COMS W6998-5: Algorithms through Geometric Lens (Fall'17) Sep 25, 2017

Lecture 6 – Dimension Reduction

Instructors: *Alex Andoni, Ilya Razenshteyn* Scribes: *Kiran Vodrahalli*

1 Introduction

Let's continue where we stopped last time. We were looking at dimension reduction, with the setting $X \subset \mathbb{R}^d$, with $|X| = n$ points. We would like to learn a map $f : X \to \mathbb{R}^{d'}$ with $d' << d$. And we would furthermore like that for all x_1, x_2 we have $\|f(x_1) - f(x_2)\|_2 \leq (1 \pm \epsilon)\|x_1 - x_2\|_2$.

We also formulated last time the **Johnson-Lindenstrauss** theorem:

Theorem 1. *Johnson-Lindenstrauss.*
If we have n points and care about preserving distances up to factor ϵ, we can definitely perserve distances if $d' = \Omega\left(\frac{\log n}{\epsilon^2}\right)$.

Last time, we proved the following theorem:

Theorem 2. *Dimension reduction on the simplex.*
Consider the sphere $S^{d'-1}$. Then $\exists x_1, \cdots, x_n \in S^{d'-1}$ such that all pairwise dot products are small:

$$|\langle x_i, x_j \rangle| \leq \epsilon$$

for $n = 2^{\Omega(\epsilon^2 d')}$. For points on the unit sphere, $\|x_i - x_j\|_2^2 = \|x_i\|_2^2 - 2\langle x_i, x_j \rangle + \|x_j\|_2^2 = 2(1 - \langle x_i, x_j \rangle)$. Thus, this lemma implies for dimension d', there exist n points on the sphere s.t.

$$2(1 - \epsilon) \leq \|x_i - x_j\|_2^2 \leq 2(1 + \epsilon)$$

Then, for the vertices of the simplex Δ^d, which has $d + 1$ points in $\mathbb{R}^d$ s.t. $\|x_i - x_j\|_2 = 1$, there exists a mapping $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ onto the sphere $S^{d'-1}$ for d' s.t. $2^{\Omega(\epsilon^2 d')} \geq d + 1$, which by the lemma preserves distances, thus giving that dimension

$$d' = \Omega(\frac{\log d + 1}{\epsilon^2}) = \Omega(\frac{\log |\Delta_d|}{\epsilon^2})$$

suffices for dimension reduction.

The proof of this theorem essentially follows by using the probabilistic method to find points on the sphere which are approximately orthogonal by using the rotational symmetry of Gaussians, followed by a union bound to bound the probability of failure.

Now we want to do things in full generality. We will still use the probabilistic method. Let me describe the method (which gave rise to the method of random projections).

2 Random Dimension Reduction

We want to find a dimension reduction map $f : \mathbb{R}^d \to \mathbb{R}^{d'}$. Our map will be a linear map. We are taking a matrix $A \in \mathbb{R}^{d' \times d}$ with d' rows and d columns. Our dimension reduction for vector $x \in \mathbb{R}^d$ will be $Ax \in \mathbb{R}^{d'}$. Now it's a very simple construction: Matrix A is completely independent on the specific set of points that we care about. A specific way to construct A is to take entries of A to be i.i.d. Gaussians.

Theorem 3. *Gaussians perserve norm. (Oblivious dimension reduction)*
For all $x \in \mathbb{R}^d$, $\forall \epsilon, \delta > 0$,

$$\mathbb{P}\left\{\|Ax\|_2^2 \in d'(1 \pm \epsilon)\|x\|_2^2\right\} \geq 1 - \delta$$

where $d' = \Omega\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$.

This theorem implies Johnson-Lindenstrauss. First, $\|Ax - Ay\| = \|A(x - y)\|$ and then apply the theorem to get a bound on the ℓ_2 distance between points. Now consider our point set X, which is n points. If we ask for the above property to hold over the n^2 distances, we can apply union bound to upper bound

$$\mathbb{P}\left\{\exists x_1, x_2 : \|Ax_1 - Ax_2\|_2^2 \notin d'(1 \pm \epsilon)\|x_1 - x_2\|_2^2\right\} \leq \delta n^2 \leq \frac{1}{10}$$

and thus we see $\delta = \frac{1}{10n^2}$, and we get $d' = \mathcal{O}\left(\frac{\log n}{\epsilon^2}\right)$. To get rid of the d' in front, we will need to scale the Gaussians by $1/\sqrt{d'}$. Thus we get the reduction from this second theorem to Johnson-Lindenstrauss.

