Massachusetts Institute of Technology
6.854J/18.415J: Advanced Algorithms
Ankur Moitra

Wednesday, February 24, 2016

Problem Set 3

Due: Wednesday, March 2, 2016 – 7 pm
Dropbox Outside Stata G5

Collaboration policy: collaboration is *strongly encouraged*. However, remember that

1. You must write up your own solutions, independently.
2. You must record the name of every collaborator.
3. You must actually participate in solving all the problems. This is difficult in very large groups, so you should keep your collaboration groups limited to 3 or 4 people in a given week.
4. Write each problem in a separate sheet and write down your name on top of every sheet.
5. **No bibles. This includes solutions posted to problems in previous years.**

1 Simple Approximate Nearest Neighbors

Here, we will consider a very simple, deterministically valid algorithm for approximate nearest neighbors whose major caveat is query time with exponential dependence on the dimension.

More specifically, we will focus on the $(R, (1+\epsilon)R)$ PLEB problem for the ℓ_∞ norm (but this scheme will work for other norms as well). We will define a data structure given a point set P, a radius R and an error parameter ϵ. Given any query point q, if there is any $p \in P$ such that $||p-q||_\infty \leq R$, the algorithm must output some $p' \in P$ such that $||p'-q||_\infty \leq (1+\epsilon)r$. If there is no such p, the algorithm may output *either* such a p' or NO.

The scheme we will use is simple: we will overlay an n-dimensional cubic grid over the points with each cell having some side length r, so that each $p = (p_1, p_2, ..., p_d) \in P$ is assigned to a grid cell $(\lfloor p_1/r \rfloor, \lfloor p_2/r \rfloor, ..., \lfloor p_d/r \rfloor)$. We will store a table (for instance, a hash table), indexed by the grid cells, storing the list of points contained in each cell. Note that the space used by this data structure is linear in n and d, since each point is assigned to only one cell and can be stored in d space.

Given a query point q, we will simply search all grid cells within distance R of q–that is, all grid cells that contain a point within R of q. If any of these cells is nonempty, we will take the first point we find and return it. Otherwise, output NO.

(a) By construction, if there is a $p \in P$ within R of q, this algorithm will not output NO, since p's grid cell is nonempty.

However, to satisfy the definition of the PLEB problem, we additionally must ensure that the output point is within $(1+\epsilon)R$ of p. This guarantee only holds if r is sufficiently small.

What is the largest value we can choose for r to make this a valid algorithm for the approximate PLEB?

(b) What is the query time of this algorithm, as a function of the dimension d and error parameter ϵ? You may assume that $\frac{1}{\epsilon}$ is an integer and that we can do the table lookup for the d-dimensional grid squares in $O(d)$ time.

(c) How can we modify this data structure and query algorithm, so that the query time becomes linear in d with no dependence on ϵ or n, while the *space* inherits the bad dependence on d and ϵ (while still being linear in n)? Again, you may assume that the table lookup takes $O(d)$ time.

2 Locality Sensitive Hashing for ℓ_2

In this problem, we will analyze a simple locality-sensitive hashing scheme for ℓ_2. For convenience we will assume the distance scale is 1: in the notation from class, we will be constructing a $(1, c, p_1, p_c)$ locality sensitive hash family.

To produce a hash function, we will choose d values $g_1 ... g_d$, each one an independently sampled standard Gaussian ($N(0,1)$), and a real number s chosen uniformly at random from $[0,1]$.

The hash of a point x will then be

$$H_{g,s}(x) = \left\lfloor s + \sum_{i=1}^{d} g_i x_i \right\rfloor .$$

In other words, we first project the point to the real line with a Gaussian random vector, then quantize to a randomly shifted one-dimensional grid on that line. We're asking for closer points to be more likely to end up in the same cell.

(a) Prove that for any choice of two points x and y, the probability (over g and s) that $H_{g,s}(x) = H_{g,s}(y)$ depends only on $\|x-y\|_2$.

Hint: consider the quantity $\sum_{i=1}^{d} g_i x_i - \sum_{i=1}^{d} g_i y_i$, and remember than the sum of an independent $N(0,a)$ and $N(0,b)$ is $N(0,a+b)$.

(b) Argue that there is some fixed constant $p_1 > 0$ such that for any x, y satisfying $\|x-y\|_2 \leq 1$,

$$\mathbb{P}[H_{g,s}(x) = H_{g,s}(y)] \geq p_1.$$

You don't need to give an explicit value for p_1.

(c) Prove that that for some universal constant B, for *any* x, y,

$$\mathbb{P}[H_{g,s}(x) = H_{g,s}(y)] \leq \frac{B}{\|x-y\|_2}.$$

Again, you don't need to give any explicit value for B.

