Exercise Set XI, Algorithms II 2024

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.

Additional problems on LSH and Nearest Neighbor

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

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

and $\mathcal{H}_2$ satisfies

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

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,

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

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?

Submodularity

3 Consider two submodular functions seen in class: the cut function $\delta(\cdot)$ of a graph and the coverage function $c(\cdot)$ of finite collection of sets. Are they monotone? Give proofs or counterexamples. *(We say that a set function f is* monotone *if $f(B) \geq f(A)$ for all $A \subseteq B \subseteq N$.)*

4 Let $f_1, \ldots, f_k : 2^N \to \mathbb{R}$ be submodular functions on the ground set N. Show that if $\lambda_1, \lambda_2, \ldots, \lambda_k \geq 0$ then the function $g : 2^N \to \mathbb{R}$ defined by $g(S) = \sum_{i=1}^{k} \lambda_i f_i(S)$ is a submodular function.

5 In the MaxSAT problem, we are given a set of m disjunctions over n variables and their negations. That is, each clause is of the form:

$$\ell_1 \vee \ell_2 \vee \ldots \vee \ell_d\,,$$

where each ℓ_i is either a variable or the negation of a variable. The goal is to assign each variable a value true or false, so that as many clauses as possible are true. Show how to formulate MaxSAT as a submodular optimization problem with a single matroid constraint.

6 *(Homework problem previous year)* You have just finished developing your first really cool App. Your target group are the students at EPFL and you plan to sell the App *expensively*. However, to launch your application (and to spread the word about how cool it is), you are willing to give away k free copies. To maximize the influence of these free copies you have the following graph model:

- The "friendship" graph has a vertex v for each student at EPFL and an edge $\{u, v\}$ if u and v are friends.
- For each student v, there is a probability $0 \leq p_v \leq 1$ that models how likely v is to tell their friends about your App (assuming v knows about it).

Given this information, the goal is to give free copies to k students so as to maximize the expected number of students that will have heard about your App. More formally, let A be a random set of students that contains each student v with probability p_v independently. Then the goal is to find a set $S = \{u_1, \ldots, u_k\}$ of k students so as to maximize

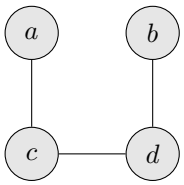
$$f(S) = \mathbb{E}_A\left[|\{v : v \text{ is informed if students in } A \text{ are spreading the word}\}|\right],$$

where v is informed if there is a path from v to a vertex in S in which all vertices, except possibly v, are in A. For an example see below.

Prove that the function f is submodular. Thus your problem reduces to that of maximizing a submodular function subject to a cardinality constraint. Since f is clearly monotone, we can use the greedy algorithm explained in class to launch our product in an awesome way.

Hint: show first that f is submodular if $p_v \in \{0, 1\}$ for every student v. Then notice that any probability assignment $p : V \to [0, 1]$ can be written as a weighted sum of such probability assignments.

Example. We have students a, b, c, d with $p_a = 1/2$, $p_b = 1/3$, $p_c = 0$, $p_d = 1$ and the friendship graph is



We are going to evaluate $f(\{a, b\})$. Notice that since c is never in the set A since $p_c = 0$ and d is always in A since $p_d = 1$ we have four potential outcomes of A. We consider each outcome separately:

Case 1 $A = \{d\}$**:** This happens with probability $(1 - p_a)(1 - p_b) = 1/3$ and students $\{a, b\}$ will be informed about your App.

Case 2 $A = \{d, a\}$**:** This happens with probability $p_a(1 - p_b) = 1/3$ and students $\{a, b, c\}$ will be informed about your App.

Case 3 $A = \{d, b\}$**:** This happens with probability $(1 - p_a)p_b = 1/6$ and students $\{a, b, c, d\}$ will be informed about your App.

Case 4 $A = \{d, a, b\}$**:** This happens with probability $p_a p_b = 1/6$ and students $\{a, b, c, d\}$ will be informed about your App.

Hence, in this example we have

$$f(\{a, b\}) = \underbrace{\frac{1}{3} \cdot 2}_{\text{Case 1}} + \underbrace{\frac{1}{3} \cdot 3}_{\text{Case 2}} + \underbrace{\frac{1}{6} \cdot 4}_{\text{Case 3}} + \underbrace{\frac{1}{6} \cdot 4}_{\text{Case 4}} = 3\,.$$