CS 565 – Fall 2018 – Project 1

Problems due 11:59PM, Monday October 22. .

Instructions:

1. You will find the datasets on Piazza under resources, named `wine.csv` and `churn.csv`.
2. Please use the directory name project1 while submitting the project via `gsubmit`. For details, refer to Piazza instructions.
3. The 3-page report on the project must be in `.pdf` format.
4. DO NOT submit the dataset along with your source code.
5. Source codes will be checked for plagiarism using automated tools.

You may use a programming language of your choice for the implementation. However, irrespective of the programming language used, your program must accept command line arguments in the specific format described below. The output of your program should be a .csv file with a specific format described below, for automated grading. Please refer to the Deliverables section for further details.

General overview. In this project you will implement `K-means` and `K-means++` and use it to analyze two datasets.The `Wine` dataset is smaller and has a very clear clustering structure. The `Churn` dataset is somewhat larger, has both categorical and numerical attributes. Your grade for this project will come from three components; the quality of your code (40 pts.), the analysis of the experiments on `Wine` (30 pts.) and the analysis on the `Churn` data (30 pts.). You will submit the source code for your implementations which we will run to test. Further, you will submit a pdf file of **no more than 3 pages** describing the experiments. Exact details of these are given at the end of this project description.

1 Wine experiments

Wine data description. In the `Wine` dataset you will find the chemical analysis of 178 different wines. Each line in the data corresponds to the analysis of one type of wine. There are 12 attributes corresponding to the following chemical components; Alcohol, Malic acid , Ash , Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Non-flavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, Proline. Each value is a real number.

Kmeans. You will implement the `K-means` algorithm. Your function should take 3 arguments `filename`, k and `init`.

- `filename` is the path to the source file. (Note, that you need to implement reading the data. There are prewritten functions available to read csv files for both Python and Java.)
- k is the number of input clusters
- `init` takes two strings ”random” and ”`k-means++`”. The default should be ”random”.

The `init` argument specializes how the algorithm is initialized. The ”random” option initializes the cluster centers by choosing randomly from the data points. This yields the vanilla `K-means` algorithm. If the input is ”`k-means++`” then the initial cluster centers should be chosen based on the heuristics in `K-means++` .

The output will be an array, the same length as X, containing the cluster id of each item.

Analysis of K-means experiments. In order to be able to run `K-means` you first need to settle on an appropriate **distance function**. For this experiment you should choose the best option among the L_1, L_2 and L_2^2 norms. You don't need to run experiments with the three different objective functions, but you need to explain your choice in your write-up.

As you may have noticed, the values of the attributes in this dataset are in different ranges. So that we don't compare apples to oranges often it makes sense to **normalize** the values so that you compare them on the same range. Explain your normalization process in your write-up.

Finally, you will need to **find** the optimal number k of clusters. In order to find this value of k, you may plot the value of the cost function over a range of values of k, for both `K-means` and `K-means++` separately. Describe in the write-up any insights you may discover during this process.

PCA. In this exercise, we explore a popular dimensionality reduction technique known as Principal Component Analysis. You are not required to implement the `PCA` algorithm. For Python, the scikit-learn module provides an implementation of `PCA`
`http://scikit-learn.org/stable/modules/`
`generated/sklearn.decomposition.PCA.html.`
Use the `fit transform` method of the `PCA` implementation from the scikit-learn module and **reduce the dimensions** of each wine to 2. You can also use implementations of `PCA` in Java. Feel free to post any good package resources on `PCA` to Piazza.

Plot each wine on a 2D **scatter plot** using the values of 2 latent attributes found in the previous step. Furthermore, **color** each data point in the scatter plot with a color corresponding to the wines cluster label from the previous exercise. Describe your observations on this plot in your write-up.

As a sanity check, run `K-means` and `K-means++` on the reduced dataset with the 2 latent variables. **Compare** whether the cluster assignments are the same as when you ran the algorithms on the original dataset. Comment in your write-up whether you find the outcome of this test surprising or not and why.

2 Customer Churn experiments

Customer churn data description. This dataset contains information on costumers of a large telecommunication company. There is information on a total of 7044 costumers and 20 attributes. The attribute values may be real as well as categorical. Notice that the first line in the data contains the attribute names. You can see this first line to learn more about the attributes. You'll need to cluster this dataset in to 2 clusters (i.e. the costumers who churn - leave the company - or stay).

Data analysis. For these experiments you will reuse your code.

Here the number of clusters is given - 2. However, the data has multiple type of values; categorical values as well as real values in various ranges. To get a meaningful clustering you first need to **preprocess** the data to obtain appropriate numerical values for each category. Find ways to process the data (e.g. turn categorical data into boolean vectors, normalize real values) for the clustering task. In your write-up explain in detail what alterations you did to your data.

The attributes in this dataset are well-interpretable for laypersons too. Describe any **pattern** you may see for the two clusters. For this data we in fact have some ground truth available - in file `churn_truth.csv`, information whether the customer did or did not churn. Compare this data to your clustering outcome and with help of this try to identify outliers (e.g. people that are misclassified). This may strengthen or weaken your confidence in the patterns you discovered. Elaborate on this as it seems appropriate.

Deliverables. Your code should be executable from command line with first argument as the path to the .csv file, second argument as number of clusters, and third argument as a string which can be either random or k-means++. For example, a python source code should be executable with the command:

```
python source.py </path/to/churn.csv> <k> <init>
```

On executing this command, your program should create a file named `output.csv` in the same directory. The format of `output.csv` is as follows; each line contains a single integer which is the cluster id of that wine/customer.

The source code should be self-sufficient (external library dependencies will be resolved by the grader). Source code that does not compile/execute as per the instructions or contains unresolved bugs will lose all points for the implementation. There are points reserved for the readability of the code.

We encourage you to ask questions on Piazza. However, please do not ask syntax.