PR3: Text Clustering

Published Date:
Apr. 11, 2018, 9:30 p.m.

Deadline Date:
Apr 28, 2018, 11:59 pm

Description:

****
This is an individual assignment.
****
Overview and Assignment Goals:

The objectives of this assignment are the following:

- Use an existing clustering algorithm (e.g., K-means).
- Implement a variation of the ***DBSCAN*** clustering algorithm.
- Deal with text data (news records) in document-term sparse matrix format.
- Design a proximity function for clusters of text data.
  - Think about the Curse of Dimensionality.
  - Use dimensionality reduction if needed.
- Think about best metrics for evaluating clustering solutions.

Detailed Description:

For the purposes of this assignment, you will implement a variation of the DBSCAN clustering algorithm. ***You may not use libraries for this portion of your assignment***. Additionally, you will gain experience with internal cluster evaluation metrics.

Input data (provided as training data) consists of 8580 text records in sparse format. No labels are provided.

Cluster the data such that you obtain > 100 clusters using an off-the-shelf clustering method (e.g., K-means). You can use existing libraries for this step.

Implement a variation of the DBSCAN clustering algorithm that takes as input clusters, rather than individual points. You must decide how to compute inter-cluster distances to figure out which points are core, border, or noise points. Think about the choices we talked about when we discussed agglomerative clustering.

For evaluation purposes (leaderboard ranking), we will use the Normalized Mutual Information Score (NMI), which is an external index metric for evaluating clustering solutions. Essentially, your task is to assign each of the instances in the input data to K clusters identified from 1 to K. All objects in the training data set must be assigned to a cluster. Thus, you can either assign all noise points to cluster K+1 or apply post-processing after DBSCAN and assign noise points to the closest cluster.

The leaderboard will report the NMI on 50% samples from the dataset.

The train.dat file is a simple CSR sparse matrix containing the features associated with different feature ids in the input file. It differs from previous train.dat files in that it does not contain labels as the first element in each row.

Some things to note:

- The public leaderboard shows results for 50% of randomly chosen test instances only. This is a standard practice in data mining challenges to avoid gaming of the system. The private leaderboard will be released after the deadline and evaluates all the entries in the data set.
- Each day, you can submit a prediction file up to 5 times.
- The final ranking will always be based on the last submission.
- format.dat shows an example file containing 8580 rows with random cluster assignments from 1 to K. Where K is the number of clusters that you detect.
- There are no test.dat files in this assignment.

Rules:

- This is an individual assignment. Discussion of broad level strategies is allowed but any copying of submission files and source codes will result in honor code violation.
- Feel free to use the programming language of your choice for this assignment.
- While you can use libraries and templates for dealing with input data you should implement your own *clusterDBSCAN* clustering algorithm.

Deliverables:

- Valid submissions to the Leader Board website: https://coe-cmp.sjsu.edu/clp/ (username is your MySJSU email, password is your MySJSU password).
- **Canvas Submission of source code and report:**
  - Create a folder called pr3_SJSU-ID
  - Include a 2-4 page, single-spaced report describing details regarding the steps you followed for developing the clustering solution for text data. The report should be in PDF format and the file should be called **report.pdf**. Be sure to include the following in the report:
    1. Name and SJSU ID.
    2. Rank & NMI for your submission (at the time of writing the report). If you chose not to see the leaderboard, state so.
    3. Your approach (pseudocode for clusterDBSCAN).
    4. Determine the radius *Eps* for *MinPts* varying from 3 to 21 in steps of 2 for the given dataset.
    5. Implement/Use your choice of internal evaluation metric and plot this metric on the y-axis for the clusters resulted with the *Eps* and *MinPts* in the steps above.
    6. Describe, any feature selection/reduction or custom proximity measure you used in this study.

Grading:

Grading for the Assignment will be split on your implementation (70%), report (20%) and ranking results (10%).

Files:

- On Canvas