CS 229r: Algorithms for Big Data		Fall 2015
	Lecture 17 — 10/28	
Prof. Jelani Nelson		*Scribe: Morris Yau*

1 Overview

In the last lecture we defined subspace embeddings a *subspace embedding* is a linear transformation that has the Johnson-Lindenstrauss property for all vectors in the subspace:

Definition 1. *Given $W \subset \mathbb{R}^n$ a linear subspace and $\varepsilon \in (0,1)$, an ε-**subspace embedding** is a matrix $\Pi \in \mathbb{R}^{m \times n}$ for some m such that*

$$\forall x \in W : (1-\varepsilon)\|x\|_2 \leq \|\Pi x\|_2 \leq (1+\varepsilon)\|x\|_2$$

And an oblivious subspace embedding

Definition 2. *An (ϵ, δ, d) oblivious subspace embedding is a distribution D over R^{mxn} such that $\forall U \in R^{mxn}$, $U^T U = I$*

$$P_{\Pi \sim D}(\|(\Pi U)^T(\Pi U)\| > \epsilon) < \delta$$

In this lecture we go over ways of getting oblivious subspace embeddings and then go over applications to linear regression. Finally, time permitting, we will go over low rank approximations.

2 General Themes

Today:

- ways of getting OSE's
- More regression
- Low rank approximation

We can already get OSE's with Gordon's theorem. The following are five ways of getting OSE's

- net argument
- noncommutative kintchine with matrix chernoff
- moment method
- approximate matrix multiplication with Frobenius error
- chaining

2.1 Net Argument

Concerning the net argument which we'll see the details in the pset. For any d-dimensional subspace $E \in R^n$ there exists a set $T \subset E \cap S^{n-1}$ of size $O(1)^d$ such that if Π preserves every $x \in T$ up to $1 + O(\epsilon)$ then Π preserves all of E up to $1 + \epsilon$

So what does this mean, if we have distributional JL than that automatically implies we have an oblivious subspace embedding. We would set the failure probability in JL to be $\frac{1}{O(1)^d}$ which by union bound gives us a failure probability of OSE of δ.

2.2 Noncommutative Khintchine

For Noncommutative Khintchine let $\|M\|_p = (\mathbb{E}\,\|M\|_{S_p}^p)^{\frac{1}{p}}$ with $\sigma_1, ..., \sigma_n$ are $\{1, -1\}$ independent bernoulli. Than

$$\|\sum_i \sigma_i A_i\|_p \leq \sqrt{p} \max\left\{\|(\sum_i A_i A_i^T)^{\frac{1}{2}}\|_p, \|(\sum A_i^T A_i)^{\frac{1}{2}}\|_p\right\}$$

To take the square root of a matrix just produce the singular value decomposition $U\Sigma V^T$ and take the square root of each of the singular values.

Now continuing we want

$$P(\|(\Pi U)^T(\Pi U) - I\| > \epsilon) < \delta$$

. We know the above expression is

$$P(\|(\Pi U)^T(\Pi U) - I\| > \epsilon) < \frac{1}{\epsilon^p} E\|(\Pi U)^T(\Pi U) - I\|^p \leq \frac{C^p}{\epsilon^p} E\|(\Pi U)^T(\Pi U) - I\|_{S_p}^p$$

We want to bound $\|(\Pi U)^T(\Pi U) - I\|_p$ and we know

$$(\Pi U)^T(\Pi U) = \sum_i z_i z_i^T$$

where z_i is the i'th row of ΠU This all implies

$$\|(\Pi U)^T(\Pi U) - I\|_p = \|\sum_i z_i z_i^T - E\sum y_i y_i^T\|_p$$

where $y_i \sim z_i$. Now we do the usual trick with proving bernstein. By convexity we interchange the expectation with the norm and obtain

$$\leq \|\sum_i (z_i z_i^T - y_i y_i^T)\|_p$$

which is just the usual symmetrization trick assuming row of Π are independent. Then we simplify

$$\leq 2\|\sum_i \sigma_i z_i z_i^T\|_{L^p(\sigma,z)} \leq \sqrt{p}\|(\sum_i \|z_i\|_2^2 z_i z_i^T)^{\frac{1}{2}}\|_p$$

This approach of using matrix concentration inequalities has beene used by

The following was observed by Cohen, noncommutative khintchine can be applied to sparse JL

$$m \geq \frac{dpolylog(\frac{1}{\delta})}{\epsilon^2}, s \geq \frac{polylog(\frac{d}{\delta})}{\epsilon^2}$$

but Cohen is able to obtain $m \geq \frac{d\log(\frac{d}{\delta})}{\epsilon^2}, s \geq \frac{\log(\frac{d}{\delta})}{\epsilon}$ for s containing dependent entries as opposed to independent entries. There is a conjecture that the multiplies in $d\log(\frac{d}{\delta})$ is actually an addition. This will have significance in compressed sensing.