In particular, this means that p_c, the largest possible collision probability when $\|x-y\|_2 \geq c$, will be at most $\frac{B}{c}$. That in turn means that as $c \to \infty$, $\frac{\log(1/p_1)}{\log(1/p_c)}$ will approach 0. Using this with the results from class, the additional polynomial overhead in locality sensitive hashing can be made arbitrarily small by allowing a weaker approximation factor.

(d) (**Optional**) Consider the rescaled version of this scheme: computing $H_{h,s}(x/S)$ for some $S > 1$ (but where we are still interested in points at distance scale about 1). This is equivalent to quantizing the real line to spacing S rather than 1.

As $S \to \infty$, the collision probabilities will all approach 1, but what will happen to $\frac{\log(1/p_1)}{\log(1/p_c)}$? Can you get an improved bound using larger S?

Note that the downside to choosing S too large is that the number of invocations of the base hash function for each table lookup will increase as the probabilities go to 1. Eventually this extra cost will outweigh the benefit.

3 Alternative ℓ_2 Approximation

Note: this problem is closely related to the Problem 3 from the last problem set.

In class, we showed Johnson-Lindenstrauss: that for any $0 < \epsilon, \delta \leq 1/2$, with a Gaussian random matrix Π of height $O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$, $\|\Pi x\|_2^2$ is within ϵ multiplicative error of $\|x\|_2^2$ with probability at least $1-\delta$.

However, such a matrix is dense and could be somewhat expensive to apply. "Sparse Johnson-Lindenstrauss" results exist, with only $O\left(\frac{\log(1/\delta)}{\epsilon}\right)$ nonzero entries per column (i.e. only about an ϵ fraction of entries). In this problem, we will show a similar way to approximate ℓ_2 norms with an even sparser linear transform, so that the work has no dependence on ϵ. The caveat is that the estimate will not just be $\|Ax\|_2^2$, but will be a somewhat more complicated procedure.

Specifically, for embedding from R^n and with two parameters q and k, we will define k linear maps from R^n to R^q; applying each of them to x will leave us with k q-dimensional vectors $v_1 ... v_k$ (we will refer to the rth coordinate of v_j as $v_j[r]$). Each of these maps will be picked independently at random from the same distribution. These maps can each be applied in linear time in n, so that the whole procedure takes $O(kn)$ time.

For each $j \in \{1...k\}$ (i.e. for each independent copy) and for each $i \in \{1...n\}$ we choose an independent random value $r_{j,i}$, picked uniformly at random from $\{1...q\}$. We additionally choose an independent random sign (± 1) $s_{j,i}$. We could view these as k random hash functions from $\{1...n\}$ to $\{1...q\}$ and $\{-1, 1\}$, which can be useful in the streaming setting.

To compute v_j, we initialize v_j to 0. We then iterate through each i from 1 to n and perform

$$v_j[r_{j,i}] \Leftarrow v_j[r_{j,i}] + s_{j,i}x_i$$

In other words, x_i was added to a random bucket with a random sign. This is essentially the same as the sketching procedure from Problem 3 on the last problem set. To get an intuition for why increasing q makes this more accurate, consider the case when q is so large that there are no "collisions": each x_i is mapped to a unique entry of v_j. In that case $\|v_j\|_2^2$ will be exactly equal to $\|x\|_2^2$.

(a) Show that $\mathbb{E}[\|v_1\|_2^2] = \|x\|_2^2$ (by symmetry, this holds for all other v_j as well).

(b) Show that the variance $\mathbb{E}[(\|v_1\|_2^2 - \|x\|_2^2)^2] \leq \frac{2}{q}\|x\|_2^4$ (again, this holds for all v_j).

Hint: Try writing an expression for $\|v_1\|_2^2 - \|x\|_2^2$ (*deterministically*, as a function of the specific random choices made) as a sum over monomials in x_i, $s_{1,i}$, and some indicator variables based on the $r_{1,i}$. You may want to introduce indicator variables $c_{1,i,i'}$ representing a "collision" between i and i' ($r_{1,i} = r_{1,i'}$). Once you have this, you can get such an expression for the square by the general transformation $\left(\sum_a t_a\right)^2 = \sum_{a,a'} t_a t'_a$. At this point you can apply linearity of expectation. Think carefully about which terms must have expectation 0!

(c) Conclude that if $q = \frac{5}{\epsilon^2}$,

$$(1-\epsilon)\|x\|_2^2 \leq \|v_1\|_2^2 \leq (1+\epsilon)\|x\|_2^2$$

with probability at least $\frac{3}{5}$.

