

Weekplan: Approximate Near Neighbor

Philip Bille Inge Li Gørtz

References and Reading

- [1] Notes by Aleksandar Nikolov
- [2] Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, Andoni and Indyk, Communications of the ACM, January 2008.

We recommend reading [1] in detail and [2] section 1-3.

Probability theory cheat-sheet

Markov's inequality: For Y being a positive-valued random variable,

$$P[Y \geq t] \leq \frac{\mathbb{E}[Y]}{t}.$$

Exercises

1 Hamming Distance Let P be a set of n bit vectors each of length d . Give a data structure (for Hamming Distance) for which INSERT can be implemented in $O(1)$ time and NEARESTNEIGHBOR in $O(nd)$ time.

2 LSH Hamming Distance Let T_1, T_2 be hash tables of size 5 with hash functions $h_1(x) = (3x + 4) \bmod 5$ and $h_2(x) = (7x + 2) \bmod 5$. Let $g_1(x) = x_1 x_4 x_8$ and $g_2(x) = x_1 x_7 x_7$.

Insert the bit strings $x = 10110011$, $y = 10001101$, $z = 00110010$, $u = 01001010$, $v = 01001000$ and draw the hash tables. Compute the result of ApxNearNeighbor(10111010).

3 c -Approximate Closest Pair under Hamming Distance Assume you have a data structure bit vectors for which INSERT and APXNEARESTNEIGHBOR run in time $T(n)$. The distance metric is the Hamming distance, n is the number of bit strings in the data structure, and APXNEARESTNEIGHBOR(x) returns a point no more than $c \cdot \min_{z \in P} d(x, z)$ away from x .

Give an algorithm that given a set P of n bit vectors each of length d finds a pair x, y of distinct strings in P in time $O(T(n) \cdot n + dn)$ such that $d(x, y) \leq c \cdot \min_{u, v \in P, u \neq v} d(u, v)$.

4 Hamming Distance Analysis From the proof of Claim 1 on the slides: Prove that $Lp_1^k \geq 2$.
Hint: Recall that $k = \lg n / \lg(1/p_2)$.

5 Hamming Distance Analysis 2 In this exercise we will analyse the LSH scheme for Hamming Distance. Recall that in a query we stop after we have checked $6L + 1$ strings. Let $F = \{y \in P : d(x, y) > cr\}$ (strings far from x) and let z^* be a fixed string with $d(x, z^*) \leq r$. We say that y collides with x if $g_j(x) = g_j(y)$ for some $j \in \{1, \dots, L\}$.

5.1 Explain why it is enough to prove that the following two properties hold:

1. the number of strings in F that collides with x at most $6L$.
2. z^* collides with x .

5.2 Let y be a string in F . Prove that $P[y \text{ collides with } x \text{ in } T_j] \leq 1/n$. *Hint:* Recall that $k = \log n / \log(1/p_2)$.

5.3 Let $X_{y,j} = 1$ if y collides with x in T_j and 0 otherwise, and let $X = \sum_{y \in F} \sum_{j=1}^L X_{y,j}$. Prove that $E[X] \leq L$.

5.4 Use Markov's inequality to show that $P[X > 6L] < 1/6$.

5.5 Prove that if there exists a string z^* in P with $d(x, z^*) \leq r$ then with probability at least $2/3$ we will return some y in P for which $d(x, y) \leq cr$.

6 Jaccard distance and Sim Hash The *Jaccard similarity* of two sets is defined as $\text{JSIM}(A, B) = \frac{|A \cap B|}{|A \cup B|}$. In *Min-Hash* you pick a random permutation π of the elements in the universe and let $h(A) = \min_{a \in A} \pi(a)$.

6.1 Let $S_1 = \{a, e\}, S_2 = \{b\}, S_3 = \{a, c, e\}, S_4 = \{b, d, e\}$. Compute the Jaccard similarity of each pair of sets.

6.2 Let S_1, S_2, S_3, S_4 be as above and let the random permutation be (b, d, e, a, c) , i.e., $\pi(a) = 4, \pi(b) = 1$, etc. Compute the min-hash value of each of the sets.

6.3 Prove that the probability that the min-hash of two sets is the same is equal to the Jaccard similarity of the two sets, i.e., that $P[h(A) = h(B)] = \frac{|A \cap B|}{|A \cup B|}$.

6.4 The Jaccard distance is defined as $d_J(a, b) = 1 - \text{JSIM}(A, B)$. Show that the Jaccard distance is a metric. That is, show that:

1. $d_J(A, B) \geq 0$ for all sets A and B ,
2. $d_J(A, B) = 0$ if and only if $A = B$,
3. $d_J(A, B) = d_J(B, A)$,
4. $d_J(A, B) \leq d_J(A, C) + d_J(C, B)$ for all sets A, B and C .

Hint: For 4. use $P[h(A) = h(B)] = \frac{|A \cap B|}{|A \cup B|}$.