UNIVERSITÉ PARIS-SUD

Report of Clustering

Yuru HE and Yuetian XU

10/03/2019

I-PROBLEM DESCRIPTION

Clustering is a statistical analysis method used to organize raw data into homogeneous silos. Within each cluster, the data is grouped according to a common characteristic. The scheduling tool is an algorithm that measures the proximity between each element based on defined criteria.

In the first part of TP, we need to understand the basic clustering ideas. After that, we use the bubble sorting to sort the randomly generated points, then calculate ‘c’ according to three different methods, then perform dynamic programming, calculate ‘C’ for clustering, and finally output the specific classification.

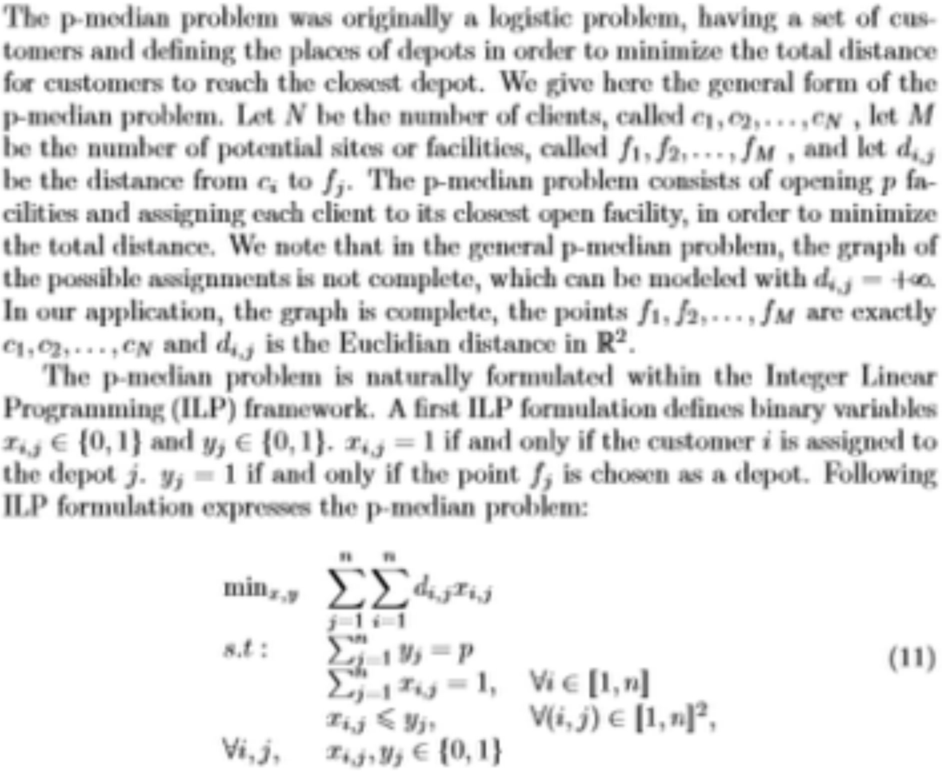
After implementing the basic functions, we use the k-median method as an example to optimize the whole algorithm.

II-STATE OF ART

In this part, our work related to three ways, in the state of art of the p-median, the p-center, and the k-median.

2.1 p-median

According to reference[1] provided by the teacher, we can learn that:

