CS4800 Algorithms & Data – Fall 2015
Course Project – due: Dec 2nd, 2015

1 Introduction

This project requires you to implement the clustering algorithm described in section 4.7 of the textbook (p. 157). Clustering is a very important concept and has many applications in data analysis.

Intuitively, the *single-link* clustering algorithm presented in the textbook groups a given set of objects $p_1, \ldots, p_n$ (images, documents, vectors or numbers, etc.) into a set of clusters based on objects' similarity. Similarity is defined by a distance function $d(p_i, p_j)$ which is "small" when the objects p_i, p_j are alike and "large" when p_i, p_j are "very different".

Specifically, given a number $k > 1$, the algorithm first calculates all pairwise distances $d(p_i, p_j)$ and then groups the objects into clusters $C_1, C_2, \ldots, C_k$ so that distances between objects in the same cluster are "small", while distances between objects in different clusters are "large". The meaning of words "similar", "large", and "small" depends on the context (hence the quotation marks); in our case, it will be the choice of the number k that will define it.

So how, for a given number $k > 1$, is this grouping accomplished? One builds a minimum spanning tree with weights given by the function d and removes $k - 1$ most expensive edges in it. The result is the a forest of k connected tress — the sought clusters $C_1, \ldots, C_k$. Accordingly, in this project you will need to implement Kruskal's algorithm and a version of the Union-Find structure needed for its operation.

Two implementation languages are allowed: Java or Python. **Except for linked lists and sorting subroutines, you are not allowed to use any ready made implementations of Union-Find or Kruskal's algorithm (if in doubt, please check with me)**. Your program should be able to interpret the data prepared for this project and produce output described below.

2 Data

You will work with the segment.arff dataset, available on Blackboard. This dataset is based on a set of images to which a number of low-level image processing operators were applied. As a result, each image is represented by a sequence of numerical descriptors whose meaning is explained in the header of the data file. Based purely on those numerical descriptors, the goal is to find clusters in the data which correspond to different types of objects (buildings, trees, sky etc).

The header portion of the file, explaining the meaning of each attribute, is kept for your information only. Your program should ignore those lines (i.e. skip them during reading — you should not delete them manually) before working on the data portion of the file.

The data portion of the file contains $N \approx 2,000$ instances in the following format:

$F_1^1, F_1^2, \ldots, F_1^D$, label_1

$F_2^1, F_2^2, \ldots, F_2^D$, label_2

$\vdots$

$F_N^1, F_N^2, \ldots, F_N^D$, label_N

where F_i^j, label_i $(i = 1, \ldots, N, j = 1, \ldots, D)$ are the values of the j^{th} feature and category for the i^{th} instance respectively.

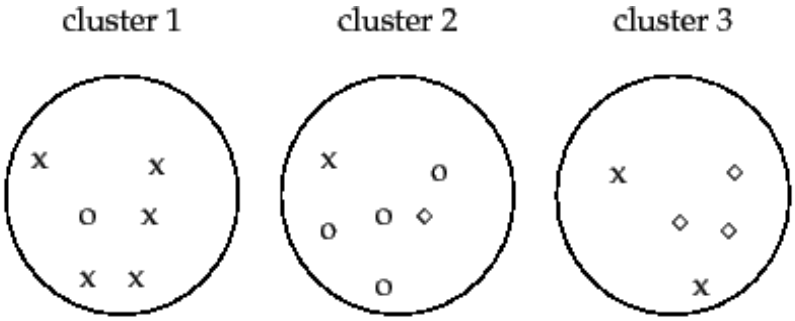
Each row of the file corresponds to a vertex in your graph and the edge weight between a pair of vertices (v_i, v_j) is defined by the Euclidean distance:

$$d(v_i, v_j) = \sqrt{\sum_{d=1}^{D} (F_i^d - F_j^d)^2}$$

Your clustering algorithm should ignore the labels. However, to judge how well it performed, you will need return to the labels later, in the report portion of the project (where you will calculate the *purity* of your clustering).

3 Cluster Evaluation

Knowing what the descriptor vectors actually represent (i.e., knowing the instances' labels) allows one to measure the *purity* of the obtained clusters. Purity is an *external* measure of clustering quality. To calculate it, each cluster is assigned to the category which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned instances and dividing by N. Figure 16.1 below illustrates how *purity* is calculated for the case of particular 3 clusters.



Source:
"Introduction to Information Retrieval", Christopher D. Manning, Prabhakar Raghavan ,and Hinrich Schutze, Cambridge University Press. 2008.

▶ **Figure 16.1** Purity as an external evaluation criterion for cluster quality. Majority class and number of members of the majority class for the three clusters are: x, 5 (cluster 1); o, 4 (cluster 2); and ◇, 3 (cluster 3). Purity is $(1/17) \times (5+4+3) \approx 0.71$.

Your program should take an integer $k > 1$ and the segment.arff data file as input and produce a k-clustering together with a calculation of the obtained clusters purity.

4 Deliverables

- Code implementing the described algorithm which takes as input a given number k (of clusters) and the data file in the format given above. The code should follow Kruskal's algorithm and utilize a Union-Find structure written by you. The only ready-made, nontrivial data structures/algorithms you can use are linked lists and sorting.
- A README file, with simple, clear instructions on how to compile and run your code.
- A report which contains
  1. A plot of *purity* (y-axis) as a function of k (x-axis) where $k = 1, 2, \ldots, N$.
  2. A table listing values of k and the corresponding *purities* for $k = 10, 20, 30, \ldots, N$.

3. One important question in clustering algorithms is the choice of "best" k. Discuss the pros and cons of using *purity* as the basis for choosing the optimal k.
4. **(Extra 10 points)** Given the *purities* calculated for $k = 1, 2, \ldots, N$, propose a method to choose the optimal k for your clustering algorithm.

5 How to turn in your code

- **Your program must run on CCIS machines in WVH 166 lab**
- **Zip all your files (code, README, written report, etc.) in a zip file named** $\{firstname\}\_\{lastname\}\_CS4800\_project.zip$ **and upload it to Blackboard**