CS 229r: Algorithms for Big Data Fall 2013

Lecture 13 — October 15, 2013

Prof. Jelani Nelson *Scribe: Sam Elder*

1 Introduction and Overview

We're starting a new topic: Numerical linear algebra. We'll see some of related topics, too. We'll spend about four lectures on this before moving to compressed sensing.

Today, we'll look at approximate matrix multiplication. Next lecture we'll get to (oblivious) subspace embeddings and least squares regression. For all of these, we'll develop randomized approximate algorithms that compute what they're supposed to compute faster than deterministic exact algorithms for the problems.

Let's start by talking about matrix multiplication. We have matrices A, which is $n \times r$, and B, which is $n \times p$, and we want to compute $A^T B$. (This awkward formulation will make some later lemmas seem natural.)

The standard algorithm takes time rnp, which just has three for loops. You can do faster. If A and B are both square, i.e. $r = n = p$, then you can do it in time $O(n^\omega)$, where $\omega < 2.373\ldots$. Strassen [7] was the first one doing better, and he got $O(n^{\log_2 7})$ by some clever recursion. Coppersmith and Winograd [2] in the 1980s got a better exponent, and there have been some recent improvements by Williams [9] in 2011 and Stothers [6] in 2010. No one uses these, and people still only occasionally use Strassen.

2 Approximate Matrix Multiplication

That's square matrices and exact computation, so let's look at approximate matrix multiplication. What do we want?

We want to quickly find matrices $\tilde{A}$, which is $m \times r$ and $\tilde{B}$ which is $m \times p$, where $m \ll n$, such that $\left\|\tilde{A}^T\tilde{B} - A^T B\right\|$ is small. In this lecture, that norm will be the Frobenius norm, and small will be $\epsilon \|A\|_F \|B\|_F$. Recall that $\|X\|_F = (\sum_{i,j} X_{ij}^2)^{1/2}$ (treat the matrix as a vector and take its ℓ_2 norm).

2.1 Sampling Algorithm

We'll look at two algorithms for matrix multiplication. The first one will use sampling and is due to Drineas, Kannan, and Mahoney [3]. Here's the approach: Notice that if A consists of rows $x_1^T, \ldots, x_n^T$ and B consists of rows $y_1^T, \ldots, y_n^T$ (so each of these vectors are column vectors), then

$A^T B = \sum_{k=1}^n x_k y_k^T$. Indeed,

$$(A^T B)_{ij} = \sum_k A_{ki} B_{kj} = \sum_{k=1}^n (x_k)_i (y_k)_j = (\sum_{k=1}^n x_k y_k^T)_{ij}.$$

The idea is that we have a these n outer products, and we'll only sample $m \ll n$ of them. We won't use the uniform distribution, though: We'll choose $\tilde{A}^T \tilde{B} = \dfrac{1}{m} \sum_{t=1}^m \dfrac{x_{k_t} y_{k_t}^T}{p_{k_t}}$, where p_k is the probability that we sample out product $x_k y_k^T$.

In terms of matrices, $\tilde{A} = \Pi A$ and $\tilde{B} = \Pi B$, where Π has exactly one nonzero entry in each row, chosen according to some distribution. If we pick a column k with probability p_k, then we'll put a $1/\sqrt{m p_k}$ entry in that row. The probability distribution isn't uniform; we'll take $p_k \propto \|x_k\|_2 \|y_k\|_2$. Notice that if you were to implement this as a streaming algorithm, you'd need two passes over the data, the first one to determine the p_k, and the second to pick them with that probability. Perhaps it could be solved with some variant of reservoir streaming though.

The claim is that this is going to be an unbiased estimator and that its variance is small. We can't exactly improve this probability by doing a bunch of trials and taking the median since these are matrices, but there is a way we can boost it anyways.

So we have $\tilde{A}^T \tilde{B} = \frac{1}{m} \sum_{t=1}^m \frac{x_{k_t} y_{k_t}}{p_{k_t}}$. Call the summand Z_t. We'll look at $\mathbb{E}\, Z_t$, and that'll tell us about the expectation of this sum by linearity of expectation. We have

$$\mathbb{E}\, Z_t = \frac{1}{m} \sum_{k=1}^n \frac{\mathbb{P}(k_t = k) x_k y_k^T}{p_k} = \frac{1}{m} A^T B,$$