2.3 Moment Chernoff

Consider the following combinatorial argument

$$P(\|(\Pi U)^T(\Pi U) - I\| > \epsilon) < \frac{1}{\epsilon^p} E\|(\Pi U)^T(\Pi U) - I\|^p \leq \frac{1}{\epsilon^p} Etr((\Pi U)^T(\Pi U) - I)$$

We know that the trace of an exponentiated matrix is

$$E(tr(B^p)) = \sum_{i_1, i_2, .. i_{p+1}} \prod_{t=1}^{p} B_{i_t i_{t+1}}$$

The rest is just combinatorics.

2.4 AMM$_F$

For the main result of this section see [6] The basic observation by Nguyen is that

$$\|(\Pi U)^T(\Pi U) - I\| < \|(\Pi U)^T(\Pi U) - I\|_F$$

so what we want is

$$P_\Pi(\|(\Pi U)^T(\Pi U) - I\| > \epsilon) < \delta$$

We know that $U^T U = I$ so this is exactly the form of matrix multiplication discussed two lectures before. So rewriting we obtain

$$P_\Pi(\|(\Pi U)^T(\Pi U) - U^T U\| > \epsilon' \|U\|_F^2) < \delta$$

Where the Frobenius norm of U is d because it's composed of d orthonormal vectors. So we may set $\epsilon = \frac{\epsilon}{d}$ and we need $O(\frac{1}{\epsilon'^2 \delta}) = O(\frac{d}{\epsilon^2 \delta})$ rows.

2.5 Chaining

The basic idea in chaining is to do a more clever net argument than previously discussed. See for example Section 3.2.1 of the Lecture 12 notes on methods of bounding the gaussian width $g(T)$. *Chaining* is the method by which, rather than using one single net for T, one uses a sequence of nets (as in Dudley's inequality, or the generic chaining methodology to obtain the γ_2 bound discussed there).

See [3] by Clarkson and Woodruff for an example of analyzing the SJLT using a chaining approach. They showed it suffices to have $m \geq \frac{d^2 \log^{O(1)}(\frac{d}{\epsilon})}{\epsilon^2}, s = 1$. As we saw above, in later works it was shown that the logarthmic factors are not needed (e.g. by using the moment method, or the AMM_F approach). It would be an interesting exercise though to determine whether the [3] chaining approach is capable of obtaining the correct answer without the extra logarithmic factors.

Note: $s = 1$ means we can compute ΠA in time equal to the number of nonzero entries of A .

3 Other ways to use subspace embeddings

3.1 Iterative algorithms

This idea is due to Tygert and Rokhlin [7] and Avron et al. [2]. The idea is to use gradient descent. The performance of the latter depends on the *condition number* of the matrix:

Definition 3. *For a matrix A, the* ***condition number*** *of A is the ratio of its largest and smallest singular values.*

Let Π be a 1/4 subspace embedding for the column span of A. Then let $\Pi A = U\Sigma V^T$ (SVD of ΠA). Let $R = V\Sigma^{-1}$. Then by orthonormality of U

$$\forall x : \|x\| = \|\Pi A R x\| = (1 \pm 1/4)\|ARx\|$$

which means $AR = \tilde{A}$ has a good condition number. Then our algorithm is the following

1. Pick $x^{(0)}$ such that

   $$\|\tilde{A}x^{(0)} - b\| \leq 1.1\|\tilde{A}x^* - b\|$$

   (which we can get using the previously stated reduction to subspace embeddings with ε being constant).

2. Iteratively let $x^{(i+1)} = x^{(i)} + \tilde{A}^T(b - \tilde{A}x^{(i)})$ until some $x^{(n)}$ is obtained.

We will give an analysis following that in [3] (though analysis of gradient descent can be found in many standard textbooks). Observe that

$$\tilde{A}(x^{(i+1)} - x^*) = \tilde{A}(x^{(i)} + \tilde{A}^T(b - \tilde{A}x^{(i)}) - x^*) = (\tilde{A} - \tilde{A}\tilde{A}^T\tilde{A})(x^{(i)} - x^*),$$

where the last equality follows by expanding the RHS. Indeed, all terms vanish except for $\tilde{A}\tilde{A}^T b$ vs $\tilde{A}\tilde{A}^T\tilde{A}x^*$, which are equal because x^* is the optimal vector, which means that x^* is the projection of b onto the column span of $\tilde{A}$.

Now let $AR = U'\Sigma'V'^T$ in SVD, then

$$\begin{aligned}\|\tilde{A}(x^{(i+1)-x^*})\| &= \|(\tilde{A} - \tilde{A}\tilde{A}^T\tilde{A})(x^{(i)} - x^*)\| \\ &= \|U'(\Sigma' - \Sigma'^3)V'^T(x^{(i)} - x^*)\| \\ &= \|(I - \Sigma'^2)U'\Sigma'V'^T(x^{(i)} - x^*)\| \\ &\le \|I - \Sigma'^2\| \cdot \|U'\Sigma'V'^T(x^{(i)} - x^*)\| \\ &= \|I - \Sigma'^2\| \cdot \|\tilde{A}(x^{(i)} - x^*\| \\ &\le \frac{1}{2} \cdot \|\tilde{A}(x^{(i)} - x^*)\|\end{aligned}$$

by the fact that $\tilde{A}$ has a good condition number. So, $O(\log 1/\varepsilon)$ iterations suffice to bring down the error to ε. In every iteration, we have to multiply by AR; multiplying by A can be done in time proportional to the number of nonzero entries of A, $\|A\|_0$, and multiplication by R in time proportional to d^2. So the dominant term in the time complexity is $\|A\|_0 \log(1/\varepsilon)$, plus the time to find the SVD.

