EPFL

Exercise Set X, 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.

Locality Sensitive Hashing

1 *(Final exam question from a previous year)* **LSH for Jaccard similarity.**

Recall the Jaccard index that we saw in Exercises: Suppose we have a universe U. For non-empty sets $A, B \subseteq U$, the Jaccard index is defined as

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$

Design a locality sensitive hash (LSH) family $\mathcal{H}$ of functions $h : 2^U \to [0,1]$ such that for any non-empty sets $A, B \subseteq U$,

$$\Pr_{h \sim \mathcal{H}}[h(A) \neq h(B)] \begin{cases} \leq 0.01 & \text{if } J(A,B) \geq 0.99, \\ \geq 0.1 & \text{if } J(A,B) \leq 0.9. \end{cases}$$

(In this problem you are asked to explain the hash family and argue that it satisfies the above properties. Recall that you are allowed to refer to material covered in the course.)

2 In this problem we design an LSH for points in $\mathbb{R}^d$ with the ℓ_1 distance, i.e.

$$d(p,q) = \sum_{i=1}^{d} |p_i - q_i|.$$

Define a class of hash functions as follows: Fix a positive number w. Each hash function is defined via a choice of d independently selected random real numbers $s_1, s_2, \ldots, s_d$, each uniform in $[0, w)$. The hash function associated with this random set of choices is

$$h(x_1, \ldots, x_d) = \left(\left\lfloor \frac{x_1 - s_1}{w} \right\rfloor, \left\lfloor \frac{x_2 - s_2}{w} \right\rfloor, \ldots, \left\lfloor \frac{x_d - s_d}{w} \right\rfloor \right).$$

Let $\alpha_i = |p_i - q_i|$. What is the probability that $h(p) = h(q)$, in terms of the α_i values? It may be easier to first think of the case when $w = 1$. Try also to simplify your expression if w is much larger than α_i's, using that $(1 - x) \approx e^{-x}$ for small values of $x \geq 0$.

3 *(*)* A certain ex-president's "university" has experienced a lot of cheating and you have been contacted to fix the problem. In particular, you should design a system for detecting plagiarism. To your help you have downloaded all the recent theses from the web. Let n be the number of theses. To simplify matters, you use the "bag of words" representation which just represents each thesis i by a vector v_i. The vector has an entry for each word in the English language that equals the number of times that word appears in the thesis. For example, in the ex-president's thesis, say thesis i, the word *rigorous* appears only once, whereas *tremendous* appears 1000 times, and so

$$v_i(\text{“rigorous”}) = 1 \qquad \text{and} \qquad v_i(\text{“tremendous”}) = 1000\,.$$

To deal with theses of different lengths, we normalize the vectors to have unit length. Let $u_1, \ldots, u_n$ denote the normalized vectors, i.e., $u_i = v_i/\|v_i\|_2$ for $i = 1, \ldots, n$. A good distance measure of similarity between two normalized vectors p and q (corresponding to two different theses) is the cosine similarity (or angular similarity) defined as follows:

$$\text{dist}(p, q) = \text{“the angle between } p \text{ and } q\text{”} = \cos^{-1}(\langle p, q\rangle)\,.$$

To detect plagiarism, it seems reasonable to inspect similar theses. That is, if the normalized vector q corresponding to a newly submitted thesis satisfies $\text{dist}(q, u_i) \leq 1^\circ$ for some $i = 1, \ldots, n$, then we would like to inspect u_i for plagiarism. To find such close u_i's, we wish to design an efficient (approximate) nearest neighbor search data structure. To do so, you need to design a family of locally sensitive hash (LSH) functions $\mathcal{H}$ that map normalized "bag of word" vectors to $\{0, 1\}$ such that for some $p_1 > p_2$:

- If $\text{dist}(q, u_i) \leq 1^\circ$ then $\Pr_{h\sim\mathcal{H}}[h(q) = h(u_i)] \geq p_1$. (we need to inspect u_i)
- If $\text{dist}(q, u_i) \geq 10^\circ$ then $\Pr_{h\sim\mathcal{H}}[h(q) = h(u_i)] < p_2$. (it is safe to ignore u_i)

What values of p_1 and p_2 do you get?

Hint: Randomly cut the sphere into two halves.

Submodular Functions (after Tuesday's lecture)

4 There is a set $[n] = \{1, \ldots, n\}$ of n different ice cream tastes. Maggie Simpson's total happiness goes up by $v_i \geq 0$ if she eats ice cream $i \in [n]$. However, as her stomach has bounded size, her happiness can never exceed some certain value $B > 0$. In other words, if Maggie eats $S \subseteq [n]$, her total happiness is

$$f(S) = \min\left(\sum_{i\in S} v_i, B\right)\,.$$

Show that f is a submodular function.

5 Let $f : 2^N \to \mathbb{R}$ be a submodular function. Show that the following functions are also submodular:

- $g(S) = f(S \cup A)$ where A is a fixed set.
- $g(S) = f(S \cap A)$ where A is a fixed set.
- $g(S) = f(N \setminus S)$.

6 *(half *)* Let $f : 2^N \to \mathbb{R}$ be a submodular function. Let $A \subseteq N$ and suppose that $A(p)$ is a random subset of A where each element $u \in A$ appears in $A(p)$ with probability p. Show that:

$$\mathbb{E}[f(A(p))] \geq (1-p)f(\emptyset) + pf(A).$$

To prove this, use that the Lovàsz extension $\hat{f}$ of f is the convex closure. That is, for input $z \in [0,1]^n$, we have that $\hat{f}(z)$ equals the minimum

$$\mathbb{E}_{S \sim \mu}[f(S)]$$

over all distributions μ of subsets satisfying the marginal probabilities: $\Pr_{S\sim\mu}[i \in S] = z_i$ for all $i \in \{1, 2, \ldots, n\}$.

7 Consider a directed $G = (V, E)$ and define the set function $f : 2^V \to \mathbb{R}$ by

$$f(S) = |\{(u,v) \in E : u \in S, v \notin S\}| \qquad \text{for every } S \subseteq V\,.$$

That is, $f(S)$ equals the number of arcs that exits the set S.

7a Show that f is a (non-monotone) submodular function

7b Let S be a random subset of vertices obtained by including each vertex with probability $1/2$ independently of other vertices. Show that

$$\mathbb{E}[f(S)] = |E|/4 \geq \mathrm{OPT}/4\,,$$

where $\mathrm{OPT} = \max_{T \subseteq V} f(T)$.

Also give an example of a graph where $|E| = \mathrm{OPT}$ and thus it shows that the analysis is tight with respect to OPT.

7c (*) Consider any submodular function f that is

- non-negative: $f(T) \geq 0$ for all T.

Let S be a random subset of elements obtained by including each element with probability $1/2$ independently of other elements. Then

$$\mathbb{E}[f(S)] \geq \mathrm{OPT}/4\,,$$

where $\mathrm{OPT} = \max_T f(T)$.

This shows that the simple randomized algorithm actually gives a good approximation to any (even non-monotone) submodular function assuming it is non-negative.