so the expectation is correct. Now the variance calculation. We have

$$\begin{aligned}
\mathbb{E} \left\| \tilde{A}^T \tilde{B} - A^T B \right\|_F^2 &= \sum_{i=1}^r \sum_{j=1}^p \mathbb{E}((\tilde{A}^T \tilde{B})_{ij} - (A^T B)_{ij})^2 \\
&= \sum_{i,j} \operatorname{Var} \left(\sum_{t=1}^m (Z_t)_{ij} \right) = \sum_{i,j} m \operatorname{Var}((Z_t)_{ij})
\end{aligned}$$

since the Z_t are independent. And we have

$$\begin{aligned}
\operatorname{Var}((Z_t)_{ij}) &\le \mathbb{E}(Z_t)_{ij}^2 = \frac{1}{m^2} \sum_{k=1}^n \frac{p_k (x_k)_i^2 (y_k)_j^2}{p_k^2} \\
\mathbb{E} \left\| \tilde{A}^T \tilde{B} - A^T B \right\|_F^2 &\le \frac{1}{m} \sum_{i,j} \sum_{k=1}^n \frac{(x_k)_i^2 (y_k)_j^2}{p_k} \\
&= \frac{1}{m} \sum_{k=1}^n \frac{1}{p_k} \|x_k\|^2 \|y_k\|^2 \\
&= \frac{1}{m} \left(\sum_{k=1}^n \|x_k\| \|y_k\| \right)^2 \text{ (substituting the definition of } p_k) \\
&\le \frac{1}{m} \left(\sum_{k=1}^n \|x_k\|^2 \right) \left(\sum_{k=1}^n \|y_k\|^2 \right) \text{ (Cauchy-Schwarz)}
\end{aligned}$$

$$= \frac{1}{m} \|A\|_F^2 \|B\|_F^2 .$$

Now when we apply Chebyshev, we get

$$\mathop{\mathbb{P}}_{\Pi} \left(\left\| \tilde{A}^T \tilde{B} - A^T B \right\|_F > \epsilon \|A\|_F \|B\|_F \right) < \frac{\mathbb{E} \left\| \tilde{A}^T \tilde{B} - A^T B \right\|_F^2}{\epsilon^2 \|A\|_F^2 \|B\|_F^2} < \frac{1}{\epsilon^2 m}.$$

In conclusion, we only need to sample $m = O(1/\epsilon^2)$ outer products to get success with probability 9/10.

So we said we could bootstrap the failure probability to δ efficiently. We could just make $m = O(1/\epsilon^2\delta)$ but we can also do it more efficiently, a $\log(1/\delta)$ multiplier.

We'll see a trick due to Clarkson and Woodruff [1] from STOC '09 (quite recent!). We'll pick $\Pi_1, \ldots, \Pi_t$ where $t = O(\log 1/\delta)$ and form t matrix products $\tilde{A}_1^T \tilde{B}_1, \ldots, \tilde{A}_t^T \tilde{B}_t$. In the past, we had elements and we took the median, but these are matrices. We'd like to compute $\left\| \tilde{A}^T \tilde{B} - A^T B \right\|_F$ and see if that's small, but that would involve computing $A^T B$. So instead, we'll compare them with each other: Pick the first j you find such that

$$\left\| \tilde{A}_j^T \tilde{B}_j - \tilde{A}_i^T \tilde{B}_i \right\|_F < \frac{\epsilon}{2} \|A\|_F \|B\|_F$$

for more than half of the i's. This is something like choosing a median.

Why does this work? Well, with probability $1-\delta$, more than half of the j's have $\left\| \tilde{A}_j^T \tilde{B}_j - A^T B \right\|_F < \frac{\epsilon}{4} \|A\|_F \|B\|_F$, by the Chernoff bound. Then we just use the triangle inequality to know that for all such i, j,

$$\left\| \tilde{A}_j^T \tilde{B}_j - \tilde{A}_i^T \tilde{B}_i \right\|_F \leq \left\| \tilde{A}_j^T \tilde{B}_j - A^T B \right\|_F + \left\| \tilde{A}_i^T \tilde{B}_i - A^T B \right\|_F \leq \frac{\epsilon}{2} \|A\|_F \|B\|_F ,$$

so any such j will be counted. Could another one trick us? Nope: If more than half of the i's have $\tilde{A}_i^T \tilde{B}_i$ close to some $\tilde{A}_j^T \tilde{B}_j$, then at least one of these will be close to $A^T B$, and this implies by the triangle inequality that any success will be within $\frac{3\epsilon}{4} \|A\|_F \|B\|_F$ of $A^T B$, as desired.

