The cluster-median problem

Problem description

Cluster analysis or clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. We will focus on a particular clustering problem, named the cluster-median problem.

Given a data matrix $A = (a_{ij}), i = 1, \ldots, m, j = 1, \ldots, n$ of m points and n variables, the goal is to group them in k clusters (k being a predefined parameter) such that the points in each cluster are similar. In the cluster-median problem the criteria for similarity is that the overall distance of all the points to the median of the clusters that they belong to is minimized. The median for a subset of points $\mathcal{I} \subseteq \{1, \ldots, m\}$ is defined as the nearest point to all points of $\mathcal{I}$:

$$r \text{ is the median of } \mathcal{I} \text{ if } \sum_{i \in \mathcal{I}} d_{ir} = \min_{j \in \mathcal{I}} \sum_{i \in \mathcal{I}} d_{ij},$$

where $D = (d_{ij}), i = 1, \ldots, m, j = 1, \ldots, m$ is the matrix of the distances for each pair of points. D is computed from matrix A, using any available distance: Euclidean distance, Mahalanobis distance, etc. For simplicity, the Euclidean distance can be considered in this assignment.

Formulation as an integer optimization problem

Even though only k clusters are needed, for simplicity m clusters will be considered in the formulation, such that $m - k$ of them will be empty. The cluster whose median is the element j will be denoted as "cluster-j" The variables of the formulation are, for each $i, j = 1, \ldots, m$,

$$x_{ij} = \begin{cases} 1 & \text{if element } i \text{ belongs to cluster-}j, \\ 0 & \text{otherwise.} \end{cases}$$

The goal is to make k and only k "correct" clusters (i.e., every element belongs to one and only one cluster-j), minimizing the overall distance between points and their clusters medians. Note that if cluster-j exists, then element j will be its median and it will belong to cluster-j; in other words, if cluster-j exists then $x_{jj} = 1$. Otherwise, the total distance would not be minimized. This particular result is used in the following formulation of the cluster-median problem to force than k clusters have to be built:

$$
\begin{array}{rll}
\min & \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij}x_{ij} & \text{[Distance of all points to their cluster medians]} \\
\text{subject to} & \displaystyle\sum_{j=1}^{m} x_{ij} = 1 \quad i = 1,\ldots,m & \text{[Every point belongs to one cluster]} \\
& \displaystyle\sum_{j=1}^{m} x_{jj} = k & \text{[Exactly } k \text{ clusters]} \\
& x_{jj} \geq x_{ij} \quad i,j = 1,\ldots,m & \text{[A point may belong to a cluster only if the cluster exists]} \\
& x_{ij} \in \{0,1\} &
\end{array}
$$

The last group of m^2 constraints can be formulated summing up for i: $mx_{jj} \geq \sum_{i=1}^{m} x_{ij} \quad j = 1,\ldots,m$. Now there are only m constraints, instead of m^2, but it is known that a lesser number of constraints does not guarantee it to be a better formulation.

The above combinatorial optimization problem is a difficult one, and its solution is only viable when m is not very large. There are some other approaches based on the above formulation that provide acceptable solutions with moderate computational resources. If interested, you can find them in "T.S. Arthanari, Y. Dodge, Mathematical Programming in Statistics, Wiley, 1993, pages 348–356".

Heuristic solution as a minimum spanning tree problem

Given the m points of previous sections, it is possible to partition them in k clusters of "similar points" by solving a minimum spanning tree problem. This procedure, explained below, does not solve the integer optimization problem for the cluster-median problem of Section 3.2, but it may provide a reasonable good heuristic solution.

The procedure is as follows. Consider the m points are nodes in a graph. Distances (or costs) of arcs connecting nodes are given by matrix $D = (d_{ij}), i = 1,\ldots,m, j = 1,\ldots,m$. Using some algorithm (e.g., Kruskal's or Prim's algorithm) build the minimum spanning tree for these m nodes and arc costs. Arcs between distant nodes will not be included in the tree. Therefore, the tree considers arcs for the closest or most similar nodes (points). Now we just have to remove $k-1$ arcs from the tree to obtain k disconnected subgraphs or clusters. Which are the $k-1$ candidates to be removed? Since the goal is to obtain clusters of similar points, the $k-1$ arcs with largest distances will be removed.

Problem to be solved

The tasks are:

- Implementing and solving using AMPL the integer optimization problem of Section 3.2.
- Applying the procedure of Section 3.3 using some implementation for minimum spanning trees.

You have to find some data matrix A for this assignment. The number of points should not be very large, otherwise the solution time for the cluster-median integer optimization formulation will be too large.

You must provide a report containing the following sections:

1. A cover page with the name of the two (and exactly two!) members of the group
2. A description of the data matrix A.
3. The AMPL .mod and .dat files.
4. The optimal solution obtained with AMPL.
5. A description of how the minimum spanning tree was computed (which software used, etc.)
6. The heuristic solution obtained with the minimum spanning tree procedure.
7. A comparison of the two solutions obtained, in terms of the objective function of the integer optimization problem. Optionally, you may provide any other additional criteria for comparison.
8. Any other observation or comment you may want to add, or any problem you had when performing the assignment.