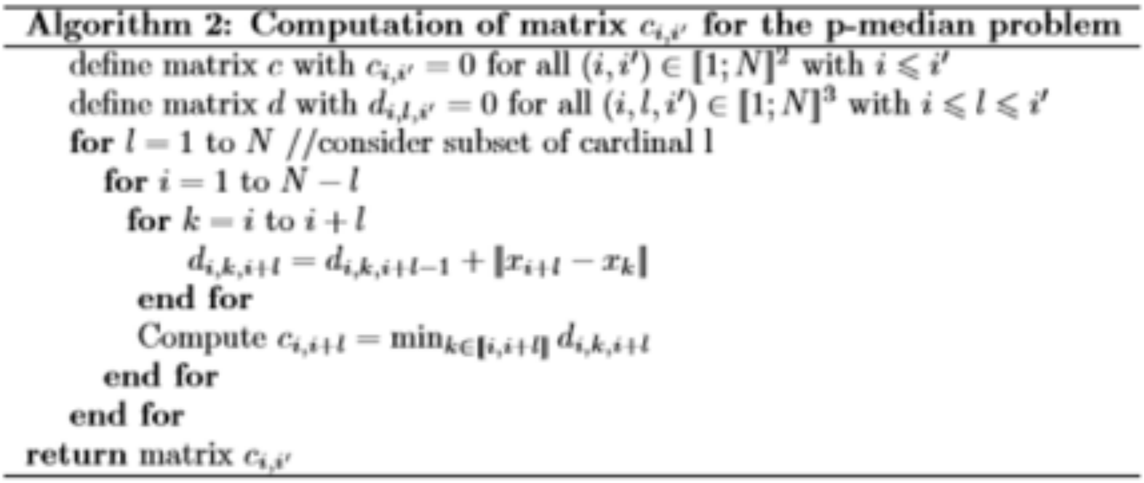


The p-median problem was originally a logistic problem, having a set of customers and defining the places of depots in order to minimize the total distance for customers to reach the closest depot. We give here the general form of the p-median problem. Let N be the number of clients, called $c_1, c_2, \dots, c_N$, let M be the number of potential sites or facilities, called $f_1, f_2, \dots, f_M$, and let $d_{i,j}$ be the distance from c_i to f_j. The p-median problem consists of opening p facilities and assigning each client to its closest open facility, in order to minimize the total distance. We note that in the general p-median problem, the graph of the possible assignments is not complete, which can be modeled with $d_{i,j} = +\infty$. In our application, the graph is complete, the points $f_1, f_2, \dots, f_M$ are exactly $c_1, c_2, \dots, c_N$ and $d_{i,j}$ is the Euclidian distance in $\mathbb{R}^2$.

The p-median problem is naturally formulated within the Integer Linear Programming (ILP) framework. A first ILP formulation defines binary variables $x_{i,j} \in \{0,1\}$ and $y_j \in \{0,1\}$. $x_{i,j} = 1$ if and only if the customer i is assigned to the depot j. $y_j = 1$ if and only if the point f_j is chosen as a depot. Following ILP formulation expresses the p-median problem:

$$\begin{array}{lll} \min_{x,y} & \sum_{j=1}^{n} \sum_{i=1}^{n} d_{i,j} x_{i,j} & \\ s.t: & \sum_{j=1}^{n} y_j = p & \\ & \sum_{j=1}^{n} x_{i,j} = 1, & \forall i \in [1,n] \\ & x_{i,j} \leqslant y_j, & \forall (i,j) \in [1,n]^2, \\ \forall i,j, & x_{i,j}, y_j \in \{0,1\} & \end{array} \qquad (11)$$

To achieve the calculation of c by the method of p-median, we can use the following ideas to implement:

