CODE REVIEW

Assignment A8 [A7]

November 16, 2017

Code Author: Shabbir Hussain

Review Author: Vineet Trivedi

Commit Id: 47eb2e529d9d4d408d7bb33014df750552b51e9b

Report:

1. For convergence of KMeans Subtask 1 the report states that it takes 5 minutes for convergence per task in subtask 1. It would be a good idea to mention the number of iterations for convergence as well.

2. One of the requirements of the assignments is:
   "Observe whether:
   A song's loudness, length, or tempo predict its hotness
   A song's loudness, length, tempo, or hotness predict its combined hotness".
   The above requirement/observation was not reported.

3. System and algorithm design for agglomerative clustering is reported.
   However, cluster and performance reporting for the following tasks in agglomerative clustering is missing:
   a. Fuzzy Length
   b. Fuzzy Tempo
   c. Fuzzy hotness
   d. Combined hotness

4. As aforementioned reporting is missing the following requirement is not completely met:
   "Use each method to perform each clustering task. Compare the results, as well as the performance of the solutions".

5. Performance reporting for Subtask 2 is missing. Neither the number of iterations nor the time taken for clustering/convergence is reported.

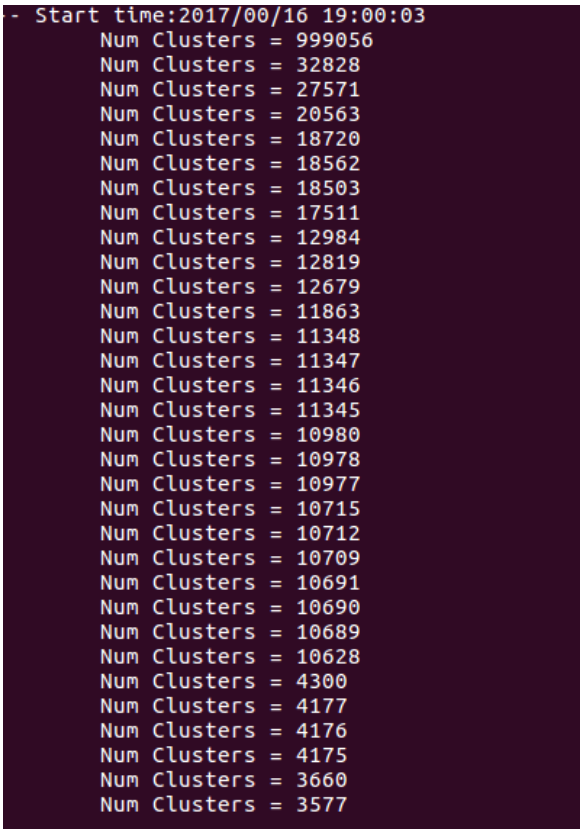
6. One of the requirements of the assignment is
   "Evaluate your solution using your a local machine and AWS".
   AWS evaluation and reporting is missing for both Subtasks.

Run:

1. It takes 54 iterations to converge for fuzzy length on the subset and 47 iterations on the full dataset which is a bit odd as convergence generally happens around 10-20 iterations for all other tasks in Shabbir's assignment as well as all tasks in my assignment.

2. On running the makefile it is observed that agglomerative clustering is performed only for fuzzy loudness.

3. According to my understanding agglomerative clustering combines two closest clusters at every iteration. Hence the total number of clusters should decrease by 1 at every iteration. In the given implementation, the total number of clusters decreases drastically and arbitrarily at every iteration:



4. The 3 files can only be accepted in '.gz' format. This is because the final part of the path has been hardcoded. It would be better for the main program to accept the files in an unzipped format and leave the compressing and decompressing part to the makefile.

```
object ArtistRecord {
  def loadCSV(sc: SparkContext, path: String): RDD[ArtistRecord] = {
    sc.textFile(path + "artist_terms.csv.gz")

object SimilarArtist {
  def loadCSV(sc: SparkContext, path: String): RDD[SimilarArtist] = {
    sc.textFile(path + "similar_artists.csv.gz")

object SongRecord {
  def loadCSV(sc: SparkContext, path: String): RDD[SongRecord] = {
    sc.textFile(path + "song_info.csv.gz")
```

Conclusion:

Overall the assignment is well done, the report is pristine, concise and informative. The use of heat maps is brilliant. However, some parts are missing mainly performance data of each task, performance on AWS and agglomerative clustering implementation of fuzzy length, fuzzy tempo, fuzzy hotness and combined hotness.