Class 8

Locality Sensitive Hashing
$c$-nearest neighbors

$S = \{x_1, x_2, \ldots, x_n\} \subseteq S^d$

(For today all of our points live on the unit sphere.)

Given $y$, find $x_j$ so that

$\|y - x_j\|_2 \leq c \left( \min_i \|y - x_i\|_2 \right)$

**Goals:**
- Space = $(d \cdot n)^{O(1)}$
- Query time = $o(n)$
c-near-neighbors

Imagine that this slide is the surface of the unit sphere....

\[ r = \min_i \left( \|x_i - y\|_2 \right) \]

Okay to return this.
Today: \((r, c)\)-near-neighbors

\[
S = \{x_1, x_2, \ldots, x_n\} \subseteq S^d
\]

(For today all of our points live on the unit sphere.)

Before:

Given \(y\), find \(x_j\) so that

\[
\|y - x_j\|_2 \leq c \min_i \|y - x_i\|_2
\]

Given \(y\) so that \(\min_i \|y - x_i\|_2 \leq r\)

find \(x_j\) so that \(\|y - x_j\|_2 \leq c \cdot r\)

**Goals:**
- Space = \((d \cdot n) \Theta(1)\)
- Query time = \(o(n)\)
$(r, c)$-near-neighbors

Imagine that this slide is the surface of the unit sphere....

Okay to return this.
Fact

• If you can solve \((r, c)\)-nearest neighbors then you can (basically) solve \(c\)-nearest neighbors.

• (See lecture notes).
Goal for today

• A solution to \((r, c)\)-approximate nearest neighbors.

• Tool: **Locality-Sensitive Hashing.**
  • Points that are near to each other have a good probability of colliding.
  • Points that are far from each other are unlikely to collide.

• Strategy:
  • Data structure: hash all the \(x_i\)
  • To query, hash \(y\). Return anything in \(y\)’s bucket.

Our strategy will actually be slightly more complicated than this, but this is the basic idea...
Our Locality Sensitive Hashing Scheme

• Let $A \in \mathbb{R}^{k \times d}$ have i.i.d. $N(0,1)$ entries.
  • Here, $k = \frac{\pi \log n}{2r}$ (we’ll see why later).
• Define $h(x) = \text{sign}(Ax)$
Actually choose $s$ independent copies of this

- Choose $s = \sqrt{n}$
- Choose $k = \frac{\pi \log n}{2r}$
- For $i = 1, \ldots, s$:
  - Let $A_i \in \mathbb{R}^{k \times d}$ have i.i.d. $N(0,1)$ entries.
  - Define $h_i(x) = \text{sign}(A_ix)$
Outline of group work

• First (problems 1-5) you will show that:
  • If $x, y$ are close, then probably there’s some $i$ so that $h_i(x) = h_i(y)$
  • If $x, y$ are far, then probably there’s no such $i$.

• Then (problems 6,7), you will show how to use this to get a $(c, r)$-near-neighbors scheme.
Group work!
Question 1

• For each row $a^T$ of $A$, we have a hyperplane $\{x \in \mathbb{R}^d : a^T x = 0\}$.

• If the corresponding coordinate of $Ax$ is negative, then $x$ lies on one side of the hyperplane, else $x$ lies on the other.

• Same cell = same side of every hyperplane = same sign in every coordinate.

\[
\begin{array}{cccc}
A & = & x & \rightarrow \\
\begin{array}{c}
0.2 \\
-1.3 \\
0.7 \\
3.2 \\
-50 \\
0.01
\end{array}
& \rightarrow & \begin{array}{c}
+1 \\
-1 \\
+1 \\
+1 \\
-1 \\
+1
\end{array}
\end{array}
\]

$A x = h(x)$
Question 2

\[ \Pr[h_i(x) = h_i(x)] = (1 - \frac{\text{angle}(x, y)}{\pi})^k \]

\[ \Pr\{\text{some hyperplane does that}\} = \frac{\text{arc length of } \odot}{\pi} = \frac{\text{angle}(x, y)}{\pi} \]

\[ \Pr\{\text{none of the } k \text{ hyperplanes do that}\} = \left(1 - \frac{\text{angle}(x, y)}{\pi}\right)^k \]
Question 3

\[ \Pr \left\{ \forall i, h_i(x) \neq h_i(y) \right\} = \left(1 - (1 - \frac{\text{angle}(x,y)}{\pi})^k\right)^s \]

