Datastream Processing - Lab 2

February 6th, 2019

Deadline: February 13th, 23h59

Instructions: Prepare an archive including your source code and a report with your results. Please use either PDF or HTML format for your report. Send it to jeremie.sublime@isep.fr

A Initial steps

Pick a programming language of your choice (Python, Java, Matlab or R are good ideas) and load your detailed code for the regular K-Means algorithm. Your algorithm should be able to open a regular dataset with numerical data and to find K clusters, and the code should be detailed enough to modify the functions. If you did not do your homeworks of finding such code, it is strongly advised that you borrow some source code from one of your classmate.

B Online K-Means for datastreams

B.1 Adapting K-Means code

Duplicate the code of your original K-Means algorithm to prepare the code for the Online K-Means algorithm that you are going to implement. Then do the following modifications to your algorithm:

1. Add the following parameters and variables to your algorithm: η the window size parameter to learn on the fly, and t the number of examples already processed.

2. Create a new function to update the centroids on the fly when a new data arrives. You will have to use your new variables t and η and the course formula:

$$\mu_c = \mu_c + \eta_t(x_t - \mu_c)$$

3. Modify the K-Means algorithm to turn it into a 1-pass algorithm. For each pass, add a function or some code to display the distance between the current data and its centroid.

4. Optional: If the programming language you picked allows it, modify your function to open the dataset and turn it to line by line reading.

B.2 Application on the SHealth dataset

The SHealth (for Samsung Health) dataset contains the information registered about the phone owner physical activity. In the original dataset, the stream was updated hourly or after each new physical activity. For convenience purposes, in this exercise we use a modified version of the dataset that contains only daily summaries of the activity over the course of 2 years. This dataset will therefore be used as a finite daily stream.

Within this context, the application must build up to date clusters of its user's daily activity and provide a cluster classification every day. Based on this clustering-based classification, the applications proposes daily objectives to groups of similar users.

In this section, we are going to use the Online K-Means that we have just implemented to build such clusters.

1. Open the readme file for these data and take a look at the different attributes.
    - Justify which ones you would use for a datastream clustering.
    - Build a parser function to read this dataset and fetch the attributes to your algorithm in the right format. Remember that since you have the constraint to process the data line by line, you cannot standardize the data. Only normalization using expert knowledge will be possible. Carefully justify your choices.
    - If it was not fully handled by your parser function, build a custom distance function for this dataset. Possible operations on the attributes include but are not limited to: weighting some attributes or groups of attributes, merging attributes, discarding irrelevant attributes, using different known distances depending on the type of attribute. Carefully justify your choices.
2. Try to run your algorithms with different values for K and η. For each pair of values, display the evolution of the distance the data and the centers as the stream is processed. Comment on the resulting curves. In particular, you may highlight days that feature spikes in your distance function.
3. How can you use the information provided by the previous curves to better tune your parameters ?
4. Modify your code to create a new cluster when a data is to far from all existing centers, and to remove the clusters that did not get any data assigned for a while. Explain any difficulties encountered while tuning this new version of the algorithm. Comment.
5. Explain what the "cold start problem" is. Propose a modification of your centroids initialization to alleviate this issue and run again your algorithm. Comment on the new results.

C Datastream with coresets (bonus)

Propose an implementation of the coreset building procedure for the K-Means algorithm.