MIT 6.854/18.415: Advanced Algorithms Spring 2016

Lecture 24 – May 4, 2016

Lecturer: Michael Cohen (for Ankur Moitra)

Scribes: Shalom Abate, Arthur Delarue, Giancarlo Sturla, and Alexander Wallar

1 Subspace Embedding

Definition 1. *An embedding for a set $S \subseteq \mathbb{R}^n$ with distortion ϵ is an $m \times n$ matrix Π such that $\forall x \in S$, $(1-\epsilon)||x||^2 \leq ||\Pi x||^2 \leq (1+\epsilon)||x||^2$*

Definition 2. *A subspace embedding is an embedding for a set S, where S is a k-dimensional linear subspace*

Claim 3. *For any k-dimensional linear subspace S, there is an embedding for S with $m = k$ and $\epsilon = 0$. Just set $Pi = U^T$, where U is a matrix whose columns form an orthonormal basis for S.*

Proof of claim. For any $x \in S$, we can write $x = Ux'$, where $||x'|| = ||x||$. Then $U^T x = (U^T U)x' = x'$. □

Disadvantage for algorithms:

- Often finding an orthonormal basis and/or multiplying by it will end up being as slow as the problems we are trying to solve in the first place.
- For some applications we actually don't know the subspace we're trying to embed!

2 Oblivious subspace embeddings

Definition 4. *A (k, ϵ, δ)-Oblivious Subspace Embedding (OSE) is a random matrix Π such that, for any fixed k-dimensional subspace S, with probability at least $1-\delta$, Π is a subspace embedding for S with distortion ϵ.*

The reader will note that for $k = 1$, this simply reduces to the randomized Johnson-Lindenstrauss (JL) property.

Claim 5. *Any matrix satisfying the randomized JL property for $\epsilon' = \epsilon/4$, $\delta' = \frac{\delta}{2^{100k}}$ is a (k, ϵ, δ)-OSE.*

To prove the claim, we first give the following lemma, which simply says it is enough to consider a finite subset of the subspace.

Lemma 6 (from random matrix lecture)**.** *There exists a set T of size 2^{100k} unit vectors in $\mathbb{R}^k$ such that for any symmetric matrix A,*

$$\|A\|_{op} \leq 4 \left| \max_{x \in T} x^T A x \right| .$$

Proof of claim. For any subspace S with orthonormal basis U, for any $x \in S$, we can write $x = Ux'$, with x' a k-dimensional vector. Now write down the matrix $M = U^T \Pi^T \Pi U - I$.

Then we know that for every $x' \in T$, $x'^T M x' = \|\Pi U x'\|_2^2 - \|x'\|_2^2 = \|\Pi U x'\|_2^2 - \|U x'\|_2^2$. The maximum distortion of Π is equal to $\|M\|_{op}$, while for any vector x' in $\mathbb{R}^k$, $\left|x'^T M x'\right|$ is the distortion of that particular vector. Then we use a union bound over all of T to show that all x' in T are simultaneously preserved up to distortion ϵ' with probability at least $1 - \delta$. Using lemma 6, we get $\|M\|_{op} \leq 4\epsilon' = \epsilon$. $\square$

3 Beyond JL embeddings

Since we have JL embeddings (e.g. dense Gaussian matrices) of size $O(\varepsilon^{-2} \log \frac{1}{\delta})$, this implies that there are (k, ε, δ)-OSEs of size $O(\varepsilon^{-2}(k + \log(\frac{1}{\delta})))$, or $O(k)$ if we ignore ε and δ. In other words, we only have to pay a constant factor for obliviousness.

The disadvantage for dense Gaussian matrices in particular is that, again, often multiplying by the embedding matrix will be comparably slow to the original problem we're solving. One approach is to use faster matrices that satisfy the randomized JL property, like the aptly-named Fast JL matrices [1]. However, there are other fast matrices that are more specific to subspace embedding - in particular, sparse matrices.