Observation 4. *First we see that A is completely independent from the set of points we care about. We may need to apply dimension reduction to a set of points we don't know in advance. This random projection technique is pretty useful, but we still have a dependence on $1/\epsilon^2$ which is painful when we need very high accuracy.*

Observation 5. *When you did ℓ_2 sketching, you had Tug-of-Wars++ sketch. In that case, you had Rademacher variables. The Tug-of-War had much worse dependence on δ; but here, we need very good dependence on δ. It's much easier to prove good dependence for Gaussian entries. But actually, a similar bound holds for random ± 1 (Rademacher) matrix, but this theorem is more complicated. Using Gaussians is more natural.*

2.1 Proof of Oblivious Dimension Reduction

Proof. Recall we have Ax as our reduced dimension vector. Let's consider the first entry of Ax, $(Ax)_1$. We have

$$(Ax)_1 = \sum_i g_i x_i \sim \mathcal{N}(0,1)\|x\|_2$$

This is a fundamental property of Gaussians called 2-stability.

Definition 6. 2-*Stability of Gaussians.*
Suppose you have $g_1, \cdots, g_d$ i.i.d. $\mathcal{N}(0,1)$, then

$$\sum_{i=1}^{d} g_i x_i \sim \mathcal{N}(0,1)\|x\|_2$$

This will be true for every coordinate of Ax, and moreover they will be independent. Thus, Ax has i.i.d. entries distributed as $\mathcal{N}(0,1)\|x\|_2$. This doesn't hold for Rademacher random variables (random ± 1, a weaker property holds).

Let's prove 2-stability: This is a relatively simple corollary of the spherical symmetry of Gaussians. We have that $f_g(u) \sim e^{-\|u\|_2^2/2}$, where f_g is the Gaussian density. Note that $\sum_{i=1}^d g_i x_i = \langle g, x\rangle = \langle Ug, Ux\rangle$ where U is orthogonal ($U^T U = I$). This is true for any dot product. Now we can choose U to be very special. By rotating x, we can make it proportional to the first basis vector. Choose U such that it satisfies $Ux = e_1\|x\|_2$. Then, we get $\langle Ug, Ux\rangle = \langle \tilde{g}, \|x\|_2 e_1\rangle$ invoking spherical symmetry of Gaussians. This is just $\tilde{g}_1\|x\|_2 \sim \|x\|_2\mathcal{N}(0,1)$, as desired.

This lemma can also be proved directly by observing the following two properties of variance for Gaussian variables: $a\mathcal{N}(0,1) \sim \mathcal{N}(0,a^2)$, and $\mathcal{N}(0,a^2) + \mathcal{N}(0,b^2) = \mathcal{N}(0,a^2+b^2)$.

Now we have used 2-stability to show that $(Ax)_i \sim \mathcal{N}(0,1)\|x\|_2$. Now let's understand $\|Ax\|_2^2$, why is it concentrated around what we expect? We have, where $g_i \sim N(0,1)$ i.i.d.,

$$\begin{aligned} \|Ax\|_2^2 &= \sum_{i=1}^d g_i^2 \|x\|_2^2 \\ &= \chi^2(d')\|x\|_2^2 \end{aligned} \tag{1}$$

where $\chi^2(d')$ is the chi-squared distribution with d' degrees of freedom. Now we will give a lemma:

Lemma 7. *Concentration of χ^2.*
Suppose $Z \sim \chi^2(s)$. Then $\mathbb{E}\left[\chi^2(s)\right] = s$ by linearity of expectation. Now $Var(Z) = C * s$. That is,

$$\mathbb{P}\left\{|X - s| \geq \epsilon * s\right\} \leq e^{-\Theta(\epsilon^2 s)}$$

Proof. This can be thought of as a kind of instantiation of the Central Limit Theorem (CLT). I'll provide a reference to the full proof of this lemma after the class. □

Recall that $\|Ax\|_2^2 \sim \chi^2(d')\|x\|_2^2$. If $e^{-\Theta(\epsilon^2 d')} < d$, then $d' = \Omega\left(\frac{\log 1/\delta}{\epsilon^2}\right)$ so that $\|Ax\|_2^2 \in (1\pm\epsilon)d'\|x\|_2^2$. Thus we are done.