(d) Show that for this setting of q, we may set k to $O(\log(1/\delta))$, and if we then define M to be the *median* of $\|v_1\|_2^2 ... \|v_k\|_2^2$,

$$(1-\epsilon)\|x\|_2^2 \leq M \leq (1+\epsilon)\|x\|_2^2$$

with probability at least $1-\delta$.

4 The Fast Johnson-Lindenstrauss Transform

In this problem, we'll go through an analysis of a "fast Johnson-Lindenstrauss transform." This will be a random m by n matrix M satisfying a randomized Johnson-Lindenstrauss property: that is, for any vector $x \in R^n$,

$$(1-\epsilon)\|x\|_2^2 \leq \|Mx\|_2^2 \leq (1+\epsilon)\|x\|_2^2$$

with probability at least $1-\delta$.

However, unlike the standard Johnson-Lindenstrass matrices (like dense Gaussian random matrices), the matrix M will be very computationally efficient to apply: near-linear time in

n to embed an n-dimensional vector, independent of ϵ or δ. We will show this works with embedding dimension $m = O\left(\frac{\log(n/\delta)\log(1/\delta)}{\epsilon^2}\right)$.

To construct it, we will use a fixed n by n matrix H, traditionally taken to be a "Hadamard matrix" (Hadamard matrices exist only for n powers of two; however, any n can just be rounded up to the nearest power of two, avoiding this problem). We will only care about three properties of H:

1. H is an orthogonal matrix: i.e. $\|Hy\|_2^2 = \|y\|_2^2$ for all y.
2. Every entry of H has absolute value $\frac{1}{\sqrt{n}}$.
3. Hy can be computed in $O(n \log n)$ time.

We will not even use that last property, but it is the reason why this result is useful in the first place!

To determine the random matrix M, we will simply choose a random sample H_S (sampled with replacement) of m rows of H–that is, each row of H_S is just a random row selected from H. We additionally choose a random diagonal sign matrix D (each diagonal entry a random ± 1). We then write

$$M = \sqrt{\frac{n}{m}} H_S D.$$

Note that for any y, $H_S y$ can be immediately computed from H_y, by taking the coordinates corresponding to the selected rows of H. Similarly, the random $H_S y$ can be thought of as being obtained by sampling the coordinates of Hy.

For this problem, assume that $0 < \epsilon, \delta \leq 1/2$.

(a) Show that for any x and *any* fixed diagonal sign matrix D,

$$\mathop{\mathbb{E}}_{H_S}\left[\frac{n}{m}\|H_S D x\|_2^2\right] = \|HDx\|_2^2 = \|x\|_2^2.$$

That is, even with D fixed, the fast Johnson-Lindenstrauss transform gives an unbiased estimator.

(b) (**Optional**) Show that there exists some universal constant C such that with probability at least $1 - \delta/2$, for a random choice of D,

$$\|HDx\|_\infty \leq \frac{C\sqrt{\log(n/\delta)}}{\sqrt{n}}\|x\|_2.$$

This shows that with the random choice of D, no one coordinate will account for much more than its "share" of $\|HDx\|_2^2$, giving us a good setup for sampling the coordinates in the next step. This is the purpose of the D in the construction.

The following tail bound on "Rademacher sums" should be helpful: For any x, if σ_i are independent random signs (± 1),

$$\mathbb{P}\left[\left|\sum_{i=1}^{n} \sigma_i x_i\right| \geq t\right] \leq 2\exp\left(-\frac{t^2}{2\|x\|_2^2}\right).$$

(c) Show that for any fixed choice of D for which the bound from part (b) holds, as long as m is at least some $O\left(\frac{\log(n/\delta)\log(1/\delta)}{\epsilon^2}\right)$ value, then with probability at least $1-\delta/2$ over a random choice of H_S,

$$(1-\epsilon)\|x\|_2^2 \leq \frac{n}{m}\|H_S D x\|_2^2 \leq (1+\epsilon)\|x\|_2^2.$$

Hint: Chernoff!

(d) Putting this all together, show that for a random $M = \sqrt{\frac{n}{m}} H_S D$,

$$(1-\epsilon)\|x\|_2^2 \leq \|Mx\|_2^2 \leq (1+\epsilon)\|x\|_2^2$$

with probability at least $1-\delta$.

Of course, you may assume part (b) even if you did not prove it.

(e) (**Optional and Tricky**) Can you prove a better bound on this method? A natural but tricky variation of the proof here can reduce the extra $\log(n/\delta)$ factor to a $\log(1/(\epsilon\delta))$. There have also been research papers applying far more sophisticated techniques to this.