Lecture 11

Fast Johnson-Lindenstrauss and applications

11.1 Fast Johnson-Lindenstrauss

The Johnson-Lindenstrauss dimension reduction lemma has many algorithmic applications, and we will see some of them today. Typically, the lemma is used to speed up an algorithm by first, applying the dimension reduction map to reduce the dimension of the problem to some small k, and second, solving the low-dimensional problem. For this to be effective it is important that the map can be applied efficiently. Using the construction from the previous lecture, where the dimension reduction matrix was a $k \times d$ matrix G with entries i.i.d. Gaussian, computing the image of a vector $x \in \mathbb{R}^d$ will take time about kd. In some applications x might be a sparse vector, in which case the running time would be $k\|x\|_0$, where $\|x\|_0$ is the number of non-zero entries of x. If $\|x\|_0$ is a constant, this still depends on k, which depending on the problem could be large.

In order to do better, we want to show that the JL lemma holds for some random matrices G that are more structured, and such that Gx can be computed very efficiently. Today we will see that in fact we can choose G to have many entries equal to to 0: each column of G will have only s nonzero entries, where $s = \Theta(\varepsilon^{-1} \log(1/\delta))$ is independent of d and k. We'll see that we can get exactly (up to constant factors) the same guarantees as with the completely random G, but now the computation time for Gx is much faster: it no longer depends on k!

Consider the following construction for a $k \times d$ matrix G. Each column of G is split into s contiguous blocks of size k/s. In each block there will be a single non-zero entry. We use an indicator variable $\eta_{r,i} = 1$ if the r-th entry of the i-th column is nonzero, and $\eta_{r,i} = 0$ otherwise. The nonzero entries will be $\sigma_{r,i}/\sqrt{s}$, where $\sigma_{r,i} \in \{\pm 1\}$ are chosen uniformly at random. Thus we define

$$G_{r,i} = \frac{\eta_{r,i}\sigma_{r,i}}{\sqrt{s}}$$

for every $r \in \{1, \ldots, k\}$ and $i \in \{1, \ldots, d\}$. Before we go on let's verify that this is at least

good in expectation:

$$\begin{aligned}\mathbf{E}\left[\|Gx\|^2\right] &= \mathbf{E}\left[\frac{1}{s}\sum_{r=1}^{k}\sum_{i,j=1}^{d}\eta_{r,i}\eta_{r,j}\sigma_{r,i}\sigma_{r,j}x_i x_j\right] \\ &= \frac{1}{s}\sum_{r=1}^{k}\sum_{i=1}^{d}\eta_{r,i}^2 x_i^2 \\ &= \sum_{i=1}^{d} x_i^2 = \|x\|^2,\end{aligned}$$

where for the second equality we used $\mathbf{E}[\sigma_{r,i}] = 0$ and $\sigma_{r,i}^2 = 1$, and for the third we used that there are exactly s nonzero $\eta_{r,i}$ per column so $\sum_{r=1}^{k}\eta_{r,i} = s$.

We have not yet specified how the locations $\eta_{r,i}$ of the nonzero entries are chosen. There is only one important condition that these entries must satisfy: we need that there are few collisions between non-zero entries in different columns. More precisely we will require that for any $i \neq j \in \{1,\ldots,d\}$,

$$\sum_{r=1}^{k}\eta_{r,i}\eta_{r,j} = O\Big(\frac{s^2}{k}\Big). \tag{11.1}$$

Note that if we chose the nonzero locations uniformly at random within each block then the expected number of collisions per pair of columns is precisely $s \times (s/k) = s^2/k$. Using a Chernoff bound, the probability that there are more than e.g. $2s^2/k$ collisions is exponentially small in s^2/k, so if $s^2/k = \Omega(\log(d/\delta))$ we can apply a union bound and the probability that any pair of columns has more than $2s^2/k$ collisions will be at most δ. So a random construction works, provided the sparsity s, and hence the final computation time, depends logarithmically on d. In fact it is possible to remove this requirement by using a finer analysis (based on a weaker assumption than (11.1)), but we will not show this in this lecture. We're going to cheat a little bit, assume we have a way to choose the $\{\eta_{r,i}\}$ such that (11.1) holds, and show the following:

