Science, PAM is similar to the k-means algorithm, but with one important difference: instead of calculating the mean of the items in each cluster to determine its centroid, PAM chooses one of the actual data points within the cluster to represent it. When compared to k-means, this choice makes PAM more resistant to outliers. The algorithm begins by randomly selecting 'k' medoids and then assesses the effect of switching medoids with non-medoids in succeeding iterations. If the total dissimilarity between data points and its medoid decreases thanks to the swap, the swap will be accepted. The algorithm is convergence when there are no more advatageous swaps.

```
% Extract x_ij and y_j from the Solution Vector
if isempty(sol)
    [cluster_assignments, medoid_indices, ~] = kmedoids(data, k);
else
    x = reshape(sol(1:n*n), [n, n]);
    y = sol((n * n + 1):end);
    cluster_assignments = zeros(n, 1);
    for i = 1:n
        [~, j] = max(x(i, :));
        cluster_assignments(i) = j;
    end
    [~, medoid_indices] = find(y > 0.5);
    medoid_indices = data(medoid_indices, :);
end
```

Figure 5. K-Medoid Algorithm as Alternative Method

Finally, the code provides a second plot to display the data points along with its clusters to highlight for visualization which is shown in figure 6:

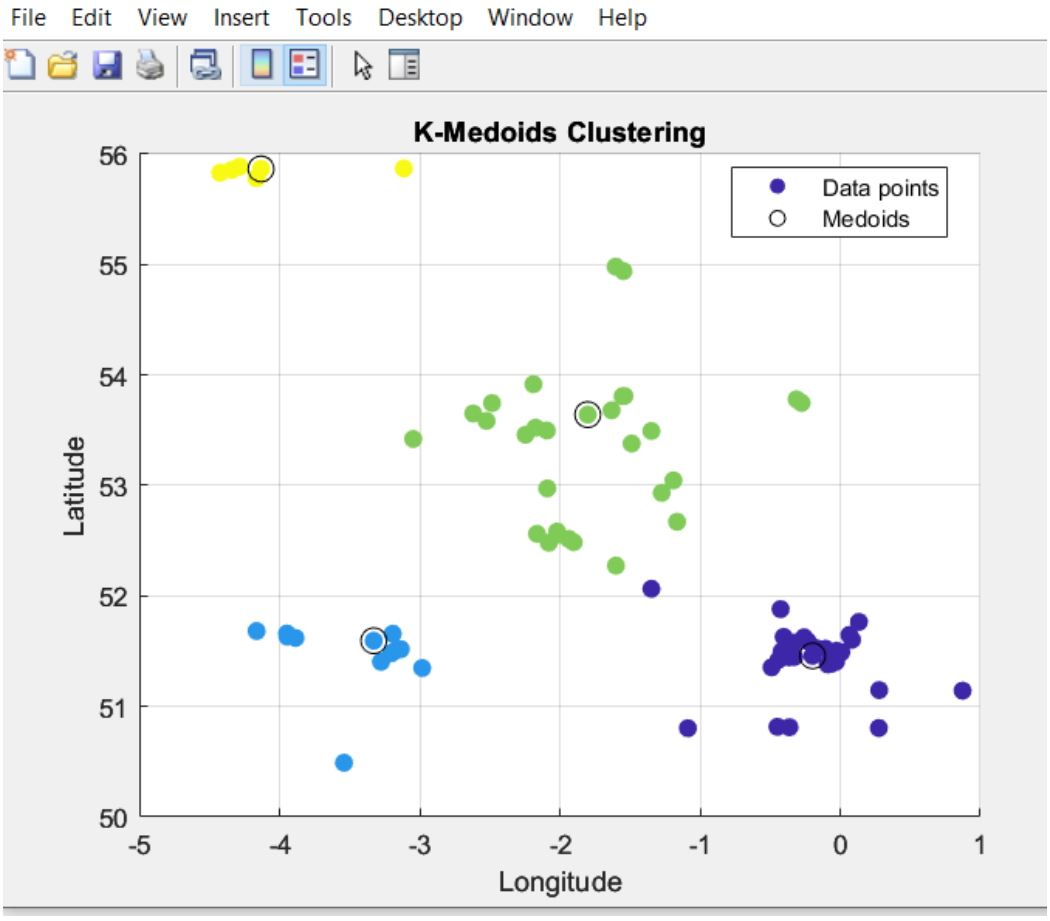

Figure 6. K-Medoids Clustering Visualization

5. Conclusion

In disaster management, and particularly in emergency response scenarios such as ambulance service allocation, it is vital to distribute resources efficiently and strategically in disaster management. This paper highlighted the K-Medoids algorithm as an effective clustering technique and explained its relation to integer programming, thus providing a mathematical foundation for decision-making in the optimization of resource allocation. On top of that, the K-Medoids algorithm's ability to select actual data points as cluster centers lends itself well to real-world applications such as determining appropriate ambulance service locations.

In the context of Great Britain, this methodology can be specifically applied to optimally allocate ambulance service centers in relation to accident-prone areas. Through intelligent allocation, it is plausible to not only improve response times but also allocate resources where they are most needed.

Future work could investigate smore complex constraints problem into the model. In addition, the author uses other clustering method and data mining techniques to handle the data and compare the accuracy and efficiency of each training methods like: Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Mean-Shift Clustering.

6. References

Knapp, T. R. (1977). Finite population unbiasedness and unbiasability of sample variances and sample covariance under multistage sampling. The American Statistician, 31(4), 165-168. https://doi.org/10.2307/2283635

Bodaghi, B., & Sukhorukova, N. (2018). Solving multi-resource allocation and location problems in disaster management through linear programming. arXiv preprint. arXiv:1812.01228. https://doi.org/10.48550/arXiv.1812.01228

Seif, G. (2018, September 17). Understanding K-means, K-means++ and K-medoids Clustering Algorithms. Towards Data Science. Retrieved from https://towardsdatascience.com/understanding-k-means-k-means-and-k-medoids-clustering-algorithms-ad9c9fbf47ca