□

3 Fast Dimension Reduction

So oblivious dimension reduction was a short and self-contained result, as long as you believe the χ^2 lemma (which you should based on CLT), and it's nice.

However, you need to multiply Ax: This takes time around $\mathcal{O}(d' \times d)$. Is that a lot? Let's write some concrete numbers. Say we had a vector which had 10^6 dimensions. So you take some Gaussian matrix, and you'd like to do dimension reduction into 1000 dimensions. Then you take your Gaussian matrix which would have size 1000×10^6, this is roughly 10^9 iterations. How fast can you actually do this? If you use BLAS library, it would take 0.2 seconds or so. So this is relatively fast. But if you have to do this multiplication many times, this would become a bottleneck: 0.2 seconds builds up. But we can do something faster.

Theorem 8. *Fast Johnson-Lindenstrauss Transform (Ailon & Chazelle '04)*
For every $\epsilon, \delta > 0$, there exists a distribution over matrices $A \in \mathbb{R}^{d' \times d}$ s.t. $\forall x$ we have

$$\mathbb{P}\left\{\|Ax\|_2^2 \in d'(1 \pm \epsilon)\|x\|_2^2\right\} \geq 1 - \delta$$

and also $x \to Ax$ happens in time $\mathcal{O}(d \log d)$. But there is a drawback: Here, $d' = \Omega\left(\frac{\log(1/\delta)\log(d/\delta)}{\epsilon^2}\right)$. We pay an extra factor of $\log(d/\delta)$.

So $\mathcal{O}(d \log d)$ is very fast. Instead of a billion operations in our previous example, we only spend 20 million operations. Now we can do things in a millisecond instead of 0.2 seconds.

In this theorem, there is no better analysis of the method which we will present: It's optimal. There was a whole bunch of follow-up work however involving further tradeoffs.

The simplest thing you could consider doing for dimension reduction is to subsample the vector x.

3.1 Subsampling

We consider the following process: Define $(Ax)_i = x_{j(i)}$, where $j(i) \in [d]$ is chosen uniformly at random. So what does A look like? Essentially, there is one 1 per row, at uniformly random entries. Essentially, it's a selection matrix. Thus subsampling is a linear transformation. Well, does it work? It doesn't work sometimes. When doesn't it work? Well, if you have a sparse vector. I will now prove that this is the only case where this won't work. Informally, I'd like to claim that subsampling works if the vectors are spread sort of uniformly (i.e. ℓ_2 mass is not concentrated on a subset).

Let's try to analyze this estimator from first principles. We need to analyze expectation and variance.

Let's calculate $\mathbb{E}\left[\|Ax\|_2^2\right]$. Let's again think in terms of linearity of expectation. What's the expected square of a random entry of x? With probability $1/d$, the first coordinate will be x_i^2, for all $i \in [d]$. Then since there are d' entries, we need to add d' times. Thus, we get the expectation is $\frac{d'}{d}\|x\|_2^2$.

Then, $\mathrm{Var}(\|Ax\|_2^2) = \sum_{i=1}^{d} \mathrm{Var}((Ax)_i^2)$. Then, $\mathrm{Var}((Ax)_i^2) \leq \mathbb{E}\left[(Ax)_i^4\right] = \frac{\|x\|_4^4}{d}$. The first inequality follows from the definition of variance, and the second equality follows from the same argument we used to get the expectation, using linearity of expectation.