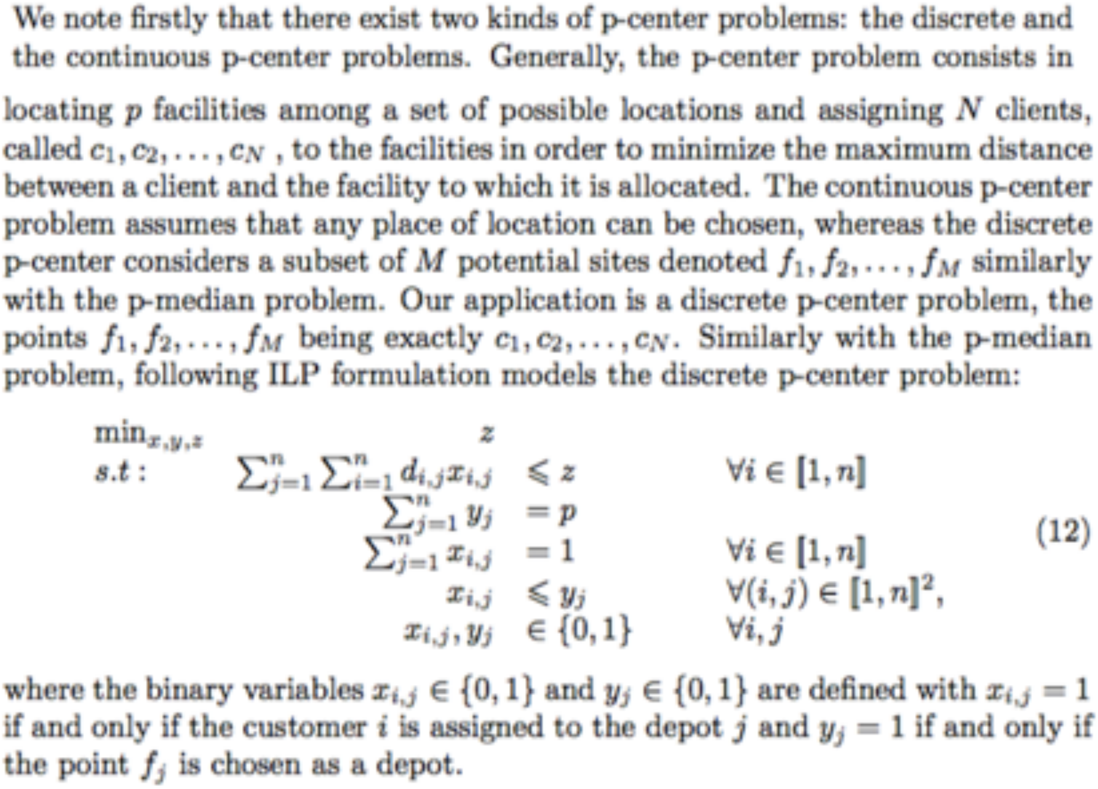


Algorithm 2: Computation of matrix $c_{i,i'}$ for the p-median problem

```
define matrix $c$ with $c_{i,i'} = 0$ for all $(i,i') \in [1;N]^2$ with $i \leqslant i'$
define matrix $d$ with $d_{i,l,i'} = 0$ for all $(i,l,i') \in [1;N]^3$ with $i \leqslant l \leqslant i'$
for $l = 1$ to $N$ //consider subset of cardinal l
  for $i = 1$ to $N - l$
    for $k = i$ to $i + l$
        $d_{i,k,i+l} = d_{i,k,i+l-1} + \|x_{i+l} - x_k\|$
    end for
    Compute $c_{i,i+l} = \min_{k \in [\![i,i+l]\!]} d_{i,k,i+l}$
  end for
end for
return matrix $c_{i,i'}$
```

2.2 p-center

According to reference[1] provided by the teacher, we can learn that:

