Assignment 3

Clustering using Apache Mahout

Due: 11:59 PM November 5, 2014

1 Introduction

In your previous assignment you completed supervised learning, where correct labels were used to learn from data. Most of the real world problems presents itself without correct labels. Learning about the structure from unlabelled data is called unsupervised learning. There are various techniques for doing this and in this assignment we will focus on clustering.

Clustering is finding the common groups in the given data. Most of the times clustering is used to learn the hidden relations in the data which may not be apparent. Clustering can be either "Hard", where every instance is clustered into only one cluster, or "Soft", where a probability for the instance belonging to each clustered is calculated. In this assignment you will learn both the techniques. You will be doing K-Means clustering for "Hard Clustering" and Fuzzy C-Means clustering for "Soft Clustering".

2 K-Means Clustering

In K-Means clustering you will minimize the overall sum of squared distance of every instance from the cluster centroid. K-Means is essentially minimizing the following equation

$$\sum_{l=1}^{K} \sum_{x_i \epsilon X_l} ||x_i - \mu_l||^2$$

where, K is the number of clusters, and $\mu_l$is the centroid in the lth cluster.

2.1 Dataset

You are provided a text file, which contains the full reviews of 10 different product. It is extracted from the amazon dataset. However you are not provided any further information other than the reviews.

Input file is present at following location:
Diadem (Accessible from any CS Public Machines / Vertica)
/home/o/class/cs129a/assignment3/input_pa3_random.txt
Akubra
/home/bigdata1/assignment3/input_pa3_random.txt
Deerstalker
/home/bigdata1/assignment3/input_pa3_random.txt

2.2 Implementation

You are suggested to use normalized TF/IDF weighted vectors to generate the input and cosine similarity as distance measure. For clustering of textual documents this combination has been known to give best results.

You may however experiement with other techniques and submit the result which gives the best output. Please explain in text file, the technique and implementation you have used. Please note that the quality of final result is dependent on the initial clusters. You should run your tests few times with random seed and make note of the scores. Apace Mahout can randomly pick the initial clusters based on random seed, however you may experiment with your own initial cluster centers.

2.3 Pre-Processing

As in all the textual analysis, some pre-processing will always boost your results. You should consider removing stop-words, although TF-IDF is insensitive to stop-words, this will save the computing time and the memory required. You should consider stemming, as in previous assignment. It may also be a advantageous to make the character-case uniform and remove the extra punctuations.

2.4 Submission

Please submit a text file where each line is a number from 1 to 10 which indicates the category of the product. Do not worry about the actual numbering, your score will be determined by techniques which is insenstive to nomenclature of the categories.

2.5 Evaluation

Your code will be evaulated, based on accuracy and F1 while comparing it with the actual categories. Do not worry about the actual group number. We know that your group number might not be same as our group number and thus we have the script to take care of it. Since you do not have the label, you cannot find the accuracy of your clustering but there are other ways to know if your clustering is doing good. When clustering is evaluated without the knowledge of correct label, it is called as Internal Evaluation of Clustering. See the section 3.3. Please read **this** and **this**. Also more information has been provided in the **Mahout Tutorial**

3 Fuzzy C-Means Clustering

As you may have noticed that a category for a product is not always clearly defined. You may have experienced while browsing ebay or Amazon that same product could sometime end up in more than one category. Fuzzy C-Means (FCM), gives probability assignment to each category and no one category is significantly large then the product could be considered in each of the categories where the probabilty is significant. While FCM is very similar to K-Means in design, its implementation differs in the sense that the centroid of a cluster is the mean of all the points, each of them weighted by their degree of belonging to the cluster. Consider the following equation from wikipedia on FCM.

$$c_k = \frac{\sum_x w_k(x)^m x}{\sum_x w_k(x)^m}$$

where, $w_k(x)$ is degree of belonging the a given cluster which is calcuated as

$$w_k(x) = \frac{1}{\sum_j \left(\frac{d(center_k, x)}{d(center_j, x)}\right)^{2/(m-1)}}$$

where, m is a paramter which determines level of cluster fuzziness. This determines how many categories you would be comfortable assigning to a product.

3.1 Implementation

Like in K-Means you should use the same vectors as input. However you should experiment with the value of "m", (level of fuzziness) for the best score. "m" can be supplied as argument while FCM. FCM is implemented as fkmeans in Apache Mahout. Kindly see the documentation provided online.

3.2 Submission

Kindly submit a text file, where each line represents the "soft" clusters for the given instance in the following format.

clusterid:probability, clusterid:probability, ...

where clusterid is a number from 1 to 10 and probability is the confidence number for the product category to be "clusterid".

Once again do not worry about the actual numbering of the product categories. We will take care of it. We will use a weighted formula for the prediction to comparing against the actual class category. For e.g 10*probSince you do not have the label, you cannot find the accuracy of your clustering but there are other ways to know if your clustering is doing good. Please read the links given in section 2.5

3.3 Extra Credit

For any kind of technique you implement to evaulate your clusters you will get some extra credit. We would recommend either **Adjusted Mutual Information**or **Dunn Index**. Mahout already has Dunn Index, so feel free to use the implementation and make it work for your code. Find the Dunn Index **code here**