

# CSE 5800 Mining/Learning and the Internet—HW3

Due 6:30pm, Wed, Oct 21

Submit Server: `course=cse5800 , project=hw3`

1. Implement these clustering algorithms:
  - (a) K-means
  - (b) Bisecting K-means with largest cluster to split
  - (c) Bisecting K-means with least overall similarity to split
  - (d) Agglomerative Hierarchical Clustering with Intra-Cluster Similarity technique (IST)
  - (e) Agglomerative Hierarchical Clustering with Centroid Similarity technique (CST)
  - (f) Agglomerative Hierarchical Clustering with UPGMA
2. Each document is represented by a TF-IDF vector, each component is:  $tf_i \times idf_i$ , where:
  - $tf_i$  is the frequency of term  $i$  in the document divided by the total number of terms in the document and
  - $idf_i = \log(D/df_i)$ , where  $df_i$  is the number of documents that contain term  $i$  and  $D$  is the total number of documents
3. Allow these parameters:
  - (a) number of (final) clusters
  - (b) number of iterations (*ITER* in the paper) for Bisecting K-means
4. Measure performance using:
  - (a) Entropy
  - (b) F-measure
  - (c) Overall Similarity
  - (d) Jaccard Coefficient
5. Three data sets:
  - (a) toy data set on the course web site
  - (b) news data set on the course web site
  - (c) your own data set
6. A report (in pdf) that discusses the following:
  - (a) Sensitivity analysis of parameters: for the second data set,
    - i. vary each of the parameters (keeping the rest constant),
    - ii. calculate each performance measurement,
    - iii. plot performance vs. value of a parameter,
    - iv. discuss the value for each parameter that seems to achieve the highest performance.
  - (b) Compare the clustering algorithms: for the second data set,
    - i. plot performance vs. number of clusters for different algorithms
    - ii. discuss the relative performance of different algorithms
7. Implementation:
  - (a) use one of these programming languages: C, C++, Java, or LISP.
  - (b) input files: a file for the topic names; each topic has a file, which has multiple documents, each document starts with `--DocID--`
  - (c) three modules:
    - i. Preprocess: input the documents, output TF-IDF vectors
    - ii. Cluster: input the TF-IDF vectors; for each cluster, output DocID's in the cluster and the top 3 words in the centroid
    - iii. Evaluate: input DocID's, their class labels and cluster membership; output performance