CS265, Fall 2020

Class 8: Agenda, Questions, and Links

1 Announcements

- HW3 due Friday!

2 Questions?

Any questions from the minilectures? (JL Lemma; Intro to nearest neighbors)

- Go into small groups and ask each other your questions.
- **New!!** Please go to `pollev.com/cs265` and ask your questions/comments there, or else upvote others' questions.

3 Locality Sensitive Hashing

[Slides with setup. Summary below.]

Recall the setup for c-approximate-nearest neighbors. We have a set S of size n, and **for today $S \subset \mathbb{S}^d$ lives on the surface of the d-dimensional sphere.** That is, $S = \{x_1, \ldots, x_n\}$, so that $x_i \in \mathbb{R}^d$ and $\|x\|_2 = 1$.

Our goal is to come up with some data structure to store the x_i's, so that:

- We don't use too much space (eg, uses space poly(n), where the exponent in the polynomial doesn't depend on d).
- Given $y \in \mathbb{S}^d$, we can find $x_i \in S$ so that
$$\|x_i - y\|_2 \leq c \cdot \min_j \|x_j - y\|_2$$
in time sublinear in n.

3.1 Nearest-Neighbors vs. Near Neighbors

[slides with overview; summary below]

Consider the following problem, called (r, c)**-near-neighbors.** We have a set $S \subset \mathbb{S}^d$ of size n, and our goal is to come up with some data structure (that doesn't use too much space) to store the x_i's, so that the following holds.

> Given $y \in \mathbb{S}^d$ so that $\min_j \|x_j - y\|_2 \leq r$, we can find $x_i \in S$, in sublinear time, so that $\|x_i - y\|_2 \leq cr$.

It turns out that if we can solve (r, c)-near-neighbors (for a decent range of r's) then we can solve c-nearest-neighbors. Check out the lecture notes for more on this.

3.2 A solution to (r, c)-near-neighbors

[Slides for set-up; summary below]

Let s, k be parameters, chosen as follows:

$$s = \sqrt{n}, \qquad k = \frac{\pi \log n}{2r}$$

For $i = 1, \ldots, s$, let $A_i \in \mathbb{R}^{k \times d}$ have i.i.d. $\mathcal{N}(0,1)$ entries. For $y \in \mathbb{S}^d$, define

$$h_i(y) = \mathrm{sign}(A_i y),$$

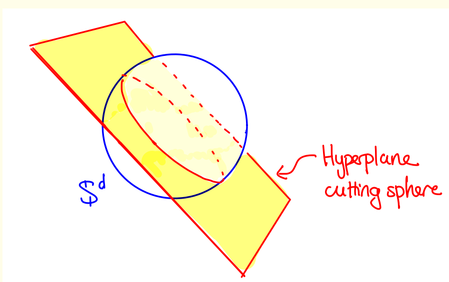
where for a vector $a \in \mathbb{R}^k$, $\mathrm{sign}(a) \in \{\pm 1\}^k$ is just the vector whose i'th entry is $+1$ if $a_i > 0$ and -1 if $a_i \leq 0$.

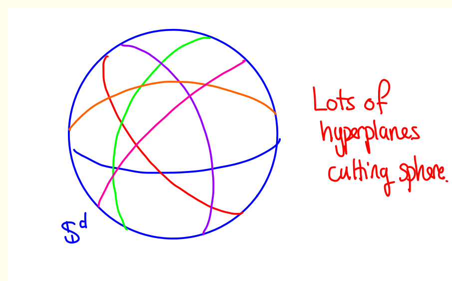
Group Work

Important: as you make progress on the question(s), one person in each room should record your progress on http://PollEv.com/cs265.

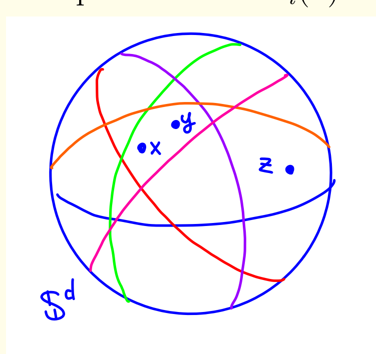
1. Consider a hash function $h_i : \mathbb{S}^d \to \{\pm 1\}^k$ as defined above. Explain why "$h_i(x) = h_i(y)$" has the following geometric meaning:

   Imagine choosing k uniformly random hyperplanes in $\mathbb{R}^d$, and using them to slice up the sphere $\mathbb{S}^d$ like this:





   Then $h_i(x) = h_i(y)$ if and only if x and y are in the same "cell" of this slicing. For example, in the picture below $h_i(x) = h_i(y) \neq h_i(z)$.



   Hint: Use the spherical symmetry of the Gaussian distribution.

2. Explain why, for $x, y \in \mathbb{S}^d$, and for any $i = 1, \ldots, s$,

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

where $angle(x, y) = \arccos(x^T y)$ is the arc-cosine of the dot product of x and y, aka, the angle between x and y.

Hint: Think about the geometric intuition in the plane spanned by x and y.

When you are done, record your progress on pollEverywhere.

3. Suppose that $x, y \in \mathbb{S}^d$. Fill in the blank:

$$\Pr[\forall i \in \{1, \ldots, s\}, h_i(x) \neq h_i(y)] = \text{-----}$$

When you are done, select your choice on the pollEverywhere.

4. Using our choices of s and k above, along with extremely liberal use of the approximation that $1 - x \approx e^{-x}$ for small x, convince yourself that

$$\Pr[\forall i \in \{1, \ldots, s\}, h_i(x) \neq h_i(y)] \approx \exp\left(-n^{1/2 - angle(x,y)/(2r)}\right).$$

5. Fill in the blanks:

   (a) If $angle(x, y) \leq r$, then

   $$\Pr[\exists i \in \{1, \ldots, s\} \text{ so that } h_i(x) = h_i(y)] \geq \text{------}$$

   (b) If $angle(x, y) \geq 5r$, then

   $$\Pr[\exists i \in \{1, \ldots, s\} \text{ so that } h_i(x) = h_i(y)] \leq \text{-------}.$$

   When you are done, record your progress on the pollEverywhere.

   A family of hash functions that does this is called *locality sensitive hashing,* because the probability that x and y hash to the same bucket depends on their "locality," eg, how close they are to each other.

6. Explain why, if you pretended that "$angle(x, y)$" was "$\|x - y\|_2$" everywhere, that this would give a (c, r)-near-neighbors scheme for some constant c. (It's okay if each query succeeds with probability 1/2 or something like that).

   Hint: As your data structure, imagine storing s hash tables, one for each h_i. In hash table i, you have 2^k buckets, one for each possible outcome of h_i, and you store a pointer to x in bucket $h_i(x)$. One query y, where should you look to find an x_j that's close to y?

7. Explain why it's okay to pretend that "$angle(x, y)$" is "$\|x - y\|_2$," perhaps at the cost changing the constants around.

   Hint: You can use the fact that

$$\frac{2}{\pi} angle(x, y) \leq \|x - y\|_2 \leq angle(x, y)$$

   for any $x, y \in \mathbb{S}^d$.