Theorem 11.1 (Kane-Nelson)**.** *For any integer $d > 0$ and $0 < \varepsilon, \delta < 1/2$ the distribution on $k \times d$ real matrices G described above is such that if $k = \Omega(\varepsilon^{-2}\log(1/\delta))$ and $s = \Omega(\varepsilon^{-1}\log(1/\delta))$ then for any $x \in \mathbb{R}^d$,*

$$\mathbf{Pr}\left((1-\varepsilon)\|x\|^2 \leq \|Gx\|^2 \leq (1+\varepsilon)\|x\|^2\right) > 1-\delta.$$

The main ingredient in the proof is a concentration bound due to Hanson and Wright that applies to quadratic forms: for B a real $n \times n$ matrix we are interested in studying $\sum_{i,j=1}^{n} B_{ij}z_i z_j$ where the $z_i \in \{\pm 1\}$ are random signs. The expectation of this is

$$\mathbf{E}\Big[\sum_{i,j=1}^{n} B_{ij}z_i z_j\Big] = \sum_i B_{ii} = \mathrm{Tr}(B).$$

The following theorem shows that the concentration properties of this expression are governed by two different norms of B: the operator norm $\|B\|$ (the largest singular value), and the Frobenius norm

$$\|B\|_F = \sqrt{\mathrm{Tr}(B^T B)} = \sqrt{\sum_{i,j=1}^{n} B_{i,j}^2}.$$

Theorem 11.2 (Hanson-Wright)**.** *Let $z = (z_1, \ldots, z_n)^T$ be a vector of i.i.d. Rademacher $\{\pm 1\}$ random variables. For any $B \in \mathbb{R}^{n \times n}$ and $p \geq 2$,*

$$\mathbf{E}\left|z^T B z - \mathrm{Tr}(B)\right|^p \leq C^p \max\left\{\sqrt{p}\|B\|_F, p\|B\|\right\}^p,$$

for some universal constant $C > 0$ independent of B, n, p.

Exercise 1. Show that the moment bound stated in Theorem 11.2 is equivalent to the following tail bound: there exists constants $C', C'' > 0$ such that for all $t > 0$,

$$\mathbf{Pr}\left(\left|z^T B z - \mathrm{Tr}(B)\right| > t\right) \leq C' e^{-C'' \min\left(\frac{t^2}{\|B\|_F^2}, \frac{t}{\|B\|}\right)}.$$

Proof of Theorem 11.1. We can write $(Gx)_r = \sum_j \eta_{r,j}\sigma_{r,j}x_j/\sqrt{s}$, and define

$$\begin{aligned} Z &= \|Gx\|^2 - 1 \\ &= \frac{1}{s}\sum_{r=1}^{k}\sum_{i,j} \eta_{r,i}\eta_{r,j}\sigma_{r,i}\sigma_{r,j}x_i x_j - 1 \\ &= \frac{1}{s}\sum_{r=1}^{k}\sum_{i \neq j} \eta_{r,i}\eta_{r,j}\sigma_{r,i}\sigma_{r,j}x_i x_j. \end{aligned}$$

Now the key observation is that we can rewrite this as $\sigma^T B \sigma$, where B is a block-diagonal matrix with k blocks, and each block is $d \times d$ with entries $\eta_{r,i}\eta_{r,j}x_i x_j$ for $i \neq j$, and 0 on the diagonal. In particular we have $\mathrm{Tr}(B) = 0$. To apply Hanson-Wright, we need to estimate the Frobenius norm and the operator norm of B. Let's start with the Frobenius norm:

$$\begin{aligned} \|B\|_F^2 &= \frac{1}{s^2}\sum_{i \neq j} x_i^2 x_j^2 \Big(\sum_{r=1}^{k} \eta_{r,i}\eta_{r,j}\Big) \\ &\leq O\Big(\frac{1}{k}\Big)\|x\|_2^4 \\ &= O\Big(\frac{1}{k}\Big), \end{aligned}$$

where the second line is by assumption (11.1).

