CS60021: Scalable Data Mining
Practice Problems: Optimization

1. Suppose we have a (d_1, d_2, p_1, p_2)-*sensitive* family $\mathcal{F}$. We construct a new family $\mathcal{G}$ as follows. Each function $g \in \mathcal{G}$ is formed from a set of r independently chosen functions of $\mathcal{F}$, say $f_1, f_2, \ldots, f_r$ for some fixed value of r. Now, $g(x) = g(y)$ if and only if for all $i = 1, \ldots, r$, $f_i(x) = f_i(y)$. Which of the following statements is correct?

   a. $\mathcal{G}$ is an (d_1, d_2, p_1, p_2) family

   b. $\mathcal{G}$ is an (d_2, d_1, p_2, p_1) family

   c. $\mathcal{G}$ is an $(d_1, d_2, (1-p_1)^r, (1-p_2)^r)$ family

   d. $\mathcal{G}$ is an (d_1, d_2, p_1^r, p_2^r) family

2. Recall the *Band Construction* technique which we used to amplify the probabilities. In that we used k hash functions inside a band and L hash tables (i.e. L rows of bands). And the probability that two points x and y collide is given as $1-(1-p^k)^L$. Let us call this as *k-way AND* followed by *L-way OR*. Now what would the probability of collision be if we do *k-way OR* followed by *L-way AND*.

   (a) It doesn't matter, the probability remains same i.e. $1-(1-p^k)^L$

   (b) The probability changes to $(1-(1-p^k))^L$

   (c) The probability changes to $(1-(1-p)^L)^k$

   (d) The probability changes to $(1-(1-p)^k)^L$

3. Suppose you have a (k, L) LSH. As you increase k, while keeping L fixed, how is the recall likely to change?

   (a) Definitely increase.

   (b) Fluctuates.

   (c) Definitely decreases.

   (d) Increases and then decreases.

4. (*True/False*) If H is a LSH family, for any $h \in H$, the true nearest neighbor of the query is never present in the same bin as the query, and that is the reason we do multi-probing.

   (a) *True*

   (b) *False*

5. Given two unit vectors with angle distance 0.5π, the probability that a random hyperplane would separate these vectors is

(a) π

(b) $1/\pi$

(c) 0.5

(d) 1.

6. (*True/False*) Multiprobe LSH is likely to query more buckets in a single hash table than naive LSH.

   A. True

   B. False

   C. Depends on hash functions used

7. Your friend has created a "data-dependent LSH" where the LSH uses singular/eigen vectors for projecting ($v^t x$) and partitions them using < 0 or ≥ 0. Your friend claims that for a fixed query, the "probability of giving a nearest neighbor" is 80%. Which of the following statements is likely true.

   A. The above construction sounds believable.

   B. It is not believable and the claim about probability does not make sense.

8. (*True/False*) Suppose you have a (k, L) LSH. As you increase k, while keeping L fixed, how is the recall likely to change?

   A. Definitely increase.

   B. Fluctuates.

   C. Definitely decreases.

   D. Increases and then decreases.

9. Recall the "data dependent LSH" formulation:

$$\begin{aligned} \min \sum_{i,j} W_{ij} \|y_i - y_j\|^2 & \\ \text{sub. to. } \sum_i y_i y_i^T &= I \\ \sum_i y_i &= 0 \\ y_i &\in \{+1, -1\}^k \end{aligned}$$

   Suppose we removed the two constraints $\sum_i y_i y_i^T = I$ and $\sum_i y_i = 0$ (recall this 0 represents a vector). What do you think will happen to the resulting optimization problem?

   A. The solution can still be used for create LSH tables, possibly after some rounding, of equivalent performance.

   B. The vectors y_i that are output from the optimization problem, can still be used to create a LSH table, but of slightly worse performance.

   C. The output y_i vectors are all the same, and does not make sense to use in LSH as all k bits of the hash function will be the same for every point (i.e. this is like using only one hash function instead of k).

10. Spectral LSH, the relaxed version of above, produces only one LSH table of k-bits. If you are not satisfied with the recall from this single table, which of the following methods *cannot* be used to increase the recall of your LSH.

    A. You could solve the above problem, L times and use these for the different tables.

    B. When querying, you could query with the original query as well as use perturbations of it to query different tables.

    C. You could perturb individual bits of the k-bit hash-index to find alternate buckets to look into.

11. Suppose instead of solving the problem in lecture 13b, slide 8, you suggest to the professor that you could solve the following problem: find a k-means clustering of the data, with $k = 2$. If μ_1 and μ_2 are the two centers, then use the vector $\mu_1 - \mu_2$ for projection, and the LSH bit is created by using $\text{sign}(x^t(\mu_1 - \mu_2))$.

    A. This seems to be a reasonable way of creating a LSH.

    B. Doesn't make sense at all.

12. Consider the Multiprobe LSH setting with L hash-tables and k hash-functions in each table.

    You know this property of LSH "buckets that are one step away (i.e. only one hash value different from the k hash values of the query object) are more likely to contain objects that are close to query object than buckets that are two steps away."

    Using this property you decide on the perturbation vectors of the form $\Delta = \{-1, 0, +1\}^k$. And you probe only the buckets that are at most s steps away.

    Then, the number of buckets you probe is:

    A. $L \times k \times s$

    B. $L \times \sum_{n=1}^{s} k \times 2^n$

    C. $L \times \sum_{n=1}^{s} \binom{k}{n} \times 2^n$

    D. $L \times \sum_{n=1}^{s} k^n \times 2^n$