Now, it's okay to use Chebyshev inequality on $\|Ax\|_2^2$ for constant probability claims: We need to understand

$$\mathbb{P}\left\{\left|\|Ax\|_2^2 - \mathbb{E}\left[\|Ax\|_2^2\right]\right| \geq t\sqrt{\mathrm{Var}[\|Ax\|_2^2]}\right\} \leq \frac{1}{t^2}$$

In order to get $\delta \approx 1/10$, it's enough to set $d' = \Omega\left(\frac{1}{\epsilon^2} \cdot d \cdot \frac{\|x\|_4^4}{\|x\|_2^4}\right)$. Let's understand this quantity, it's not clear how we can bound it. There are two extremes here: First, if our vector is very sparse: everything is in a single entry. Suppose $x = e_1$. Then, $\|x\|_2 = \|x\|_4 = 1$. Thus, we need to sample around d/ϵ^2 entries: This is terrible because this is worse than linear. Second case is suppose our vector is $x = (1, 1, 1, \cdots, 1)$. Here, $\|x\|_2 = \sqrt{d}$ and $\|x\|_4 = d^{1/4}$. Here, $d' = \Omega(1/\epsilon^2)$: This is great! It matches the original dimensional reduction result.

What I will tell you later is how to reduce to the second case where subsampling works. We will not always be able to reduce to the second case where entries are sort of equal.

Another note: Chebyshev only gets good bounds for constant probability, it's too weak to get good dependence for small δ. We will use the Bernstein inequality (I will send reference for this, but morally, everything follows from CLT).

Theorem 9. *Bernstein Inequality.*
Suppose you have $X_1, \cdots, X_n$ i.i.d. random variable with $\mathbb{E}[X_i] = 0$, $\mathrm{Var}(X_i) = 1$, and moreover, $|X_i| \le M$. We look at

$$\mathbb{P}\left\{|\sum X_i| \ge t\right\} = e^{-\Theta(1)\cdot\frac{t^2}{n+Mt}}$$

Let's understand to what extent we want our vector to be well-concentrated for the Ailon-Chazelle bound to hold. Let's try to get rid of the $\|\cdot\|_4$, it's confusing. Let's use ℓ_∞ norm.

Definition 10. $\|\cdot\|_\infty$. $\|x\|_\infty = \max_i |x_i|$.

Claim 11.

$$\frac{\|x\|_4^4}{\|x\|_2^4} \le \frac{\|x\|_\infty^2}{\|x\|_2^2}$$

We need this ℓ_∞ assumption to use Bernstein to claim the vectors are supported on some bounded interval $[-M, M]$. We can get a concrete M for ℓ_∞ norm.

If you apply Bernstein to this setting, you get the following claim (a high-probability version of what we saw before):

Lemma 12. *Let A be a matrix that corresponds to subsampling. Then, with probability*

$$\mathbb{P}\left\{\|Ax\|_2^2 \in (1\pm\epsilon)\frac{d'}{d}\|x\|_2^2\right\} \ge 1-\delta$$

dimension $d' = \Omega\left(\frac{\log 1/\delta}{\epsilon^2}\frac{d\|x\|_\infty^2}{\|x\|_2^2}\right)$ suffices.

The intuition for this claim is that if we want to understand the average of a vector, subsampling works if all the mass is not only on a subset of it. So we want to understand how much a vector needs to be spread to get this bound.

Let us show now how to reduce the general case to the case where a vector has a small ℓ_∞-ℓ_2 ratio (which is desirable). Let's start with one naive approach that doesn't work (Ailon-Chazelle gives an approach that DOES work).

Claim 13. *(Naive idea). Let's apply random orthogonal matrix to our vectors. Take $U \in \mathbb{R}^{d\times d}$ to be a random orthgonal matrix. How do we get this? Take some Gaussian matrix and apply Gram-Schmidt orthogonalization. So U has nice properties: Ux is a uniformly random vector on the sphere for fixed x, and norms are preserved. So basically Ux is random coordinate on the sphere. Then $(Ux)_1 \sim \frac{1}{\sqrt{d}}\mathcal{N}(0,1)$. Thus,*

$$\mathbb{P}\left\{|(Ux)_1| \le \sqrt{\frac{\log d/\delta}{d}}\right\} \ge 1 - \frac{\delta}{10d}$$

and

$$\mathbb{P}\left\{|(Ux)_1|_\infty \le \sqrt{\frac{\log d/\delta}{d}}\right\} \ge 1 - \frac{\delta}{10d}$$

So we get the desired result, but the problem is U is still a dense $d\times d$ matrix: We are back where we started in terms of computational complexity.

Next time, we will see a much better version of the chosen U: Hadamard matrices, which almost perserve the desired properties while matrix-vector multiplication is much faster.