CS 229r: Algorithms for Big Data Fall 2013

Lecture 11— Oct. 8, 2013

Prof. Jelani Nelson *Scribe: Arthur Safira*

1 Overview

In this unit we have been focusing on dimensionality reduction with distortion as its figure of merit. In the last lecture we proved Lipschitz-concentration and the Decoupling lemma, and discussed Alon's lower bound for dimensionality reduction with small distortion ($m \gtrsim \frac{1}{\varepsilon^2 \log \frac{1}{\varepsilon}} \log N$).

In this lecture we will talk about more efficient dimensionality reduction – more efficient in terms of **time**.

1.1 Analyzing the time efficiency of JL

The Johnson-Lindenstrauss Lemma [5] lets us transform high dimensional data to a lower dimension to run our algorithms faster:

$$T(N,n) \overset{\text{dim reduction}}{\overbrace{\rightarrow}} T(N, O(\frac{1}{\varepsilon^2}\log(N))) + \text{Time to perform dim reduction}$$

where $T(N,n)$ denotes the time of the algorithm given N vectors of dimension n. Our goal in this lecture will be to do the best we can choosing a Π such that the time to perform the dimensionality reduction is minimized.

2 Review of our previous choice(s) for Π

Our previous Π had the following:

- $\Pi \in \mathbb{R}^{n \times m}$
- $\Pi_{i,j} = \frac{\pm 1}{\sqrt{m}}$, each independent
- $\text{time}(\Pi x) = O(mn)$
  - More carefully, we can write the time complexity as $O(m\|x\|_0)$, where $\|x\|_0$ is the size of the support of x. This might seem nit-picky, but it is not uncommon for x to be quite sparse. For example, some effective text processing machine learning algorithms keep vectors of dimension D = number of words in the dictionary, and keep track of the frequency of words in an e-mail to discern whether or not a given message is spam.

In the previous lecture, although we proved JL with Π as described above, we mentioned many other choices would work, too, such as one with independent random entries from $N(0,1)$ (We also gave some more general conditions for what probability distributions could be used for our matrices — check out the previous lecture for more info). None of these, however, help the above runtime.

3 Let's find a better Π

It's clear that one way we could win is doing a better job in choosing Π; it would be great if we could choose one that had more 0's so that we would be multiplying numbers a a whole lot less.

This and next lecture we will have two approaches for two different cases:

1. Choosing Π to be fast for sparse vectors
2. Choosing Π to be fast for dense vectors

3.1 What Π is fast for sparse vectors?

As just hinted at, the best way we know to deal with sparse vectors is to make Π itself a sparse matrix.

History Table

Reference	m	s	**Notes**
JL & others [5]	$\approx \frac{4}{\varepsilon^2}\ln(1/\delta)$	m	Last Lecture
Achlioptas [1]	$\approx \frac{4}{\varepsilon^2}\ln(1/\delta)$	$m/3$	Random Sign matrix with 2/3 prob of zero'ing each matrix element; $m/3$ is an expectation.
Thorup & Zhang [9]	$\frac{1}{\varepsilon^2\delta}$	1	First Problem Set
Dasgupta, Kumar, & Sarlós [4]	$O(\frac{1}{\varepsilon^2}\ln(1/\delta)$	$\tilde{O}(\frac{1}{\varepsilon}\log^2(1/\delta))$[1]	$\tilde{O}(\cdot)$ hides $\log^{O(1)}(1/\varepsilon)$
Kane, Nelson [7]	$O(\frac{1}{\varepsilon^2}\ln(1/\delta)$	$O(\frac{1}{\varepsilon}\log(1/\delta))$	

where m is the target dimension and s is number of non-zero entries per column of the matrix. With s non-zero entries, we could multiply Πx in time $\text{time}(\Pi x) = O(s\|x\|_0)$.

What is Π? We have two options:

1. Π_1: Split each column of Π into m/s blocks of size s. For each of these blocks, choose exactly one entry to be a (normalized) random sign ($\sigma = \pm 1/\sqrt{s}$), and set the rest of the matrix elements in the block to be 0.
2. Π_2: For each column, choose s entries (**without replacement**) to place a random sign r.v. (again, $\sigma = \pm 1/\sqrt{s}$).

As far as implementations are concerned, the first of these is a bit simpler as we can simply make use of hash functions $h : [n] \times [s] \to [m/s]$ and $\sigma : [n] \times [s] \to \{\pm 1\}$. Dealing with the second one is more of a hassle.

