Locality Sensitive Hashing Lab Report

Brynna Mering and Risako Owan

January 21, 2015

CS324: Data Mining

LSH vs. Brute Force

In general, we observed that the brute force approach had a longer run time than LSH and that its average similarity fluctuated in fewer cases. The average of the Jaccard similarity for the brute force approach seemed to stay around 0.35 for most of the tests. On the other hand, the average similarity for locality sensitive hashing fluctuated from 0.091 to .48 and seemed to be affected by more factors. We think that the brute force approach is more stable here because it is not influenced by the number of rows in the band. It appears not to be influenced by the number of rows in the signature matrix either. We hypothesize that the average similarity becomes more accurate as the number of documents increases. However, increasing the number of documents also expanded our runtime considerably. The main factor that caused the average similarity in our brute force approach to stray away from 0.35 was adjusting the number of neighbors (k) used. We observed that when k increased, the average similarity for the brute force approach dropped significantly and probably became less accurate (Figure 1). The run time became less desirable as well. This is because less similar documents get figured into the total average. In addition, having more neighbors increases the amount of calculation for our program.

In our observations for the locality sensitive hashing, we found that changing the number of documents, neighbors, rows per band, and hash functions all influenced the average similarity and run time. First, as the number of documents grew, run time also increased significantly. We could not collect data for when the number of documents was over 5000 because the run time was too long. On the other hand, run time decreased considerably when the number of rows per band rose. This happens because as the number of row per band increases, the number of bands falls and thus the program does fewer comparisons. At the same time, the average similarity deviated farther away from the values calculated in the brute force approach (under the same conditions) (Figure 2). This variation occurs because fewer numbers of bands causes less candidate pairs. This leads to more documents being selected randomly to fulfill the k-neighbor requirement. When we compare the run times for the different quantities of hash functions, we found that fewer hash functions caused longer run times. When we focused on the influence of the number of neighbors, we saw that the similarity level falls and randomly chosen documents increases (Figure 3). Like the brute force approach, this is probably because larger values of k require less similar documents to be brought into consideration.

\*Please note that our run times may be inconsistent with each other or with other group observations because we ran our programs on multiple computers on multiple terminal windows simultaneously.

Graphs below (Chart attached in other word document):

Figure 1: Neighbors vs. Average Similarity

Figure 2: Rows per Band vs. Average Similarity

Figure 3: Neighbors vs. Randomly Chosen Docs