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

melisr@student.ethz.ch
eholmer@student.ethz.ch
pmichal@student.ethz.ch

October 17, 2015

Approximate near-duplicate search using Locality Sensitive Hashing

We implemented the locality-sensitive hashing introduced in lectures.

Mapper

We choose parameters K, R. Then for each video V and each $k \in \{1, .., K\}$, define

$$h_k(V) = \min_{i \in V}[(a_k * i + b_k) \; mod \; n]$$

where a_k and b_k are integers drawn uniformly at random from $\{1, .., n\}$ and $\{0, .., n\}$ respectively. We stack these hashes into a vector $\mathbf{h}(V) = (h_k(V))_{k=1}^{K}$. Now, for each $l = 0, .., K/R - 1$ let $\mathbf{h_l}(V) = (h_k(V))_{k=l*R+1}^{(l+1)*R}$, we draw $\mathbf{a_l} \in \{1, .., n\}^R$ and $b_l \in \{0, .., n\}$ uniformly at random and define

$$\tilde{h}_l(V) = (\mathbf{a_l}^T \cdot \mathbf{h_l}(V) + b_l) \; mod \; n.$$

Finally, for each $l = 0, .., K/R - 1$, the mapper emits key-value pairs of the form

$$(l, \; \tilde{h}_l(V), \; (\tilde{h}_i(V))_{i=1}^{l-1}) \rightarrow (videoID, \; video \; content)$$

Reducer

Reducer receives key-value pairs grouped by their keys. If two inputs into the reducer have the same bandID and hash value i.e. $(l, \; \tilde{h}_l(V))$ then they are candidates for near duplicates, however we first check whether or not this pair has already been output based on the third item in key. If not, then we calculate the Jaccard distance to make sure we don't get any false positives and output the pair only if this is above 90%.