Problems 1, 2, and 3 require you to have read the lecture notes or watched the lecture, but should be doable.

Problem 4 is important to have seen and attempted: you will go through it during the tutorial (but it is worth giving it some thought before). Problem 5 is recommended, and while the analysis to get the final bound on C is technical and somewhat annoying, it is a good idea to try to attempt the rest.

Problem 6 is quite technical and long (especially the question marked with a $(\star)$): it is alright to skip it, or to skip that subquestion if you attempt the problem. Attempt Problem 7 if you have time: it is not necessary, but gives perspective on the use of LSH.

Warm-up

Problem 1. Give a data structure for the Nearest Neighbour problem over a d-dimensional universe using space $O(nd)$, for which QUERY runs in time $O(nd)$). (Also, show that it can maintain S dynamically, and implement INSERT and REMOVE methods running in time $O(nd)$.)

Solution 1. This is obtained by maintaining a simple linked list containing all elements of S, which takes space $O(nd)$ when storing n elements of size $O(d)$ each. Assuming (as stated in the lecture) that computing the distance or checking equality between two elements x, y takes time $O(d)$, then a lookup takes time $O(n) \cdot O(d) = O(nd)$, and so insertions and deletions as well. A nearest neighbour query on an element x also takes time $O(nd)$, by linear search: going through all $y \in S$ one by one, computing $\text{dist}(x, y)$ for each, while keeping track of the y with the minimum distance so far – and returning that element at the end.

Problem 2. Give a data structure for the Nearest Neighbour problem over $\{0,1\}^d$ using space $O(2^d)$, for which QUERY runs in time $O(2^d)$ (independent of n). (Also, can maintain S dynamically, and implement INSERT and REMOVE methods running in time $O(1)$.)

Solution 2. Use a 2^d-sized bit array A, one for each d-bit string: $\{0,1\}^d$. So every $x \in \{0,1\}^d$ can be mapped one-to-one in each location of the array A. Initially every position in A is filled with 0. To insert x, simply mark $A[x] = 1$. To remove x, mark $A[x] = 0$. This array A takes $O(2^d)$ space.

To search for the nearest neighbour, one could iterate through the array, which takes $2^d \cdot O(d) = O(d2^d)$ time when done naively (since computing distances takes time $O(d)$ for each). A better option is to run an improved BFS: think of $\{0,1\}^d$ as a graph (hypercube), and each node has d neighbours at 1 hop, $\binom{d}{2}$ at 2 hops[1], $\binom{d}{3}$ at 3 hops etc. and so, at most 2^d times to search over all of them; doing so means it is not necessary to compute the distances as we go, since the level of the BFS corresponds to the current distance we are checking.

[1] There are two unique paths with length 2, from 00 to 11 for $\{0,1\}^2$.

Problem 3. Check your understanding: since we want very efficient lookups and are willing to accept a small probability of failure for QUERY, can we use Bloom filters for the "baby version" of LSH instead of hash tables? What fails?

Solution 3. We need to actually return some element that is $C \cdot r$-near in one case and Bloom filters do not store any element in the data structure.

Problem solving

Problem 4. ($\star$) Prove a simplified version of Theorem 38 from the lecture notes, showing how to solve the "general" ANN from the "baby version," at the cost of only a logarithmic factor in the ratio

$$\Delta = \frac{\max_{x,x' \in S} \operatorname{dist}(x,x')}{\min_{x,x' \in S} \operatorname{dist}(x,x')}$$

Note that, for the Hamming space $\{0,1\}^d$, $\Delta = O(d)$, where d is the dimension.

Solution 4.

Denote the pairwise closest distance over S:

$$d_{\min} = \min_{x,x' \in S} (\operatorname{dist}(x,x'))$$

and pairwise furthest distance:

$$d_{\max} = \max_{x,x' \in S} (\operatorname{dist}(x,x')),$$

so that $\Delta = \frac{d_{\max}}{d_{\min}}$.

Algorithm:

Build for a list of thresholds in the form:

$$R := \left\{ r \leqslant d_{\max} \mid r = 2^k \cdot \frac{d_{\min}}{2 \cdot C}, k \in \{0, 1, \ldots, O(\log \Delta)\} \right\}.$$

Denote $r_1, \ldots, r_{|R|}$ the list of thresholds from smallest to largest.

For each threshold $r \in R$, build your "baby" data structure $\triangleright\ O\left(\log \frac{d_{\max}}{d_{\min}}\right)$ of them in total

Binary/doubling search over R

Check the "baby" data structure with the middle threshold.

if it returns something **then**

  continue on the smaller parts.

else

  continue on the big parts.