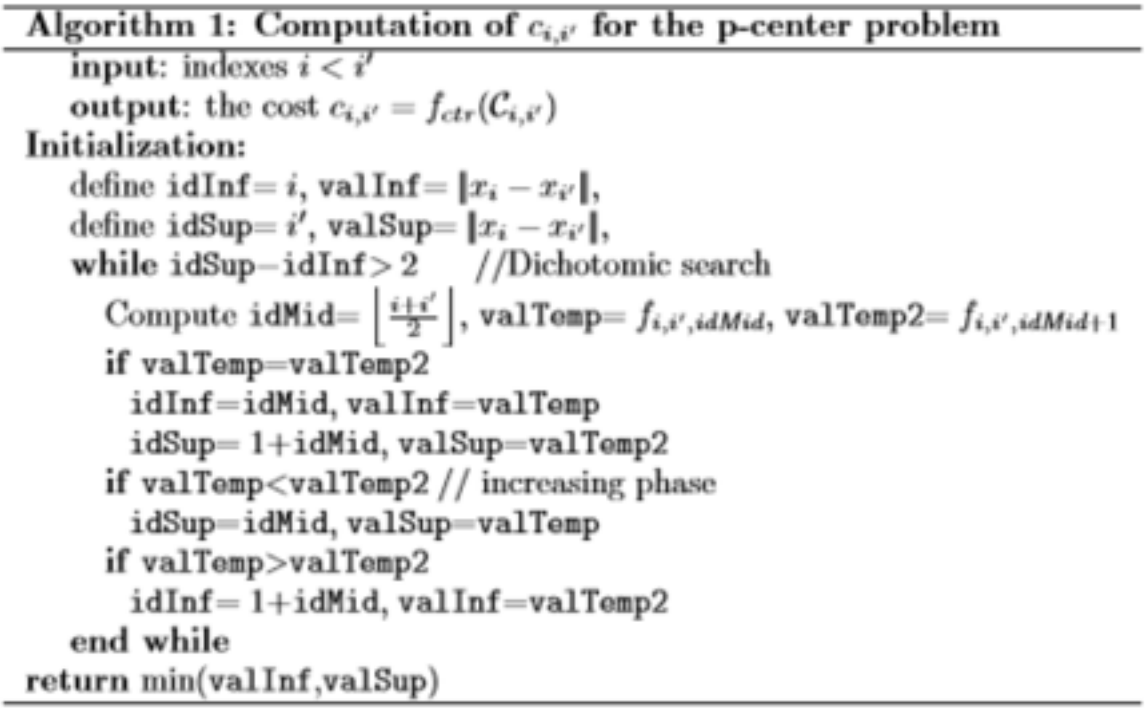


We note firstly that there exist two kinds of p-center problems: the discrete and the continuous p-center problems. Generally, the p-center problem consists in locating p facilities among a set of possible locations and assigning N clients, called $c_1, c_2, \ldots, c_N$, to the facilities in order to minimize the maximum distance between a client and the facility to which it is allocated. The continuous p-center problem assumes that any place of location can be chosen, whereas the discrete p-center considers a subset of M potential sites denoted $f_1, f_2, \ldots, f_M$ similarly with the p-median problem. Our application is a discrete p-center problem, the points $f_1, f_2, \ldots, f_M$ being exactly $c_1, c_2, \ldots, c_N$. Similarly with the p-median problem, following ILP formulation models the discrete p-center problem:

$$\begin{array}{llll}
\min_{x,y,z} & & z & \\
s.t: & \sum_{j=1}^{n}\sum_{i=1}^{n} d_{i,j} x_{i,j} & \leqslant z & \forall i \in [1,n] \\
 & \sum_{j=1}^{n} y_j & = p & \\
 & \sum_{j=1}^{n} x_{i,j} & = 1 & \forall i \in [1,n] \\
 & x_{i,j} & \leqslant y_j & \forall (i,j) \in [1,n]^2, \\
 & x_{i,j}, y_j & \in \{0,1\} & \forall i,j
\end{array} \qquad (12)$$

where the binary variables $x_{i,j} \in \{0,1\}$ and $y_j \in \{0,1\}$ are defined with $x_{i,j} = 1$ if and only if the customer i is assigned to the depot j and $y_j = 1$ if and only if the point f_j is chosen as a depot.

To achieve the calculation of c by the method of p-center, we can use the following ideas to implement:



Algorithm 1: Computation of $c_{i,i'}$ for the p-center problem

```
    input: indexes $i < i'$
    output: the cost $c_{i,i'} = f_{ctr}(\mathcal{C}_{i,i'})$
Initialization:
    define idInf= $i$, valInf= $\|x_i - x_{i'}\|$,
    define idSup= $i'$, valSup= $\|x_i - x_{i'}\|$,
    while idSup−idInf> 2      //Dichotomic search
      Compute idMid= $\left\lfloor \frac{i+i'}{2} \right\rfloor$, valTemp= $f_{i,i',idMid}$, valTemp2= $f_{i,i',idMid+1}$
      if valTemp=valTemp2
        idInf=idMid, valInf=valTemp
        idSup= 1+idMid, valSup=valTemp2
      if valTemp<valTemp2 // increasing phase
        idSup=idMid, valSup=valTemp
      if valTemp>valTemp2
        idInf= 1+idMid, valInf=valTemp2
    end while
return min(valInf,valSup)
```

