CSCI 4372/6397: Data Clustering

Phase 4: External Validation

Submission Deadline: April 26 (23:59:59)
Due to end of semester, no late submissions will be allowed!

Objective: Implement external validation methods to compare automatically generated partitions with external ones.

In many applications, we have multiple partitions generated either by different clustering algorithms (e.g., k-means and another partitional clustering algorithm) or by the same clustering algorithm, but using different parameters (e.g., different initial centers) and we would like to know which of these partitions fits the data better. If we know the "true" cluster label of each point, we can use an external validity method to quantify the degree of similarity between an automatically generated partition (e.g., a partition produced by k-means) and the (external) partition implied by the true cluster labels.

In this phase, you will implement two external validation methods to compare automatically generated partitions with external partitions. External validation is often accomplished using an external validity index, a function that takes two partitions and possibly some additional parameters as input and gives a numerical value indicating the degree of match between the two partitions as output.

In this phase, you will implement the Rand and Jaccard external validity indices. These indices are described in many resources, see for example, the following book by Zaki and Meira Jr.:
http://www.dataminingbook.info/pmwiki.php/Main/BookDownload

Bonus for Undergraduate Students [5 points] / Mandatory for Graduate Students: In addition to Rand and Jaccard, implement the Fowlkes-Mallows index (for a description, refer to the aforementioned book).

In this phase, you will be supplied with a similar collection of data sets (to be made available on Blackboard), but the file format will be slightly different: the very first line will contain three integers (# points, # attributes + 1, # true clusters) and, at the end of each subsequent line, the true cluster label of the point (between 0 and # true clusters – 1, inclusive) will be given. For example, the first four lines of iris_bezdek will be

150 5 3
5.1 3.5 1.4 0.2 0
4.9 3.0 1.4 0.2 0
4.7 3.2 1.3 0.2 0

meaning that the data set contains 150 points, each point is 5 – 1 = 4 dimensional, and there are 3 true clusters (that is, each point belongs to clusters 0, 1, or 2). In the example given above, all three points belong to cluster 0.

For all of the aforementioned external indices, higher values are better. The way these indices are used is quite simple. For a given data set, we first run k-means R times, each time with the same K value (equal to the # true clusters specified in the data set file), but with a different set of randomly selected centers (as in phase 1). Each run will give us a partition (of the data set) and we will compute the external validity index (say, Rand index) between that partition and the true partition (that is implied by the true cluster labels). Finally, the run that gives the highest Rand index value is declared as the best run and the partition produced by this run is taken as the best partition of the data set.

Output: Tabulate the best values of each external validity index for each data set (for each data set, k-means should be run R = 100 times, each time with a different set of randomly selected centers). Normalize the data sets with min-max normalization prior to clustering.

Language: C, C++, or Java. You may only use the built-in facilities of these languages. In other words, you may not use any external libraries or APIs.

Documentation: Microsoft Excel or Apache OpenOffice Calc table(s) of each external validity index.

Submission: Submit your source code and output file(s) via Blackboard. Do **not** submit your files individually; pack them in a single archive (e.g., zip) and submit the archive file.