[1] [4] showed $s = \tilde{O}(\varepsilon^{-1}\log^3(1/\delta))$, but tighter analyses were later given of the same construction improving the cubic dependence to quadratic [2, 6].

3.2 Dealing with Π

From here onwards, we drop the index.

Quick Notes:

- For $s = m$, both Π_1 and Π_2 are the Thorup Zhang sketch [9].
- For general s, Π_1 is just CountSketch: The matrix describes the hash functions of a particular countsketch matrix with s rows and m/s columns [3].

3.2.1 Analysis

Claim: We can set m and s such that

$$m = O(1/\varepsilon^2 \log(1/\delta)) \text{ and } s = O(\frac{1}{\varepsilon} \log(1/\delta))$$

and satisfy the usual $(1 \pm \varepsilon)$ distortion properties of the mapping Π w.p. $1 - \delta$.

Before we prove this claim, we mention some bad news in regard to our efforts here.

"Bad News" Claim: For all $N > 1$ there exists $N + 1 = n + 1$ vectors in $\mathbb{R}^n$ such that any $\Pi \in \mathbb{R}^{m \times n}, m = O(\frac{1}{\varepsilon^2} \log(N))$ preserving all pairwise Euclidean distances up to $1 + \varepsilon$ and with s non-zeros per column must have $s = \Omega(\frac{1}{\varepsilon} \frac{\log(N)}{\log(\frac{1}{\varepsilon})})$ as long as $m = O(\varepsilon^{-2} \log N)$ (and $m = O(\frac{n}{\log 1/\varepsilon})$) [8]. Note we can't let m get too close to n for the lower bound since once $m = n$ the identity matrix works and has $s = 1$.

In other words, we have a limitation in how many non-zero elements we can reduce down to if we want to reduce the dimension down to $m = O(\frac{n}{\log 1/\varepsilon})$.

On the bright side, let's move towards proving the claim.

3.2.2 Proof of Claim

Note we can write the elements of Π as $\Pi_{i,j} = \frac{1}{\sqrt{s}} \delta_{i,j} \sigma_{i,j}$ where $\sigma_{i,j}$ is a random sign and $\delta_{i,j} \in \{0, 1\}$. Without loss of generality, we can set $\|x\|_2 = 1$. Then, each component of Πx is given by

$$(\Pi x)_r = \frac{1}{\sqrt{s}} \sum_{i=1}^{n} \delta_{r,i} \sigma_{r,i} x_i.$$

Thus, we can write the norm squared as

$$\|\Pi x\|^2 = \frac{1}{s}\sum_{r=1}^{m}\sum_{i,j=1}^{n}\delta_{r,i}\delta_{r,j}\sigma_{r,i}\sigma_{r,j}x_i x_j$$

$$= \frac{1}{s}\sum_{r=1}^{m}\left(\sum_{i=1}^{n}\delta_{r,i}^2\cancel{\sigma_{r,i}^2}x_i^2 + \sum_{\substack{i,j=1\\ i\neq j}}^{n}\delta_{r,i}\delta_{r,j}\sigma_{r,i}\sigma_{r,j}x_i x_j\right)$$

$$\|\Pi x\|^2 - 1 = \overbrace{\frac{1}{s}\sum_{r=1}^{m}\sum_{\substack{i,j=1\\ i\neq j}}^{n}\delta_{r,i}\delta_{r,j}\sigma_{r,i}\sigma_{r,j}x_i x_j}^{Z}$$

Where we used the fact that

$$\frac{1}{s}\sum_{r=1}^{m}\sum_{i=1}^{n}\delta_{r,i}^2 x_i^2 = \frac{1}{s}\sum_{i=1}^{n}\sum_{r=1}^{m}\delta_{r,i}^2 x_i^2 = \frac{1}{\cancel{s}}\sum_{i=1}^{n}\cancel{s}x_i^2 = 1$$

since $\|x\| = 1$ and we have that there are exactly s non-zero elements per column.