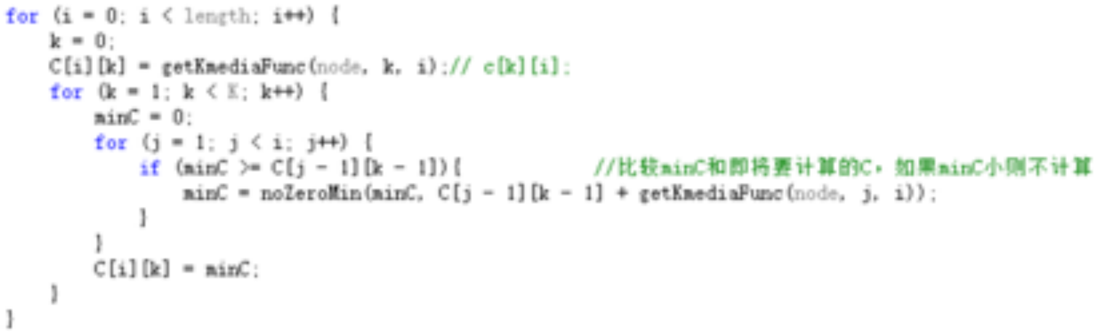
2.3 k-median

According to reference[2][3], in statistics and data mining, k-medians clustering is a cluster analysis algorithm. It is a variation of k-means clustering where instead of calculating the mean for each cluster to determine its centroid, one instead calculates the median. This has the effect of minimizing error over all clusters with respect to the 1-norm distance metric, as opposed to the squared 2-norm distance metric (which k-means does.)

This relates directly to the k-median problem with respect to the 1-Norm, which is the problem of finding k centers such that the clusters formed by them are the most compact. Formally, given a set of data points x, the k centers ci are to be chosen so as to minimize the sum of the distances from each x to the nearest ci.

The criterion function formulated in this way is sometimes a better criterion than that used in the k-means clustering algorithm, in which the sum of the squared distances is used. The sum of distances is widely used in applications such as facility location.

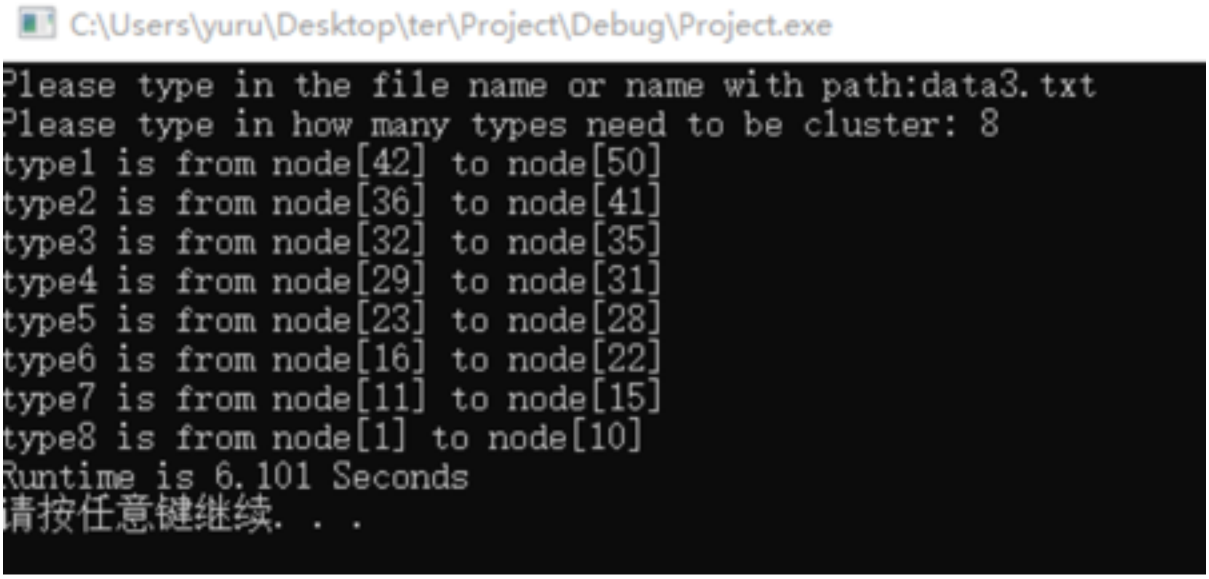
2.4 optimization



```
for (i = 0; i < length; i++) {
    k = 0;
    C[i][k] = getKmediaFunc(node, k, i);// c[k][i];
    for (k = 1; k < K; k++) {
        minC = 0;
        for (j = 1; j < i; j++) {
            if (minC >= C[j - 1][k - 1]){            //比较minC和即将要计算的C，如果minC小则不计算
                minC = noZeroMin(minC, C[j - 1][k - 1] + getKmediaFunc(node, j, i));
            }
        }
        C[i][k] = minC;
    }
}
```

