CSCI 620/Section 02/Mior, Introduction to Big Data, Spring 2235

Assignment 8 – Clustering

Description

In this assignment, we are going to implement the k-means algorithm over MongoDB using Stack Exchange data. We will use the Euclidean distance in which points will be formed by the view count and the score. You should only consider posts with the tags "**`command-line`**", "**`boot`**", "**`networking`**", "**`apt`**", and "**`drivers`**".

Your tasks

1. For each document, create a field called 'kmeansNorm'. This should contain an array in which the first position is the normalized view count and the second position is the normalized score. Let v be the value of a field and m and M the minimum and maximum values among all posts. The normalized value of v is computed as (v – m) / (M – m). (Normalization helps adjust for the different scales used in average view count and score.) **(10 points)**

2. Write a program which selects *k* random documents from each tag listed above *t* (*k* and *t* are inputs to your program). The two fields from the previous question should be inserted into new documents in a collection named centroids. Assign the centroid documents IDs from 1 to *k*. You will need to erase any previous documents in this collection if it already exists.

   (Hint: you can use $sample to select random documents.)
   (15 points)

3. Implement one step of the *k*-means algorithm by assigning a new field 'cluster' in each document with the tag *t* (a parameter) _id of the closest centroid. Then update the documents in the centroids collection with the new centroids.
   (20 points)

4. For each of the following tags listed at the beginning of the assignment, initialize the cluster centers and run k-means for up to 100 iterations (or until the clusters converge) starting with *k*=10 up to *k*=50 with a step of 5 using only posts with that tag. Plot the sum of squared errors vs the value of *k* for each of the tags (i.e. one plot per tag). Provide your code and the plots.
   (30 points)

(continued on next page)

5. For each of the previous tags, decide which one of the computed number of clusters is the best in each case. For each tag, present an example cluster of posts grouped together such that you can explain it.
   (25 points)