Next we bound the operator norm. Since B is block-diagonal it suffices to bound the norm of any block r. We can write the r-th block $B_r = (S_r - D_r)/s$ where $S_r = (\eta_{r,i}\eta_{r,j}x_i x_j)_{i,j}$

and D_r is diagonal with coefficients $\eta_{r,i}^2 x_i^2$ on the diagonal. Now $\|D_r\| \leq \|x\|_\infty^2 \leq 1$ and $\|S_r\| = \|\eta_r x\|^2 \leq \|x\|^2 \leq 1$. Overall, $\|B\| \leq 2/s$.

Applying the (tail version of) the Hanson-Wright inequality,

$$\begin{aligned}\mathbf{Pr}\left(|Z| > \varepsilon\|x\|^2\right) &= \mathbf{Pr}\left(\left|\sigma^T B\sigma - \mathrm{Tr}(B)\right| > \varepsilon\right) \\ &\leq C' e^{-C\min\left(O(\varepsilon^2 k), O(\varepsilon s)\right)} \\ &\leq \delta,\end{aligned}$$

given the choice of s and k made in the theorem. □

11.2 Approximate nearest neighbors

Suppose given n points $x_1, \ldots, x_n \in \mathbb{R}^d$. Given a query $y \in \mathbb{R}^d$, which is the closest x_i to y? A typical algorithm for this problem will have two phases:

(1) *(Preprocessing)* Construct a data structure based on the n points.

(2) *(Query)* Given a new point $y \in \mathbb{R}^d$, query the data structure and return the index of a nearest neighbor.

If d is small, $d = 2$ or $d = 3$, there are some very efficient data structures for this problem, using quasi-linear (in n) space and answering each query in $O(\log n)$ time. But as soon as d grows (think $\log n \ll d \ll n$, for instance $d = n^{0.1}$) the problem suffers from the "curse of dimensionality": either the data structure needs to have size exponential in d, or the time per query must be linear in d.

For instance, the simplest algorithm would simply store the points x_i; given query y it evaluates all distances and returns the closest point. This requires $O(nd)$ space for the data structure and $O(nd)$ time per query. At the opposite end of the spectrum, we can construct a structure based on the Voronoi diagram of the x_i. This will require space $n^{O(d)}$, but only time $O(d \log n)$ per query; you can think of it as a spatial extension of the usual $O(\log n)$ binary search.

Let's see how we can do much better by using dimension reduction based on the Johnson-Lindenstrauss lemma. First, let's settle for ε-approximate nearest neighbors: given y, find i such that $\|y - x_i\| \leq (1+\varepsilon)\min_{1\leq j\leq n}\|y - x_j\|$. Suppose also that the closest point to y has distance 1 (we can always perform binary search to reduce to this case).

Fix a grid on $\mathbb{R}^d$ where each cube has sides of length $\varepsilon/\sqrt{d}$. For each i, let G_i be the set of grid cells that contain at least one point at distance at most 1 from x_i, and store all resulting grid cells in a hash table (where the key is an identification number for the cell, and the value is the index of the point x_i associated to that cell; if a cell has more than one x_i associated to it we can keep only one as the distances will necessarily be almost the same). By a volume argument, the number of grid cells associated to any i is at most $(c/\varepsilon)^d$ for some constant c, so our data structure uses dn (to store the x_i) plus $O(n(c/\varepsilon)^d)$ space.

Given a query y, we simply find the grid cell it is contained in and return the associated point i.

Ok, so we have $dn + O(n(c/\varepsilon)^d)$ space and $O(d)$ query time (to determine the cell and hash it; we can use a simple linear hash function $h(z_1, \ldots, z_d) = (a_1 z_1 + \cdots + a_d z_d \mod p) \mod s$), where p is prime and s is the hash table size). So this is very efficient in terms of query time, but still requires space exponential in d.