8: **return** the best candidate the algorithm found.

Question: why stop at $\frac{d_{\min}}{2C}$?

Proposition. *Given query point x, there is at most one point $y \in S$ such that*

$$\operatorname{dist}(x,y) < \frac{d_{\min}}{2},$$

and y will the optimal point for x.

Proof. We prove by contradiction: suppose there are two distinct points in $y, y' \in S$ such that

$$\operatorname{dist}(x,y) < \frac{d_{\min}}{2} \text{ and } \operatorname{dist}(x,y') < \frac{d_{\min}}{2}.$$

By the triangle inequality (from dist being a metric) and definition of $d_{\min}$:

$$d_{\min} \leqslant \operatorname{dist}(y',y) \leqslant \operatorname{dist}(x,y) + \operatorname{dist}(x,y') < \frac{d_{\min}}{2} + \frac{d_{\min}}{2},$$

a contradiction. Therefore, there is at most one $y \in S$ such that $\operatorname{dist}(x,y) < \frac{d_{\min}}{2}$, which then must be the optimal point. □

Notice the threshold $\frac{d_{\min}}{2C}$ and the baby version's guarantee: if the optimal x^*'s distance $\operatorname{dist}(x^*,x) \leqslant \frac{d_{\min}}{2C}$, then the baby version will return some point (with good probability) that is at distance at most $C \cdot \frac{d_{\min}}{2C} = \frac{d_{\min}}{2}$ from x (which is guaranteed to be the optimal by the proposition).

Question: why stop at $d_{\max}$?

Proposition. *If* $\mathrm{OPT} = \operatorname{dist}(x^*,x) \geqslant d_{\max}$*, then returning any point $y \in S$ we have*

$$\operatorname{dist}(x,y) \leqslant 2 \cdot \mathrm{OPT}.$$

Proof. Suppose x^* is a closest one and that $\operatorname{dist}(x^*,x) \geqslant d_{\max}$. Let y be any point in S. Since $x^*, y \in S$, by definition, $\operatorname{dist}(x^*,y) \leqslant d_{\max}$. But then,

$$\begin{aligned}
\operatorname{dist}(x,y) &\leqslant \operatorname{dist}(x,x^*) + \operatorname{dist}(x^*,y) \\
&\leqslant \operatorname{dist}(x,x^*) + d_{\max} \\
&\leqslant 2\operatorname{dist}(x,x^*) \\
&= 2 \cdot \mathrm{OPT}.
\end{aligned}$$

This holds for any $y \in S$. □

If $\mathrm{OPT} = \operatorname{dist}(x,x^*)$ lies in between $\frac{d_{\min}}{2C}$ and $d_{\max}$, by the way we build our table, there exists $i \in \{1, 2, \ldots, |R|\}$ such that

$$r_i \leqslant \mathrm{OPT} \leqslant r_{i+1} \text{ and } r_{i+1} = 2 \cdot r_i.$$

When run with r_{i+1}, by the "baby version"'s guarantee, we will return some $y \in S$ that

$$\operatorname{dist}(x,y) \leqslant C \cdot r_{i+1} = 2C \cdot r_i \leqslant 2C \cdot \mathrm{OPT}.$$

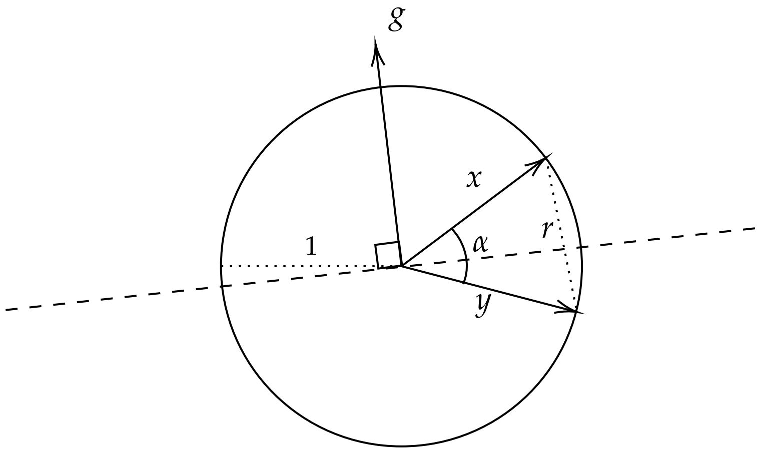
We can conclude now that no matter what OPT is, we will return a $(2 \cdot C)$-nearest neighbour.