Not that at this point we have brought the issue down to characterizing Z. We can interpret Z as an error for how far off $\|\Pi x\|$ is from 1. We would like to show something like

$$\mathbb{P}(|Z| > \varepsilon) < \delta.$$

Let's write

$$\|\Pi x\|^2 - 1 = \sigma^T A_x \sigma - \mathbb{E}(\sigma^T A_x \sigma).$$

Note that $\mathbb{E}(\sigma^T A_x \sigma)$ will exactly be the sum of diagonal terms (convince yourself!). This situation should look really familiar now; we had a similar (but not exactly the same!) situation last lecture. In the last lecture, we had A_x being a block diagonal matrix with sub matrices xx^T along the diagonal. We *almost* have this, but since so many of our entries are actually 0, we actually have block sub matrices of the form $x^{(r)}x^{(r)T}$ along the diagonal (but where we zero out the diagonal entries of the matrix), with

$$x^{(r)} = (\delta_{r,1}x_1, ..., \delta_{r,n}x_n)^T$$

How are we going to prove the tail bound on $\|\Pi x\|^2 - 1$? The same was as we did it last lecture, using the Hanson-Wright inequality:

$$\mathbb{P}(\|\Pi x\| - 1 > \varepsilon) = \mathbb{P}(|\sigma^T A_x \sigma - \mathbb{E}\sigma^T A\sigma| > \varepsilon) \lesssim \exp\left(-\min\left\{\frac{c\lambda^2}{\|A\|_F^2}, \frac{c\lambda}{\|A\|}\right\}\right)$$

Let's define a "good" event E as one such that for all $i \neq j \in [n], \sum_{r=1}^{m}\delta_{r,i}\delta_{r,j} = O(s^2/m)$. What this encapsulates is the number of times two separate columns will "collide"; that is, if we place s non-zero elements in our matrix in each of the columns i and j, we expect that for each non-zero element that the probability to collide with it is s/m, and we have s opportunities to do this so

that the expectation is $O(s^2/m)$. We expect this to be true, but our proof will actually depend on it being true; as such, we will later place bounds on the probability we have more than (for example) 5 times more collisions than this expectation include that in our analysis.

With the good event as above, we can write

$$\begin{aligned}
\|A_x\|_F^2 &= \frac{1}{s}\sum_{r=1}^{m}\sum_{i\neq j}\delta_{r,i}\delta_{r,j}x_i^2x_j^2 \\
&= \frac{1}{s^2}\sum_{i\neq j}x_i^2x_j^2\sum_{r=1}^{m}\delta_{r,i}\delta_{r,j} \\
&\leq O(s^2/m)\frac{1}{s^2}\overbrace{\sum_{i\neq j}x_i^2x_j^2}^{\leq(\sum x_i^2)^2=1} \\
&\leq O(\frac{1}{m})
\end{aligned}$$

Whew. Let's consider the operator norm now, $\|A_x\|$. The matrix A_x has the form of a block diagonal matrix; we can write

$$\|A_x\| = \frac{1}{s}\max_{r\in[m]}\|A_r\|.$$

where we pulled out the overall factor of $1/s$. We can bound this by noticing $A_r = S_r - D_r$, with $S_r = x^{(r)}x^{(r)T}$ and D_r diagonal with entries $\delta_{r,1}x_1^2, ..., \delta_{r_n}x_n^2$. We can write

$$\overbrace{\|A_r\| \leq \|S_r\| + \|D_r\|}^{\text{triangle ineq.}} \leq 1 + 1 = 2$$

since

$$\|D_r\| = \max_i \delta_{r,i}x_i^2 \leq \|x\|_\infty^2 \leq 1$$

and $\|S_r\|$ has only one eigenvector with non-zero eigenvalue, $x^{(r)}$,so that $\|S_r\| = \|x^{(r)}\|^2 \leq \|x\|^2 = 1$. Thus, $\|A_x\| \leq 2/s$.

Putting this all together, we can write that, **conditioned on E**,

$$\mathbb{P}_\sigma(\|\Pi x\|^2 - 1 > \varepsilon) \leq \max\left\{\exp(-c\varepsilon^2 m), \exp(-c\varepsilon(s/2))\right\}$$

So that we can choose $m = \Theta(\frac{1}{\varepsilon^2}\log(1/\delta))$ and $s = \Theta(\frac{1}{\varepsilon^2}\log(1/\delta))$. Remember, however, that we aren't done; we need to deal with the event we conditioned on in order to get the bound on m!

