Big Data Computing

Homework 1

Pietro Spadaccino

Assignment 1

1. Let X, Y be two documents with $Jaccard(X, Y) \geq \theta_1$, then I can calculate the false negative rate *FN* as the probability of X, Y *not* being returned by the LSH system. We know that the probability that the i-th band of the signature matrix, denoted as X_i and Y_i, is the same between the two documents is equal to $Jaccard(X, Y)$, so we can write:

$$P\,(\,X_i = Y_i\,) = Jaccard(X, Y)^r \geq \theta_1^r$$

Then *FN* is the probability of X, Y not being returned by the LSH system, which is the probability of them disagreeing on all bands:

$$FN = P\,(\,X_i \neq Y_i,\ \forall i = 1, \ldots, b\,) = (1 - P\,(\,X_i = Y_i\,)\,)^b \leq (1 - \theta_1^r)^b \tag{1}$$

Now, let X, Y be two documents with $Jaccard(X, Y) < \theta_2$, then the false positive rate *FP* is equal to the probability of X, Y being returned by the LSH system. Like above, I can write:

$$P\,(\,X_i = Y_i\,) = Jaccard(X, Y)^r < \theta_2^r$$

From this we can compute the probability that they disagree on all bands:

$$P\,(\,X_i \neq Y_i,\ \forall i = 1, \ldots, b\,) = (1 - P\,(\,X_i = Y_i\,)\,)^b > (1 - \theta_2^r)^b$$

Finally *FP* is the probability of X, Y being returned by the LSH system, which is X, Y agree on at least one band:

$$FP = P\,(\,\exists i \,|\, X_i = Y_i\,) = 1 - P\,(\,X_i \neq Y_i,\ \forall i = 1, \ldots, b\,) < 1 - (1 - \theta_2^r)^b \tag{2}$$

Constraining *FN* and *FP* to be at most p_1 and p_2 respectively, we obtain the requested equations:

$$\begin{cases} FN \leq (1 - \theta_1^r)^b & \leq p_1 \\ FP < 1 - (1 - \theta_2^r)^b & \leq p_2 \end{cases}$$

The expressions of *FP, FN* that we found using (2) and (1) are upper bounds of their real value. An alternative approach can be to calculate their exact value using the probability of two docs being returned by the LSH system:

$$1 - (1 - s^r)^b \tag{3}$$

where s is the actual similarity between the two. Then I can express the false positive rate as:

$$FP = \int_0^{\theta_2} 1 - (1 - s^r)^b \, ds \leq p_2 \tag{4}$$

and the false negative rate as the area *above* the curve (3) from $s = \theta_1$ to $s = 1$, which can be expressed as the area of a rectangle having as width $1 - \theta_1$ and as height 1 subtracted by the area *below* the curve for $s \in [\theta_1, 1]$:

$$FN = (1 - \theta_1) \cdot 1 - \int_{\theta_1}^{1} 1 - (1 - s^r)^b \, ds \leq p_1 \tag{5}$$

2. We can find the minimum values of r, b such that they satisfy the constraints and they minimize $m = r \cdot b$ by plugging in the numerical values of $\theta_1, \theta_2, p_1, p_2$ and using a Python script (sent as attachment) that enumerates pairs $<r, b>$ with $r, b \in \{1, \ldots, M\}$ for some constant M. The script calculates the optimal value of r, b using both the upper bound formulation of (1), (2) and the exact formulation of (4), (5), and it runs in less than 5s, being optimized to not to enumerate all M^2 pairs of r, b values.

The obtained results were $m = 126$ with $r = 7$ and $b = 18$ using the exact formulation and $m = 76722$ with $r = 19$ and $b = 4038$ using the upper bound formulation.

Assignment 2

1. If x, y belong to the same cluster, the expected square of their distance is:

$$\mathbb{E}\left[||x-y||^2\right] = \mathbb{E}\left[\sum_{j\neq i}^{d}(x_j-y_j)^2\right] = \sum_{j\neq i}^{d}\mathbb{E}\left[(x_j-y_j)^2\right] \tag{6}$$

where we applied the linearity of expectation and we did not count the i-th dimension, since $x_i - y_i = 0$. We know that on the one-dimensional unit segment, the average squared distance between two random points is:

$$\mathbb{E}\left[(x_j-y_j)^2\right] = \int_0^1\int_0^1 (x_j-y_j)^2 dx\, dy = \frac{1}{6} \tag{7}$$

Using equations (6) and (7) I obtain:

$$\mathbb{E}\left[||x-y||^2\right] = \frac{d-1}{6} \tag{8}$$

2. Similarly, if x, y don't belong to the same cluster their expected square distance is:

$$\mathbb{E}\left[||x-y||^2\right] = |a-b|^2 + \sum_{j\neq i}^{d}\mathbb{E}\left[(x_j-y_j)^2\right] = |a-b|^2 + \frac{d-1}{6} \tag{9}$$

where $|a-b|^2$ is the deterministic squared distance on the i-th dimension.

3. From (8) and (9) we have that points belonging to the same cluster are expected to be closer together than points belonging to different clusters. So a clustering algorithm based on euclidean distance, like k-means++, can in principle identify clusters C_1, C_2 (for $|a-b|$ not too small). Nevertheless k-means++ can encounter problems if the distances between points differ substantially from the calculated expectation. If we apply Chernoff's bound to (8) and (9), we find out that the probability of this happening decreases exponentially as d grows. On the other hand, if d grows too much, the component $\frac{d-1}{6}$ may result much greater than $|a-b|^2 \leq 1$, making the distance between points of different clusters $\approx \frac{d-1}{6}$. In this case we would have the same expected distances between points belonging to the same and to different clusters, thus making distance-based clustering impossible in the original space.

4. Assuming that we know dimension i, we can cluster the points based on their projection onto the i-th dimension. We know indeed that a point x would have its i-th component $x_i = a$ or $x_i = b$, whether it belongs to C_1 or C_2, so we can use k-means++ with distance function $d(x, y) = |x_i - y_i|$ to perform the clusterization.

5. In general, if the deterministic component is no longer a canonical vector e_i but a generic direction in the space $v \in \mathbb{R}^d$, we cannot cluster the points using distance-based algorithms on the original space. The distance $\sum(x_j - y_j)^2$ between points is no longer discriminant, since it is no longer true that the expected euclidean distance between points belonging to the same cluster is lower than the distance between points not belonging to the same cluster. A solution can be finding the principal component using PCA and only then perform the clusterization through k-means.