Problem 5. Analyse the LSH family described in the lecture notes for the Euclidean case, where a locally-sensitive hash function $h_g\colon \mathbb{R}^d \to \{-1,1\}$ is obtained by drawing a d-dimensional Gaussian random vector $g \sim \mathcal{N}(0_d, I_d)$ (all coordinates are independent $\mathcal{N}(0,1)$ normal random variables) and setting

$$h_g\colon x \in \mathbb{R}^d \to \operatorname{sign}\Big(\sum_{i=1}^{d} g_i x_i\Big)$$

We will make the (restrictive) assumption that all data points and query points have unit norm: $\|x\|_2 = 1$. Show that, for every $r > 0, C > 1$, this defines an (r, C, p, q)-LSH family with p, q such that $\rho \leq 1/C$. *[Note: this is called the SimHash scheme.]*

Solution 5. Fix any $r > 0$ (wlog, $0 < r \leq 2$, since two unit vectors are at distance at most 2), and $C > 0$. Suppose that $x, y \in \mathbb{R}^d$ are two unit-norm vectors at distance r: $\|x\|_2 = \|y\|_2 = 1$, $\|x - y\|_2 = r$. Then $\Pr_g[h_g(x) \neq h_g(y)]$ is exactly the proba-



bility, over the choice of g, that $\langle g, x\rangle$ and $\langle g, y\rangle$ have different signs, which is the probability that x and y fall on different sides of the hyperplane defined by g (that is, whose normal vector is g). Looking at the plane defined by x, y, and letting α be the angle between x and y (see Figure), this is the probability the (projection of that) hyperplane falls between x and y, which is $\frac{\alpha}{\pi}$. So

$$\Pr_g[h_g(x) \neq h_g(y)] = \frac{\alpha}{\pi}\,.$$

Using some trigonometry (and the fact that $\|x\|_2 = \|y\|_2 = 1$) we get $r^2 = \sin^2\alpha + (1-\cos\alpha)^2$, that is, $r^2 = 2 - 2\cos\alpha$, which gives us $\alpha = \arccos(1 - r^2/2)$, and so

$$\Pr_g[h_g(x) \neq h_g(y)] = \frac{1}{\pi}\arccos\Big(1 - \frac{r^2}{2}\Big) = \frac{2}{\pi}\arcsin\frac{r}{2}\,.$$

(The last value is simpler, to state, and follows from the trigonometric identity $\arcsin x = \frac{1}{2}\arccos(1 - 2x^2)$, for $x \in [0,1]$. You don't need to prove it.) This implies that $\mathcal{H}$ is an (r, C, p, q)-LSH family for

$$p = 1 - \frac{1}{\pi}\arccos\Big(1 - \frac{r^2}{2}\Big), \qquad q = 1 - \frac{1}{\pi}\arccos\Big(1 - \frac{C^2 r^2}{2}\Big),$$

and has sensitivity

$$\rho = \frac{\log\left(1 - \frac{1}{\pi}\arccos\left(1-\frac{r^2}{2}\right)\right)}{\log\left(1 - \frac{1}{\pi}\arccos\left(1-\frac{C^2r^2}{2}\right)\right)} = \boxed{O\left(\frac{1}{C}\right)},$$

where this last inequality can be "guessed" by writing (for $r \to 0$)

$$\log\left(1 - \frac{1}{\pi}\arccos\left(1-\frac{r^2}{2}\right)\right) = \log\left(1-\Theta(r)\right) = \Theta(r)$$

(and same for the denominator); and can be proven formally as follows *(extra/not necessary!)*:

$$\begin{aligned}
\frac{\log\left(1-\frac{2}{\pi}\arcsin\frac{r}{2}\right)}{\log\left(1-\frac{2}{\pi}\arcsin\frac{Cr}{2}\right)} &\leq \frac{\frac{2}{\pi}\arcsin\frac{r}{2}}{-\log\left(1-\frac{2}{\pi}\arcsin\frac{Cr}{2}\right)} \\
&\leq \frac{\frac{r}{2}}{-\log\left(1-\frac{2}{\pi}\arcsin\frac{Cr}{2}\right)} \\
&= \frac{1}{C}\cdot\frac{1}{f\left(\frac{Cr}{2}\right)}
\end{aligned}$$

where $f(x) := \frac{-\log\left(1-\frac{2}{\pi}\arcsin x\right)}{x}$. "All" that remains is to show that $f(x) \geq 1$ for all $x \in (0, 1/2)$ (e.g., by showing that f is increasing, with $\lim_{x\to 0} f(x) = 1$). This shows that $\boxed{\rho \leq 1/C}$ (not even a need for the $O(\cdot)$).

