CSE 4334/5334 – Data Mining

Fall 2014 - Project 3
Due: 11:59pm Central Time, Tuesday, Dec 2, 2014

REQUIREMENTS:

Read the following requirements carefully, and make sure you follow every rule. If you fail to meet some requirement, marks will be deducted accordingly.

- Submit **ONE zip file** that contains all source codes, 3rd party libraries and a single PDF project document. **(5 points)**

- Requirement on your project document:
  - The file format of the document is PDF;
  - No limitation on the number of pages, or the format of the document;
  - The document must clearly describe how you design and implement your ideas; **(5 points)**
  - The document must clearly state how to execute your program using command line; **(5 points)**
  - The document must provide execution screenshots of your program. **(3 points)**

- Requirement on your source code:
  - You have to implement the whole project by yourself;
  - 3rd party libraries can be used **ONLY** for reading TSV files, stop words removal, and stemming;
  - Your source code must pass compilation. Any non-executable submission is not acceptable. **Make sure your program compiles and runs on omega.uta.edu, before you submit**.
  - Your submission must EXCLUDE the input files that we provide to you; **(2 points)**
  - You can use **any language that can run on omega.uta.edu**, though I recommend Java and Python;
    - If you use Java, you may include your Eclipse or NetBean project folder. Remember to exclude the input files;
    - If you use python, then you only need to submit your source code.

PROBLEM SCENARIO:

BACKGROUND

Project 3 asks you to cluster jobs based on job descriptions and requirements.

TASKS

You are given *jobs.tsv*, a tab-separated file which has three columns: JobID, description, and requirements. Note that this is not the file used in Project1, nor the file used in Project 2. Your task is to design/implement a clustering algorithm to cluster the given jobs into a number of clusters. You can

determine which method to use and how many clusters to be generated.

More specific tasks include:
1) **(10 points)** read information from input files.
2) **(50 points)** implement your clustering program.
3) **(20 points)** print your cluster assignments to an output file named *output.tsv*. The output file should look like the following. An example file *sampleoutput.tsv* is given to you.

```
1020868          1
628097           3
… …
284009           3
628097           9
891097           1
…
```

Your output file *output.tsv* must contain exact the same number lines as *jobs.tsv*, each of which has two tab-separated fields (JobID, ClusterNo). The ClusterNo is the JobID's assigned cluster number.

We will use your *output.tsv* to assess how good your clustering is. Whichever clustering method you choose to implement, you should always have an evaluation metrics (e.g., SSE) to assess how good your clustering result is.

Your program must be executed by the following commands, and the arguments must be in the same order:

```
java your-main-class-name /path/to/data/file/directory/ /path/to/output.tsv
or
python your-script-file.py /path/to/data/file/directory/ /path/to/output.tsv
or
./a.out /path/to/data/file/directory/ /path/to/output.tsv
```

* */path/to/data/file/directory/* is the path (e.g., */home/john/data-mining/data/)* to the directory that has the input file: *jobs.tsv*.

* */path/to/output.tsv* is the path to the output file, e.g. */home/john/data-mining/output.tsv*