2.2 Dimensionality Reduction-based Algorithm

Now let's see another way to do approximate matrix multiplication which is related to something we've seen previously in this class: the Johnson-Lindenstrauss (dimensionality reduction) lemma. Using JL for approximate matrix multiplication was first explored by Sarlós [5], but we'll present the definitions and analysis from [4] since it obtains sharper results by a logarithmic factor (Clarkson and Woodruff [1] also improved the logarithmic factor in the special case where the JL matrix of interest is a scaled random sign matrix).

Definition 1. *A distribution D over $\mathbb{R}^{m \times n}$ is said to have the* (ϵ, δ, p)-JL moment property *if for every $x \in \mathbb{R}^n$ with $\|x\| = 1$,*

$$\mathop{\mathbb{E}}_{\Pi \sim D} \left| \|\Pi x\|^2 - 1 \right|^p < \epsilon^p \delta.$$

Note that by Markov, this gives us

$$\mathop{\mathbb{P}}_{\Pi}\left(\left|\|\Pi x\|^2-1\right|>\epsilon\right)<\frac{1}{\epsilon^p}\mathop{\mathbb{E}}_{\Pi}\left|\|\Pi x\|^2-1\right|^p<\delta.$$

Having a moment bound looks a little bit stronger than just having a tail bound, but often times it isn't. Sometimes you prove some tail bound looking like

$$\forall\epsilon>0,\ \mathop{\mathbb{P}}_{\Pi\sim D}(\left|\|\Pi x\|^2-1\right|>\epsilon)<\exp(-c(\epsilon^2m+\epsilon m)).$$

However, this statement is actually equivalent to D having the $(\epsilon,\exp(-c(\epsilon^2m+\epsilon m)),\min\{\epsilon,\epsilon^2\}m)$-JL moment property for all $\epsilon>0$. One direction is just Markov as above, but for the other direction, you can use integration by parts. Let Z be a nonnegative random variable and φ its pdf. Then

$$\mathbb{E}\,Z^p=-\int_0^\infty\epsilon^p(-\varphi(\epsilon))\,d\epsilon=[\epsilon^p(1-\Phi(\epsilon))]|_0^\infty+p\int_0^\infty\epsilon^{p-1}(1-\Phi(\epsilon))\,d\epsilon=p\int_0^\infty\epsilon^{p-1}(1-\Phi(\epsilon))\,d\epsilon.$$

Note $1-\Phi(\epsilon)$ (where Φ is the cdf) is just $\mathbb{P}(Z>\epsilon)$, thus inserting the tail bound above gives a moment bound. Notice that for the moment bound, you need a tail bound like this for every ϵ, because you need to integrate. We won't focus on this too much, but just remember that tail bounds for every ϵ are equivalent to moment bounds (of this form) for every ϵ.

Theorem 2. *Suppose that D satisfies the (ϵ,δ,p)-JL moment property for some $p\geq 2$. Then for every A,B with matching numbers of rows,*

$$\mathop{\mathbb{P}}_{\Pi\sim D}\left(\left\|(\Pi A)^T(\Pi B)-A^TB\right\|_F>3\epsilon\left\|A\right\|_F\left\|B\right\|_F\right)<\delta.$$

Proof. The idea is that the JLMP implies that you preserve vectors, and we'll show that this implies you preserve dot products, and then that implies that you preserve matrix products. Arguing in terms of moments instead of tail bounds let's us exploit Minkowski's inequality (namely that $\|\cdot\|_p$ satisfies triangle inequality). Arguing in terms of tail bounds tempts you to use the union bound to say all entries of $(\Pi A)^T(\Pi B)-A^TB$ are preserved simultaneously, but this leads to worse bounds.

So suppose that for $a,b\in\mathbb{R}^n$, $\|a\|=\|b\|=1$. Then $\|a-b\|^2=\|a\|^2+\|b\|^2-2\langle a,b\rangle$ and $\|\Pi a-\Pi b\|^2=\|\Pi a\|+\|\Pi b\|^2-2\langle\Pi a,\Pi b\rangle$. Therefore, under distribution D,

$$\begin{aligned}\left\|\langle\Pi a,\Pi b\rangle-\langle a,b\rangle\right\|_p&=\frac{1}{2}\left\|(\|\Pi a\|^2-1)+(\|\Pi b\|^2-1)+(\|a-b\|^2-\|\Pi a-\Pi b\|^2)\right\|_p\\&\leq\frac{1}{2}\left[\left\|\|\Pi a\|^2-1\right\|_p+\left\|\|\Pi b\|^2-1\right\|_p+\|a-b\|^2\left\|\left\|\Pi\left(\frac{a-b}{\|a-b\|}\right)\right\|^2-1\right\|_p\right]\\&<\frac{1}{2}[\epsilon\delta^{1/p}+\epsilon\delta^{1/p}+4\epsilon\delta^{1/p}]=3\epsilon\delta^{1/p}.\end{aligned}$$