3.2 Sarlos' Approach

This approach is due to Sarlós [8]. First, a bunch of notation: let

$$\begin{aligned}x^* &= \operatorname{argmin}\|Ax - b\| \\ \tilde{x}^* &= \operatorname{argmin}\|\Pi Ax - \Pi b\|. \\ A &= U\Sigma V^T \text{ in SVD} \\ Ax^* &= U\alpha \text{ for } \alpha \in \mathbb{R}^d \\ Ax^* - b &= -w \\ A\tilde{x}^* - Ax^* &= U\beta\end{aligned}$$

Then, $OPT = \|w\| = \|Ax^* - b\|$. We have

$$\begin{aligned}\|A\tilde{x}^* - b\|^2 &= \|A\tilde{x}^* - Ax^* + Ax^* - b\|^2 \\ &= \|A\tilde{x}^* - Ax^*\|^2 + \|Ax^* - b\|^2 \text{ (they are orthogonal)} \\ &= \|A\tilde{x}^* - Ax^*\|^2 + OPT^2 = OPT^2 + \|\beta\|^2\end{aligned}$$

We want $\|\beta\|^2 \le 2\varepsilon OPT^2$. Since $\Pi A, \Pi U$ have same column span,

$$\begin{aligned}\Pi U(\alpha + \beta) = \Pi A\tilde{x}^* &= \operatorname{Proj}_{\Pi A}(\Pi b) = \operatorname{Proj}_{\Pi U}(\Pi b) \\ &= \operatorname{Proj}_{\Pi U}(\Pi(U\alpha + w)) = \Pi U\alpha + \operatorname{Proj}_{\Pi U}(\Pi w)\end{aligned}$$

so $\Pi U\beta = \operatorname{Proj}_{\Pi U}(\Pi w)$, so $(\Pi U)^T(\Pi U)\beta = (\Pi U)^T\Pi w$. Now, let Π be a $(1 - 1/\sqrt[4]{2})$-subspace embedding – then ΠU has smallest singular value at least $1/\sqrt[4]{2}$. Therefore

$$\|\beta\|^2/2 \le \|(\Pi U)^T(\Pi U)\beta\|^2 = \|(\Pi U)^T\Pi w\|^2$$

Now suppose Π also approximately preserves matrix multiplication. Notice that w is orthogonal to the columns of A, so $U^T w = 0$. Then, by the general approximate matrix multiplication property,

$$\mathbb{P}_{\Pi}\left(\|(\Pi U)^T\Pi w - U^T w\|_2^2 > \varepsilon'^2\|U\|_F^2\|w\|_2^2\right) < \delta$$

We have $\|U\|_F = \sqrt{d}$, so set error parameter $\varepsilon' = \sqrt{\varepsilon/d}$ to get

$$\mathbb{P}\left(\|(\Pi U)^T \Pi w\|^2 > \varepsilon \|w\|^2\right) < \delta$$

so $\|\beta\|^2 \le 2\varepsilon\|w\|^2 = 2\varepsilon OPT^2$, as we wanted.

So in conclusion, we don't need Π to be an ε-subspace embedding. Rather, it suffices to simply be a c-subspace embedding for some fixed constant $c = 1 - 1/\sqrt{2}$, while also providing approximate matrix multiplication with error $\sqrt{\varepsilon/d}$. Thus for example using the Thorup-Zhang sketch, using this reduction we only need $m = O(d^2 + d/\varepsilon)$ and still $s = 1$, as opposed to the first reduction in these lecture notes which needed $m = \Omega(d^2/\varepsilon^2)$.

References

[1] Noga Alon, Yossi Matias, Mario Szegedy. The Space Complexity of Approximating the Frequency Moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999.

[2] Haim Avron and Petar Maymounkov and Sivan Toledo. Blendenpik: Supercharging LAPACK's least-squares solver *SIAM Journal on Scientific Computing*, 32(3) 1217–1236, 2010.

[3] Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time *Proceedings of the 45th Annual ACM Symposium on the Theory of Computing (STOC)*, 81–90, 2013.

[4] James Demmel, Ioana Dumitriu, and Olga Holtz. Fast linear algebra is stable. *Numer. Math.*, 108(1):59—91, 2007.

[5] Xiangrui Meng and Michael W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression *Proceedings of the 45th Annual ACM Symposium on the Theory of Computing (STOC)*, 91–100, 2013.

[6] Jelani Nelson and Huy L. Nguyễn. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2013.

[7] Vladimir Rokhlin and Mark Tygert. A fast randomized algorithm for overdetermined linear least-squares regression. *Proceedings of the National Academy of Sciences*, 105 (36) 13212–13217, 2008.

[8] Tamas Sarlós. Improved approximation algorithms for large matrices via random projections. *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 143–152, 2006.

[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.