Topic 4: Locality sensitive hashing

CAIM: Cerca i Anàlisi d'Informació Massiva

Exercise list, Fall 2025

Basic comprehension questions

1. Explain in your own words the fundamental difference between a conventional cryptographic hash function (like MD5 or SHA-256) and a locality sensitive hash function. What is the primary goal of each, and why would you use one over the other for finding similar items?

Exercise 1

Suppose you are using the bit-vector LSH scheme explained in class. You have two pairs of documents:

- Pair A (Similar): Similarity $s_A = 0.9$
- Pair B (Dissimilar): Similarity $s_B = 0.3$

You decide to use an amplification strategy with $k = 4$ hash functions (stacking) and $m = 10$ repetitions (hash tables).

a) For a single hash table (using k=4), what is the probability that Pair A will collide? What is the probability for Pair B?

b) Using the full scheme (k=4, m=10), what is the overall probability that Pair A will be identified as a candidate pair (i.e., collides in at least one of the m tables)?

c) What is the overall probability for Pair B?

d) Briefly explain how these calculations demonstrate the "amplification of the gap".

Exercise 2

For simhashing, the probability of collision for two vectors x and y using a random hyperplane hash is given by $P[h_w(x) = h_w(y)] = 1 - \frac{\theta}{\pi}$, where θ is the angle between the vectors.

a) If two documents have a cosine similarity of 0.8, what is the angle θ between them in radians? (Recall that $cos(\theta) = \frac{x*y}{||x||*||y||}$).

b) What is the probability that these two documents will collide using a single random hyperplane hash?

c) If two other documents are perfectly orthogonal (cosine similarity of 0), what is their probability of collision? Does this make intuitive sense?

d) Why is it generally a good idea to restrict the random vector w to have components of only +1 or -1 in a practical implementation of simhashing?

Exercise 3

Imagine you are tasked with finding similar grayscale images. Each image is represented as a vector of 100 pixels, where each pixel has an intensity value from 0 to 255 (i.e., $d = 100, M = 256$). The notes suggest converting these integer vectors into long bit vectors by using a unary representation for each pixel value.

a) Using this scheme, what would be the total length of the resulting bit vector for a single image?

b) If two images, A and B, are at 500 Manhattan distance, what would be the similarity $s(A, B)$ in the transformed bit-vector space?

c) Briefly discuss the potential memory or computational drawbacks of this unary representation scheme, especially if M (the maximum pixel value) were much larger.

Exercise 4

You are designing a near-duplicate detection system for a database of n = 1 billion documents. You are using an LSH scheme with stacking and repetition (k and m).

a) If you choose a very small k (e.g., k=2), what is the likely consequence for the performance of your system at query time? Will the search be fast or slow? Why?

b) If you choose a very large k (e.g., k=32), what happens to the number of false negatives (i.e., similar pairs that you fail to detect)?

c) If you increase m (the number of hash tables), how does this affect the probability of finding a true similar pair (recall)? How does it affect the total query time?

d) Based on your answers, describe the fundamental trade-off between accuracy (finding all true near-duplicates) and speed in an LSH system.

Exercise 5

In an LSH system we use $k = 5$, $m = 10$. We want to get the neighbors of some item i_1 that are at least 0.9-similar to it.

1. If an item i_2 has similarity 0.9 to i_1, what is the probability that i_2 is in the candidate set?

2. If an item i_3 has similarity 0.45 to i_1, what is the probability that i_3 is in the candidate set?

3. Given the set of items that we would like to retrieve (0.9 similar at least), and those that we definitely want to exclude (0.45 similar at most) what is the recall, precision and f1-measure of this nearest-neighbor query? One cannot give exact figures here, so think what "the formula'' for all this would be.

   In the above, say that items in the grey zone between 0.9 and 0.45 do not count as false positives or as false negatives (this is new!).

4. Can you estimate the size of the candidate set? (The answer is NO, unless you suppose things about the distribution of similarities).

5. (Harder... for the braver) OK, let us suppose. Suppose that the probability that an item in the database has similarity s to i_1 decreases exponentially fast as s tends to 1. In particular i_1 has no copies, and 50% of the documents have similarity less than 0.9 with i_1. Can you now estimate the date of the candidate set? (Hint: First estimate how many documents have similarity above or below some given s. For that you need to estimate the parameters of a suitable exponential function of s ... an integral shows up...).

Exercise 6

Now let us look at it in the reverse direction. You probably need a numeric solver here.

1. We want that objects with similarity 0.9 appear as candidates 90% of the times or more, and that objects with similarity 0.7 appear as candidates 10% of the times at most. What values of k and m should we use?

   Of course, there are many pairs k, m that might work. Since the work performed by the algorithm is $O(km)$ we would like to choose a pair that, approximately, minimizes the product km.

2. We are now more strict. We want the items that are ≥ 0.99 similar to be recalled with probability 99% and we want items that are < 0.98 similar to be recalled only with probability 1%. What k, m should we use?