Problem 6. ($\star$) For the set $[d] = \{1, 2, \dots, d\}$, let the universe $\mathcal{X}$ be the set of all 2^d subsets of $[d]$, along with the *Jaccard distance*:

$$\mathrm{dist}(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}, \qquad A, B \in \mathcal{X}$$

Consider the following hash family $\mathcal{H}$: for every permutation $\pi\colon [d] \to [d]$, define $h_\pi\colon \mathcal{X} \to [d]$ by setting

$$h_\pi(A) = \min_{a\in A} \pi(a)$$

and $\mathcal{H} = \{h_\pi\}_\pi$.

a) ($\star$) Verify that the Jaccard distance is a metric on $\mathcal{X}$. What is its range?

b) What is the size of $\mathcal{H}$?

c) Show that, for every $r \in (0, 1]$ and $C > 1$, $\mathcal{H}$ is an (r, C, p, q)-LSH family for $p = 1 - r$ and $q = 1 - Cr$. What is its sensitivity parameter ρ?

Solution 6. *Preliminary technical results about sets.* For one of the three properties of a metric, we will need the following intermediate (technical and annoying to show) results, which hold for any 3 sets A, B, C:

$$|A| + |B| = |A \cup B| + |A \cap B| \tag{†}$$

(follows from "proof by drawing", or writing $A \cup B = (A \setminus B) \cup B$. In detail: $A \setminus B$ and B are disjoint, so $|A \cup B = |A \setminus B| + |B|$. Now $A \setminus B = A \setminus (A \cap B)$ and $A \cap B \subseteq A$, so $|A \setminus B| = |A| - |A \cap B|$.)

$$|A \cap C| \cdot |B \cup C| + |A \cup C| \cdot |B \cap C| \leq |C|(|A| + |B|) \tag{$*$}$$

To prove this one: by (†),

$$|B \cup C| = |B| + |C| - |B \cap C|$$

and since $|A \cap C| \leq |C|$,

$$\begin{aligned} |A \cap C| \cdot |B \cup C| &= |A \cap C| \cdot |C| + |A \cap C| \cdot (|B| - |B \cap C|) \\ &\leq |A \cap C| \cdot |C| + |C| \cdot (|B| - |B \cap C|) \\ &= |C|(|B| + |A \cap C| - |B \cap C|) \end{aligned}$$

Similarly for the other term, and so

$$|A \cap C| \cdot |B \cup C| + |B \cap C| \cdot |A \cup C| \leq |C|(|A| + |B| + \cancel{|A \cap C|} - \cancel{|B \cap C|} + \cancel{|B \cap C|} - \cancel{|A \cap C|})$$

proving ($*$). Finally, we will need

$$|C| \cdot |A \cup B| \leq |A \cup C| \cdot |B \cup C| \tag{‡}$$

which follows from the sequence of inequalities, setting $S := A \cup C$, $T := B \cup C$ and

$$\begin{aligned} |C| \cdot |A \cup B| &\leq |(A \cup C) \cap (B \cup C)| \cdot |A \cup B \cup C| \\ &= |S \cap T| \cdot |S \cup T| \\ &\leq |S| \cdot |T| && \text{(from ($*$), "$A = B = S$" and "$C = T$")} \\ &= |A \cup C| \cdot |B \cup C| \end{aligned}$$

which shows (‡).

a) It is straightforward to check that $\text{dist}(A, B) \in [0, 1]$ for every $A, B \subseteq [d]$, since $|A \cap B| \leq |A \cup B|$. (Small technicality: we assume/choose here that if $A = B = \emptyset$, then we set $\text{dist}(\emptyset, \emptyset) = 0$ to avoid a ratio $0/0$.)
We can check the 3 axioms of a metric:

Reflexivity: if $A = B$, then $A \cap B = A = A \cup B$, and $\text{dist}(A, B) = 1 - \frac{|A|}{|A|} = 1 - 1 = 0$. Conversely, if $\text{dist}(A, B) = 0$, then $|A \cap B| = |A \cup B|$, and since $A \cap B \subseteq A \cup B$ this implies $A \cap B = A \cup B$, and so $A = B$.

Symmetry: $\text{dist}(A, B) = \text{dist}(B, A)$, since $\cap$ and $\cup$ are both symmetric.