One approach is random fully sparse matrices first presented by Clarkson and Woodruff [2]. We cosntruct a random $m \times n$ matrix Π as follows. In each column, pick an entry independently at random and randomly set it to ± 1 (this is called the Rademacher distribution), and set all other entries to 0. By construction, this matrix has only one nonzero entry per column, so any vector (or matrix) can be multiplied by it in linear time in the number of nonzero entries of the input. This gives a (k, ε, δ)-OSE for $m = O(\varepsilon^{-2} \frac{k^2}{\delta})$.

We can see that this loses in two places compared to the JL matrices. First, the number of dimensions needed is k^2 rather than k, and the dependence on the failure probability goes as $\frac{1}{\delta}$ rather than $\log \frac{1}{\delta}$ so for example we can't afford to ask for a high probability of success.

An improved and simplified analysis of this was given independently by Nelson and Nguyen [5] and by Mahoney and Meng [4]. This analysis involves looking at the quantity $\mathbb{E}\left[\operatorname{Tr}((U^T \Pi^T \Pi U - I)^2)\right]$.

Another approach called OSNAP which was also introduced by Nelson and Nguyen [5] is essentially a compromise between the two. It uses matrices that are sparse, but not fully sparse. In this scenario, we create instead s nonzero entries per column in the following way. For each column, pick s entries indepndently at random and randomly set them to $\pm \frac{1}{\sqrt{s}}$ and set all other entries to 0. Note that this definition reproduces usual dense JL matrices for $s = m$, and the fully sparse case for $s = 1$. More generally, there is a tradeoff between m and s. One decent setting that can be proven, getting most of the advantages of both cases, is $m = O(\varepsilon^{-2} k \log \frac{k}{\delta})$ and $s = \frac{1}{\varepsilon} \log(\frac{k}{\delta})$.

Nelson thinks that it may be possible to reduce m to $O(\varepsilon^{-2}(k + \log \frac{1}{\delta})$ for the same value of s; i.e. losing only a constant factor as compared to dense JL matrices.

An improved analysis of this approach was given by Cohen [3].

4 Applications: Least Squares Regression

We can directly apply Oblivious Subspace Embeddings to the Least Squared Regression problem. Let's first define this problem:

We want to find a vector $x \in \mathbb{R}^k$ such that

$$x = \arg\min ||Ax - b||_2^2$$

Where $A \in \mathbb{R}^{n \times k}$. For this application of Least Squared Regression, we are interested in cases where $n \gg k$.

Recall that we can compute an exact solution $x^* = (A^T A)^{-1}(A^t b)$. However, this approach requires use to multiple a $k \times n$ matrix with a $n \times k$ matrix which takes $O(nk^2)$ time (Note: Fast Matrix Multiplication can help reduce the factor of k, but only for some small power and is not significant). Our goal is to remove the multiplicative n term from the run time complexity.

Using Oblivous Subspace Embeddings, we can use the "sketch and solve" approach to improve our runtime. First, we replace our objective to finding a vector $\hat{x}$ such that

$$\hat{x} = \arg\min ||\Pi(Ax - b)||_2^2 = \arg\min ||(\Pi A)x - (\Pi b)||_2^2$$

Note that if we choose the matrix Π intelligently, then we are essentially solving anothing Least Squares Regression problem with a much smaller n.

Furthermore, if Π is a subspace embedding with distortion ϵ for the $(k+1)$-dimensional subspace spanned by the columns of A and b, then

$$(1 - \epsilon)||Ax - b||_2^2 \leq ||A\hat{x} - b||_2^2 \leq (1 + \epsilon)||Ax - b||_2^2$$

This results in an ϵ-approximate solution to our original objective. The advantage is that the problem size is now about $O(\frac{k}{\epsilon^2})$ which greatly improves our runtime.

Side Note: We can improve the ϵ-dependence on the runtime by using an alternative analysis using both Oblivious Subspace Embeddings and a technique known as Approximate Matrix Multiplication.

References

[1] Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pages 557–563, New York, NY, USA, 2006. ACM.

[2] Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time, 2012.

[3] Michael B. Cohen. *Nearly Tight Oblivious Subspace Embeddings by Trace Inequalities*, chapter 21, pages 278–287. 2016.

[4] Xiangrui Meng and Michael W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression, 2012.

[5] Jelani Nelson and Huy L. Nguyen. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings, 2012.