Data Mining: Learning from Large Data Sets - Fall Semester 2015

Approximate near-duplicate search using Locality Sensitive Hashing

by suyifan,zjia,weixi@student.ethz.ch

October 8, 2015

1 Notation

We have M videos $v_1, ..., v_M$, represented in the form of shingles, there are in total N shingles s_1, ..., s_N. The goal is to get the pairs of video such that their Jaccard distance is bigger than a threshold (in our case we chose this threshold to 0.9).

To achieve this, for each video v_i, we will construct $r \times b$ hash functions, partition them into b bands of width r. Each band b_j is hashed into N buckets. Any videos being hased into the same bucket at any band are candidates for the near duplicates.

2 Generating linear hash function for permutation

As disscussed in the lecture, we use a randomized linear function to get the permutation of a shingle:

$$\pi(s) = h_{a,b}(s) := a \times s + b \bmod N,$$

where a and b are chosen randomly from 0 to N.

3 Hashing of band

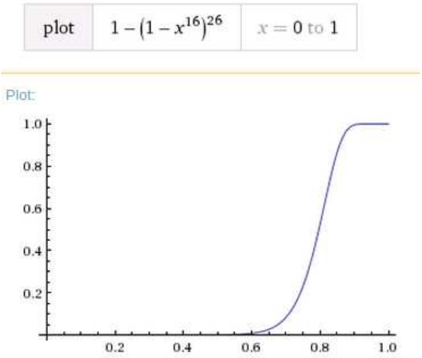
To hash a band, we also use a random linar hash function, for band $b_i = [s_1, ..., s_r]^T$:

$$\text{hash}(b_i) = \sum_{j=1}^{r} a_j \times s_j + b_i \bmod N$$

where a_j, b_i are chosen randomly from 0 to N.

4 Choice of band parameters

We chose r=16, b=26 as LSH parameter, as indicated in figure 1, the hit probability is a sigmod-shape curve, the threshold is around 0.8.



plot | $1-(1-x^{16})^{26}$ | $x = 0$ to 1

Figure 1. Probability of hit as a function of similarity

5 Results

With our code, we achieve 100% F_1 measure on the given test set, and the running time on a single machine is about 15 minutes.