EPFL

Exercise Set XI

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. Solve as many problems as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

These problems are taken from various sources at EPFL and on the Internet, too numerous to cite individually.

1 Consider two LSH hash families $\mathcal{H}_1$ and $\mathcal{H}_2$ designed for a distance function $\text{dist} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. For $r = 0.1$ and $c = 2$, $\mathcal{H}_1$ satisfies

$$\text{dist}(p,q) \leqslant r \implies \Pr_{h \sim \mathcal{H}_1}[h(p) = h(q)] \geqslant 1/2$$
$$\text{dist}(p,q) \geqslant c \cdot r \implies \Pr_{h \sim \mathcal{H}_1}[h(p) = h(q)] \leqslant 1/8$$

and $\mathcal{H}_2$ satisfies

$$\text{dist}(p,q) \leqslant r \implies \Pr_{h \sim \mathcal{H}_2}[h(p) = h(q)] \geqslant 1/8$$
$$\text{dist}(p,q) \geqslant c \cdot r \implies \Pr_{h \sim \mathcal{H}_2}[h(p) = h(q)] \leqslant 1/200$$

1a Which Hash family would you choose to build the data structure ANNS(r, c) explained in class? What would the space requirement and query time be (logs are not so important)?

1b On query $q \in \mathbb{R}^d$, asymptotically how many hash function computations are done?

2 Suppose you have a database with a set $P \subseteq \mathbb{R}^d$ of n items that are equipped with a distance function $\text{dist} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ satisfying the following sparsity condition:

$$|\{p \in P : \text{dist}(p, q) \leq 2\}| \leq 10\,.$$

Further assume that you have a $(r, c \cdot r, p_1, p_2)$-LSH hash family $\mathcal{H}$ for the considered distance function with parameters $r = 1$, $c = 2$, $p_1 = 1/2$ and $p_2 = 1/8$. That is,

$$\begin{aligned} \text{dist}(p, q) \leqslant 1 &\implies \Pr[h(p) = h(q)] \geqslant 1/2 \\ \text{dist}(p, q) \geqslant 2 &\implies \Pr[h(p) = h(q)] \leqslant 1/8 \end{aligned}$$

where the probabilities are over $h \sim \mathcal{H}$.

Exploit the sparsity condition to modify the ANNS(c, r) construction seen in class so as to obtain a structure with the *same* asymptotic preprocessing and query times, but with the following improved guarantee:

> On query $q \in \mathbb{R}^d$, if $\min_{p \in P} \text{dist}(p, q) \leq 1$, then we return $\arg\min_{p \in P} \text{dist}(p, q)$ with probability close to 1.

(Notice that this is stronger than the guarantee seen in class as in that case one is only guaranteed to return a point p' such that $\text{dist}(p', q) \leq c \cdot r$ with probability close to 1.)

What is the preprocessing time, query time, and space requirement of your solution?

3 Consider a submodular function $f : 2^N \to \mathbb{R}$ over the ground set $N = \{1, 2, 3, 4\}$. What is the value of the Lovàsz extension $\hat{f}(0.75, 0.3, 0.2, 0.3)$ as a function of f?

4 Hypergraph cuts. Let $G = (V, E)$ be a hypergraph with vertex set V and hyperedge set E (every hyperedge $e \in E$ is a subset of V; see Fig. 1 for an illustration). For $S \subseteq V$ the set of hyperedges crossing the cut $(S, V \setminus S)$ is defined as

$$E(S, V \setminus S) = \{e \in E : e \cap S \neq \emptyset \text{ and } e \cap V \setminus S \neq \emptyset\},$$

and the size of the cut $(S, V \setminus S)$ as $|E(S, V \setminus S)|$.

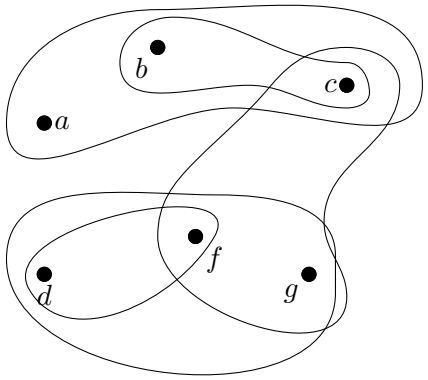

Figure 1. A hypergraph $G = (V, E)$ with $V = \{a, b, c, d, f, g\}$ and hyperedge set $E = \{e_1, e_2, e_3, e_4, e_5\}$, where $e_1 = \{a, b, c\}, e_2 = \{b, c\}, e_3 = \{d, f\}, e_4 = \{c, f, g\}$ and $e_5 = \{d, f, g\}$.

4a Give an algorithm that finds the size of the minimum cut in a given hypergraph G, i.e. outputs

$$\min_{S \subset V, S \neq \emptyset} |E(S, V \setminus S)|.$$

For example, the size of the minimum cut in the hypergraph G in Fig. 1 is 1. There are two minimum cuts: $(\{a\}, \{b, c, d, f, g\})$ and $(\{a, b, c\}, \{d, f, g\})$.

Your algorithm should run in time polynomial in the number of vertices and hyperedges in G.

4b Give a randomized polynomial time algorithm that outputs, given a hypergraph G where every hyperedge contains three vertices, a cut $(S, V \setminus S)$ such that

$$\mathbb{E}[|E(S, V \setminus S)|] \geq (3/4)OPT, \qquad (*)$$

where

$$OPT = \max_{S \subseteq V} |E(S, V \setminus S)|.$$

Note that unlike **4a**, here we are interested in the **maximum** cut. Your algorithm should run in time polynomial in the number of vertices and hyperedges in G, and you should prove that the expected size of the cut that it outputs satisfies $(*)$.