But now we can reduce the dimension! Apply the Johnson-Lindenstrauss lemma to project all the x_i to $d' = O(\varepsilon^{-2} \log n)$ dimensions. This gives $n^{O(\log(1/\varepsilon)/\varepsilon^2)}$ space. When a query y is made, we project it in $O(d\varepsilon^{-2} \log n)$ time, and answer the query in $O(d') = O(\varepsilon^{-2} \log n)$ time. For ε constant, we have space polynomial in N and query time $O(d \log n)$: this matches the query time of the basic Voronoi algorithm, but the space requirement is greatly reduced. Unfortunately the dependence on ε is rather bad. Nevertheless, this can be made into a practical algorithm as all operations are very simple, and it is used in practice. Note also the query time is dominated by the time required to compute the Johnson-Lindenstrauss embedding, and it is useful to reduce this: using the fast JL described earlier, if the queries x are sparse vectors we can get a query time that is independent of d and logarithmic in n.

11.3 Approximating matrix products

Suppose $A \in \mathbb{R}^{d \times n}$ and $B \in \mathbb{R}^{d \times m}$ are given. Naïvely the product $A^T B$ takes time $O(ndm)$ to compute. If $n = d = m$ we can do time $O(n^\omega)$, where $\omega \approx 2.373$, using fast matrix multiplication (Strassen's algorithm gives $\log_2 7 \approx 2.807$, and *much more work* gives small improvements); this can also be used to speed up the rectangular case by breaking up A, B in $d \times d$ blocks. Here we're going to see how to compute $A^T B$ approximately, with additive error $\varepsilon \|A\|_F \|B\|_F$ where $\|A\|_F = (\sum_{i,j} A_{i,j}^2)^{1/2}$ is the Frobenius norm. The idea is very simple: we insert a random low-dimensional projection $S \in \mathbb{R}^{d' \times d}$ and return the product $A^T S^T S B$. The whole product can be computed in time $O(ndd' + mdd' + nd'm)$, which is much better as long as $d' \ll n, d, p$.

One natural way to do this is via sampling: write $A^T B = \sum_{i=1}^d a_i b_i^T$ where a_i are the columns of a, and b_i the rows of B. Then we can try to approximate this sum by a random sample. Using a standard concentration bound you can check that to get additive error $\varepsilon \|A\|_F \|B\|_F$ with probability at least $1 - \delta$ it is enough to sample $\Omega(\varepsilon^{-2} \delta^{-1})$ terms. Moreover the dependence on δ can be improved to $\log(1/\delta)$ by using a standard amplification trick (repeat the experiment many times and output the median).

We'll see a way to achieve a similar guarantee using the Johnson-Lindenstrauss dimension reduction technique. Using the fast JL we saw earlier this lets us improve the dependence on ε from ε^{-2} to ε^{-1}. In fact we'll prove something slightly more general:

Theorem 11.3. *Let $\varepsilon, \delta \in (0, 1/2)$ and $\mathcal{D}$ a distribution on $d' \times d$ matrices such that for any unit vector $x \in \mathbb{R}^d$*

$$\mathop{\mathbf{E}}_{S \sim \mathcal{D}} \left| \|Sx\|_2^2 - 1 \right|^\ell \leq \varepsilon^\ell \delta \tag{11.2}$$

for some $\ell \geq 2$. Then for any A, B each having d rows,

$$\Pr_{S\sim\mathcal{D}}\left(\|A^TS^TSB - A^TB\|_F > 3\varepsilon\|A\|_F\|B\|_F\right) < \delta.$$

Recall that in the last lecture we saw that if $\mathcal{D} = \mathcal{D}_{JL}$ is the Johnson-Lindenstrauss distribution then for any unit norm vector x

$$\Pr_{S\sim\mathcal{D}}\left(\left|\|Sx\|^2 - 1\right| > \varepsilon\right) \leq e^{-\varepsilon^2 d'/8}.$$

Using the last problem from Homework 1 it follows that for any $\ell \geq 1$,

$$\mathbf{E}\left|\|S\|_x^2 - 1\right|^\ell \leq (C\sqrt{\ell/d'})^\ell$$

for some constant C. So if we set $d' = \Omega(\log(1/\delta)\varepsilon^{-2})$ and $\ell = \log(1/\delta)$ we get that $\mathcal{D}_{JL}$ satisfies (11.2). The same is true for the sparse Johnson-Lindenstrauss transform we saw earlier.