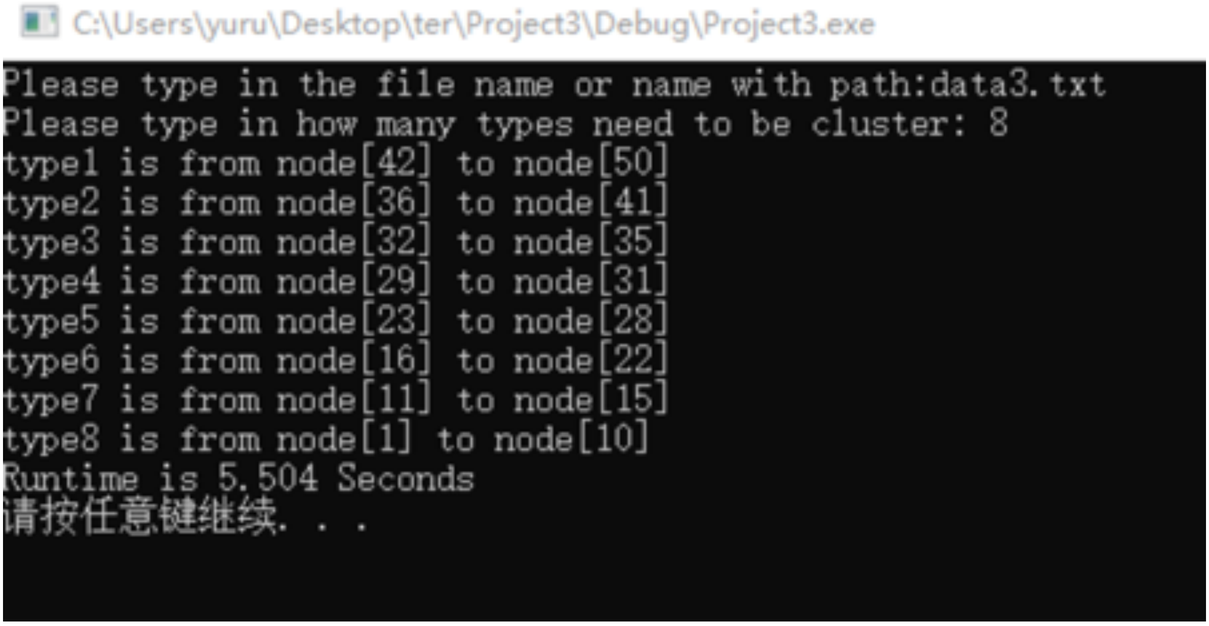
We thought of optimizing the algorithm by comparing ‘C’. In the traditional algorithm, we will calculate all the ‘c’ before we compare them. However, this process is relatively complicated, because the calculation of the k-median problem is relatively large. In order to save computation time, we can compare minC and C[j - 1][k - 1]. If the minC is smaller, this step of calculation of ‘c’ can be omitted.

The comparison of test time before and after optimization is as follows:



```
Please type in the file name or name with path:data3.txt
Please type in how many types need to be cluster: 8
type1 is from node[42] to node[50]
type2 is from node[36] to node[41]
type3 is from node[32] to node[35]
type4 is from node[29] to node[31]
type5 is from node[23] to node[28]
type6 is from node[16] to node[22]
type7 is from node[11] to node[15]
type8 is from node[1] to node[10]
Runtime is 6.101 Seconds
请按任意键继续. . .
```

(before the optimization)



```
Please type in the file name or name with path:data3.txt
Please type in how many types need to be cluster: 8
type1 is from node[42] to node[50]
type2 is from node[36] to node[41]
type3 is from node[32] to node[35]
type4 is from node[29] to node[31]
type5 is from node[23] to node[28]
type6 is from node[16] to node[22]
type7 is from node[11] to node[15]
type8 is from node[1] to node[10]
Runtime is 5.504 Seconds
请按任意键继续. . .
```

(after the optimization)

References

[1] Nicolas Dupin, El-Ghazali Talbi, Frank Nielsen, Clustering in a 2d Pareto Front: p-median and p-center are solvable in polynomial time.
[2] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data. Prentice-Hall, 1988.
[3] P. S. Bradley, O. L. Mangasarian, and W. N. Street, "Clustering via Concave Minimization," in Advances in Neural Information Processing Systems, vol. 9, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. Cambridge, Massachusetts: MIT Press, 1997, pp. 368–374.