3.2.3 Analysis of the event E

Fix (i, j) and suppose $\delta_{r_1,i}, ..., \delta_{r_s,i} = 1$. Define indicator r.v.'s $X_1, ..., X_s$, with

$$X_k = \mathbf{1}_{\{\delta_{r_k,j}=1\}}.$$

Then,

$$\sum_{i=1}^{m} \delta_{r,i}\delta_{r,j} = \sum_{k=1}^{s} X_k$$

Now we can apply the Chernoff bound to get error probability $\gamma = \delta/n^2$ as long as $s^2/m \geq c\log(1/\gamma)$, or

$$s \gtrsim \sqrt{m\log(n/\delta)} = \frac{1}{\varepsilon}\sqrt{\log(1/\delta)\log(n/\delta)}$$

This setting of γ is to ensure, by a union bound, that no pair $i \neq j$ has too many collisions (and there are $\binom{n}{2}$ such pairs). Finally, we can first apply the Thorup-Zhang matrix with $s = 1$ and $m = O(1/\varepsilon^2\delta)$; then we can write out effective Π to be the Π we have been describing all long multiplying the TZ matrix. In doing so, we can replace n by $O(1/\varepsilon^2\delta)$. To remove the $\sqrt{\log(1/\varepsilon)}$,

$$\|\Pi x\|^2 - 1 = Z = \frac{1}{s}\sum_{r=1}^{m}\sum_{i\neq j} \delta_{r,i}\delta_{r,j}\sigma_{r,i}\sigma_{r,j}x_i x_j = \frac{1}{s}\sum_{r=1}^{m} Z_m$$

We then bound the probability that Z is large with

$$\begin{aligned}
\mathbb{P}(Z > \varepsilon) &= \mathbb{P}(Z^\ell > \varepsilon^\ell) \\
&< \frac{1}{\varepsilon^\ell}\mathbb{E}(Z^\ell) \\
&= \frac{1}{\varepsilon^\ell s^\ell}\sum_{q=1}^{\ell}\sum_{\substack{r_1,\ldots,r_q \\ \ell_1,\ldots,\ell_q \\ \sum \ell_i = \ell}} \binom{\ell}{\ell_1,..,\ell_q}\mathbb{E}\left(\prod_{j=1}^{q} Z_{r_j}^{\ell_j}\right)
\end{aligned}$$

where we simply expanded into all of the terms when u expand $Z^\ell = ((1/s)\sum Z_r)^\ell$ and we used linearity of expectation. Without going into all the details, it turns out we can write $\mathbb{E}(\prod_{j=1}^{q} Z_{r_j}^{\ell_j}) \leq \prod_{j=1}^{q}\mathbb{E}(Z_{r_j}^{\ell_j})$, and the proof boils down to bounding this last expectation.

4 Moral

One moral of this story is that in a proof, you want to argue as much as you can without using too many "black boxes". In our proof here,the Hanson-Wright inequality is a black box that we used to bound our error. In the proof of this last part, we proved the bounds on the expectation from first principles [7]. In fact it is possible to get the correct bound using Hanson-Wright, but you should not condition on event E. Rather you should show using Markov's inequality on a high moment to show that the Frobenius norm squared $\|A_x\|_F^2$ is small with high probability, but then the calculations you end up doing become essentially identical to just reasoning about the moments of Z itself from first principles.

If a paper uses a lot of "black boxes" to reach their results, it's not unreasonable to consider that there might be better result if one did not use such heavy machinery and instead solved things using

only first-principles tools. Or at least, understand when each of the black boxes is or isn't tight in various applications (maybe something too powerful is being used and is thus causing suboptimal results).

References

[1] Dimitris Achlioptas. Database-friendly random projections. *J. Comput. Syst. Sci.* 66(4): 671–687, 2003.

[2] Vladimir Braverman, Rafail Ostrovsky, Yuval Rabani. Rademacher Chaos, Random Eulerian Graphs and The Sparse Johnson-Lindenstrauss Transform. *CoRR* abs/1011.2590, 2010.

[3] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theor. Comput. Sci., 312* (1):3–15, 2004.

[4] Anirban Dasgupta, Ravi Kumar, Tams Sarls. A sparse Johnson-Lindenstrauss transform. *STOC*, pgs. 341–350, 2010.

[5] William B. Johnson, Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space Contemporary Mathematics 26, 1984.

[6] Daniel M. Kane, Jelani Nelson. A Derandomized Sparse Johnson-Lindenstrauss Transform. *CoRR* abs/1006.3585, 2010.

[7] Daniel M. Kane, Jelani Nelson. Sparser Johnson-Lindenstrauss transforms. *SODA*, pgs. 1195–1206, 2012.

[8] Jelani Nelson, Huy L. Nguyễn Sparsity Lower Bounds for Dimensionality Reducing Maps. *STOC*, pgs. 101–110, 2013.

[9] Mikkel Thorup, Yin Zhang. Tabulation-Based 5-Independent Hashing with Applications to Linear Probing and Second Moment Estimation. *SIAM J. Comput.* 41(2): 293–331, 2012.