Proof of Theorem 11.3. The proof is a good application of the moment method for proving concentration bounds. Fix $x, y \in \mathbb{R}^d$ with norm 1. We can write the inner product

$$(Sx)^T(Sy) = \frac{1}{2}\left(\|Sx\|_2^2 + \|Sy\|_2^2 - \|Sx - Sy\|_2^2\right).$$

Recall the definition $\|X\|_\ell = (\mathbf{E}\,|X|^\ell)^{1/\ell}$ for a random variable X. Using the triangle inequality for $\|\cdot\|_\ell$ (this is also called *Minkowski's inequality*),

$$\begin{aligned}
\left\|(Sx)^T(Sy) - x^Ty\right\|_\ell &= \frac{1}{2}\left\|(\|Sx\|_2^2 - 1) + (\|Sy\|_2^2 - 1) - (\|Sx - Sy\|_2^2 - \|x - y\|_2^2)\right\|_\ell \\
&\leq \frac{1}{2}\left(\left\|\|Sx\|_2^2 - 1\right\|_\ell + \left\|\|Sy\|_2^2 - 1\right\|_\ell - \left\|\|Sx - Sy\|_2^2 - \|x - y\|_2^2)\right\|_\ell\right) \\
&\leq \frac{1}{2}\left(\varepsilon\delta^{1/\ell} + \varepsilon\delta^{1/\ell} + \|x - y\|_2^2\,\varepsilon\delta^{1/\ell}\right) \\
&\leq 3\varepsilon\delta^{1/\ell}.
\end{aligned}$$

Let $x_1, \ldots, x_n$ be the columns of A and $y_1, \ldots, y_m$ the columns of B. Define a random variable

$$X_{i,j} = \frac{1}{\|x_i\|_2\|y_j\|_2}\left((Sx_i)^T(Sy_j) - x_i^Ty_j\right).$$

Then $\|A^T S^T SB - A^T B\|_F^2 = \sum_{i,j} \|x_i\|_2^2 \|y_j\|_2^2 X_{i,j}^2$. Using again the triangle inequality,

$$\begin{aligned}
\big\|\|A^T S^T SB - A^T B\|_F^2\big\|_{\ell/2} &= \Big\| \sum_{i,j} \|x_i\|_2^2 \|y_j\|_2^2 \, X_{i,j}^2 \Big\|_{\ell/2} \\
&\le \sum_{i,j} \|x_i\|_2^2 \|y_j\|_2^2 \, \|X_{i,j}^2\|_{\ell/2} \\
&= \sum_{i,j} \|x_i\|_2^2 \|y_j\|_2^2 \, \|X_{i,j}\|_\ell^2 \\
&\le (3\varepsilon\delta^{1/\ell})^2 \Big(\sum_{i,j} \|x_i\|_2^2 \|y_j\|_2^2 \Big) \\
&= (3\varepsilon\delta^{1/\ell})^2 \|A\|_F^2 \|B\|_F^2.
\end{aligned}$$

Finally by Markov's inequality,

$$\begin{aligned}
\mathbf{Pr}\left(\|A^T S^T SB - A^T B\|_F > 3\varepsilon \|A\|_F \|B\|_F \right) &\le \Big(\frac{1}{3\varepsilon \|A\|_F \|B\|_F} \Big)^\ell \mathbf{E} \, \|A^T S^T SB - A^T B\|_F^\ell \\
&\le \delta.
\end{aligned}$$

□