Now, let's look at A^TB. Again write A as having rows $x_1^T,\ldots,x_n^T$ and B having rows $y_1^T,\ldots,y_n^T$. Define $X_{ij}=\frac{1}{\|x_i\|\|y_j\|}(\langle\Pi x_i,\Pi y_j\rangle-\langle x_i,y_j\rangle)$. So we can write

$$\left\|(\Pi A)^T(\Pi B)-A^TB\right\|_F^2=\sum_{i=1}^r\sum_{j=1}^p\|x_i\|^2\|y_j\|^2X_{ij}^2.$$

Since $p \geq 2$, $\|\cdot\|_{p/2}$ is still a norm, so we apply the triangle inequality to get

$$\begin{aligned}\left\| \left\|(\Pi A)^T(\Pi B) - A^T B\right\|_F^2 \right\|_{p/2} &= \left\| \sum_{i,j} \|x_i\|^2 \|y_j\|^2 X_{ij}^2 \right\|_{p/2} \\ &\leq \sum_{i,j} \|x_i\|^2 \|y_j\|^2 \left\|X_{ij}^2\right\|_{p/2} \\ &< (3\epsilon\delta^{1/p})^2 \sum_{i,j} \|x_i\|^2 \|y_j\|^2\end{aligned}$$

Also note

$$\mathbb{E}\left\|(\Pi A)^T(\Pi B) - A^T B\right\|_F^p = \left\| \left\|(\Pi A)^T(\Pi B) - A^T B\right\|_F^2 \right\|_{p/2}^{p/2} < (3\epsilon\delta^{1/p} \|A\|_F \|B\|_F)^p.$$

Now we just apply Markov to get a tail bound.

$$\mathbb{P}\left(\left\|(\Pi A)^T(\Pi B) - A^T B\right\|_F > 3\epsilon \|A\|_F \|B\|_F\right) < \frac{\mathbb{E}\left\|(\Pi A)^T(\Pi B) - A^T B\right\|_F^p}{(3\epsilon \|A\|_F \|B\|_F)^p} < \delta. \qquad \square$$

So we've proved that if we have a (ϵ, δ, p)-JLMP distribution, that's good enough. How do we pick such a distribution? Any JL distribution works. For example, we can pick the Π from PSet 1, Problem 3 (which is due to Thorup and Zhang [8]), with a random sign in each column and m rows, where $m = O(1/\epsilon^2)$ to get a success probability $2/3$; this is an $(\varepsilon, 2/3, 2)$-JLMP. This is nice because it's a 1-pass algorithm, and we've reduced the problem to something we already have some ideas for.

3 Further Numerical Linear Algebra

Next time, we'll look at subspace embeddings.

Definition 3. *If $V \subseteq \mathbb{R}^n$ is a dimension-d linear subspace, we say that Π is an ϵ* subspace embedding for V *if for every $x \in V$, $\|\Pi x\| = (1 \pm \epsilon)\|x\|$.*

We'll see how subspace embeddings relate to a lot of things we've seen. Problem 1 on the current PSet (due Thursday) shows that for any such V there is a set of $N = \exp(cd)$ vectors such that if Π satisfies the JL lemma conditions on those N vectors then Π is a subspace embedding for V.

References

[1] Kenneth Clarkson, David Woodruff. Numerical Linear Algebra in the Streaming Model. *STOC '09* 205–214, 2009.

[2] Don Coppersmith, Shmuel Winograd. Matrix Multiplication via Arithmetic Progressions. *STOC '87* 1–6, 1987.

[3] Petros Drineas, Ravi Kannan, Michael Mahoney. Fast Monte Carlo Algorithms for Matrices I: Approximating Matrix Multiplication. *SIAM J. Computing* 36, 132–157, 2006.

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

[5] Tamás Sarlós. Improved Approximation Algorithms for Large Matrices via Random Projections. *FOCS*, 143–152, 2006.

[6] Andrew James Stothers. On the Complexity of Matrix Multiplication. *University of Edinburgh PhD Thesis*, 2010.

[7] Volker Strassen. Gaussian Elimination is not Optimal. *Numer. Math.*, 13:354–356, 1969.

[8] 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.

[9] Virginia Vassilevska Williams. Multiplying Matrices Faster than Coppersmith-Winograd. *STOC '12* 887–898, 2012.