\[ \Pr[h_i(x) = h_i(y)] \text{ for any one } i \]

\[ \Pr[h_i(x) \neq h_i(y)] \text{ for any one } i \]

\[ \Pr[h_i(x) \neq h_i(y)] \text{ for all } s \text{ values of } i \]
Question 4

With our choice of $s = \sqrt{n}$, $k = \frac{11 \log n}{2r}$, 

$$\mathbb{P} \left\{ \forall i, h_i(x) = h_i(y) \right\} = \left( 1 - \left( 1 - \frac{\text{ang}(x,y)}{\pi} \right)^{\frac{k}{s}} \right)^{\sqrt{n}}$$

$$\approx \left( 1 - \exp \left( - \log(n) \cdot \frac{\text{ang}(x,y)}{2r} \right) \right)^{\sqrt{n}}$$

$$\approx \exp \left( - \sqrt{n} \cdot \frac{\text{ang}(x,y)}{2r} \right)$$

$$= \exp \left( - \sqrt{n} \cdot \frac{1}{2} \cdot \frac{\text{ang}(x,y)}{2r} \right)$$
Question 5(a)

If $\text{angle}(x,y) \leq r$,

$$P\{ \forall i, h_i(x) = h_i(y) \} \leq \exp \left( -n \frac{1}{2} \cdot \text{angle}(x,y) r \right) \geq \exp (-1)$$

$$P\{ \exists i, h_i(x) = h_i(y) \} \geq 1 - \nu$$.
Question 5(b)

If \( \theta(x, y) \geq 5 \pi \), 

\[
\exp\left(-n^{\frac{1}{2}} \cdot \theta(x, y) \right) \leq \exp\left(-n^{\frac{1}{2}} - 5^{\frac{1}{2}}\right)
\]

\[
= \exp\left(-n^{-2}\right)
\]

\[
\approx 1 - \frac{1}{n^2}
\]

\[
\Pr\{\exists i, \ h_i(x) = h_i(y) \} \leq \frac{1}{n^2}
\]
Question 6

• Query Algorithm:
  • For $i = 1,2, \ldots, s$:
    • Compute $h_i(y)$
    • If there’s some $x_j$ so that $h_i(x_j) = h_i(y)$, return it.

• If $\angle(y, x_\ell) \leq r$, then with decent probability there’s some $i$ so that $h_i(x_\ell) = h_i(y)$.
  • In particular, the algorithm will return something.

• If the algorithm returns $x_j$ then with high probability $\angle(y, x_j) \leq 5r$.
  • If $\angle(y, x_j) > 5r$, $\Pr[\exists i, h_i(x_j) = h_i(y)] \leq \frac{1}{n^2}$, and we can union bound over all $x_j$ to say that never happens whp.
Question 7

Using $\frac{2}{\pi} \angle(x, y) \leq \|x - y\|_2 \leq \angle(x, y)$, we can conclude:

- If $\angle(x, y) \leq r$, $\mathbb{P}\{\exists i, h_i(x) = h_i(y)\} \geq 1 - \frac{1}{n^2}$.

- If $\angle(x, y) \geq 5r$, $\mathbb{P}\{\exists i, h_i(x) = h_i(y)\} \leq \frac{1}{n^2}$

So just fiddle with the value of “c” and the same analysis will still work.
Wrapping up: Time and Space

- **Space:** \( n^{O(1)} \)
  - \( s \) different \( k \times d \) matrices \( A_i \): \( O(d \cdot \sqrt{n} \cdot \log n) \)
  - \( s \) hash tables, each with \( 2^k \) buckets: \( O(\sqrt{n} \cdot 2^{O(\log n)}) = n^{O(1)} \)
  - The elements of \( S \) themselves: \( O(nd) \)

- **Update time:** \( O(d \cdot \sqrt{n} \cdot \log n) = o(n) \) when \( d \) isn’t too big.
  - \( s \) different \( k \times d \) matrix-vector multiplies: \( O(s \cdot k \cdot d) = O(d \cdot \sqrt{n} \cdot \log n) \)
  - Going through all \( s \) hash tables and look in \( h_i(y) \)'s bucket to see if there’s anything else: \( O(s) = O(\sqrt{n}) \)

\[ k = O(\log n) \]
\[ s = \sqrt{n} \]