Triangle inequality: Fix any $A, B, C \subseteq [d]$. Then what we want to show

$$\text{dist}(A, B) \leq \text{dist}(A, C) + \text{dist}(C, B)$$

is equivalent to $\frac{|A \cap B|}{|A \cup B|} \geq \frac{|A \cap C|}{|A \cup C|} + \frac{|B \cap C|}{|B \cup C|} - 1$ that is,

$$\frac{|A \cap B| + |A \cup B|}{|A \cup B|} \geq \frac{|A \cap C|}{|A \cup C|} + \frac{|B \cap C|}{|B \cup C|}$$

which is the inequality that we will establish. Note that if any of A, B, C is empty, we are done. If not (all are non-empty), then

$$\begin{aligned}
\frac{|A \cap C|}{|A \cup C|} + \frac{|B \cap C|}{|B \cup C|} &= \frac{|A \cap C| \cdot |B \cup C| + |A \cup C| \cdot |B \cap C|}{|A \cup C| \cdot |B \cup C|} \\
&\leq \frac{|C| \cdot (|A| + |B|)}{|A \cup C| \cdot |B \cup C|} && (*) \\
&= \frac{|C| \cdot |A \cup B|}{|A \cup C| \cdot |B \cup C|} \cdot \frac{|A| + |B|}{|A \cup B|} \\
&= \frac{|C| \cdot |A \cup B|}{|A \cup C| \cdot |B \cup C|} \cdot \frac{|A \cup B| + |A \cap B|}{|A \cup B|} && (\dagger) \\
&\leq \frac{|A \cup B| + |A \cap B|}{|A \cup B|} && (\ddagger)
\end{aligned}$$

and we're (at last) done.

b) The LSH family contains as many functions as there are permutations of $[d]$, which is $d!$. So $|\mathcal{H}| = d!$, or, put differently, $\log_2 |\mathcal{H}| = O(d \log d)$.

c) For any two $A, B \in \mathcal{X}$, the probability (over the uniformly random choice of $h \in \mathcal{H}$ that $h(A) = h(B)$ is the probability that

$$\min_{a \in A} \pi(a) = \min_{a \in B} \pi(a)$$

over the uniformly random choice of π. To reformulate this: if the minimum value that π takes on $A \cup B$ is in $A \cap B$, then $\min_{a \in A} \pi(a) = \min_{a \in A \cup B} \pi(a) = \min_{a \in B} \pi(a)$, and $h_\pi(A) = h_\pi(B)$. But if the minimum value that π takes on $A \cup B$ is in $(A \setminus B) \cup (B \setminus A)$, then either $\min_{a \in A} \pi(a) < \min_{a \in B} \pi(a)$ (if it's in $A \setminus B$) or $\min_{a \in A} \pi(a) > \min_{a \in B} \pi(a)$ (if it's in $B \setminus A$), and in both cases $h_\pi(A) \neq h_\pi(B)$. So

$$\Pr_\pi[h_\pi(A) = h_\pi(B)] = \Pr_\pi[\arg\min_{a \in A \cup B} \pi(a) \in A \cap B] = \frac{|A \cap B|}{|A \cup B|} = 1 - \operatorname{dist}(A, B)$$

which directly implies, for every r and C, that $\mathcal{H}$ is an (r, C, p, q)-LSH family for $p = 1 - r$ and $q = 1 - Cr$ (for $C < 1/r$). The sensitivity parameter is then

$$\rho = \frac{\log \frac{1}{1-r}}{\log \frac{1}{1-Cr}} = \frac{\log(1-r)}{\log(1-Cr)} = \Theta\left(\frac{1}{C}\right) .$$

Extra: To give a rigorous proof of this last part, we can do as follows:

$$\rho = \frac{\log(1-r)}{\log(1-Cr)} \leq \frac{r}{-\log(1-Cr)} = \frac{1}{C} \cdot \frac{Cr}{-\log(1-Cr)} .$$

Now, study the function $f(x) = \frac{-\log(1-x)}{x}$ over $(0, 1)$, and show that it is positive and increasing, with $\lim_{x \to 0} f(x) = 1$. This implies $\rho = \frac{1}{C} \cdot \frac{1}{f(Cr)} \leq \frac{1}{C}$.

Advanced

Problem 7. Give a data structure for the Nearest Neighbour problem over the Euclidean space $(\mathbb{R}^d, \ell_2)$ based on kd-trees. Analyse the space complexity of the data structure and its query time.