Note: These lecture notes are closely based on lecture notes by Sanjeev Arora [1] and Matt Weinberg [2].

1 Previously Proven

1.1 Johnson-Lindenstrauss Projection

Let G be a m by d matrix where each entry is a Normal random variable, i.e. $G_{i,j} \sim \mathcal{N}(0,1)$. Let $\Pi = \frac{1}{\sqrt{m}} G$ and let

$$f(x) = \Pi x.$$

So each entry in $f(v)$ equals $v \cdot g$ for some vector g filled with scaled Normal random variables (note that Gaussian and Normal are synonmous). Other (simpler) approaches also work (See Section 2 below).

1.2 Main Theorems

Theorem 1. *(The (ϵ, δ)-JL property) If $m = 9\log(1/\delta)/\epsilon^2$ then, with probability $1-\delta$, for any vector x,*

$$(1-\epsilon)|x|^2 \leq |\Pi x|^2 \leq (1+\epsilon)|x|^2$$

Theorem 2. *Assume we are given n points $v_1, \ldots v_n \in \mathbb{R}^d$ and a fixed $\epsilon > 0$. Let $m = O(\log n/\epsilon^2)$ and set $f = \Pi$, where Π is a m by d matrix of independent $\mathcal{N}(0,1)$ random variables. Then, with probability $1-1/n$, for any i and j, $1 \leq i < j \leq n$:*

$$(1-\epsilon)|v_i - v_j| \leq |\Pi v_i - \Pi v_j| \leq (1+\epsilon)|v_i - v_j|$$

2 A Simpler Johnson-Lidenstrauss Projection

Here is a simpler Johnson-Lindenstrauss Algorithm for projection that also works.

1. $x_1, \ldots x_m \leftarrow$ vectors in $\mathbb{R}^m$ chosen as follows. Each coordinate is chosen independently and randomly from $\left\{\sqrt{\frac{1+\epsilon}{m}}, -\sqrt{\frac{1+\epsilon}{m}}\right\}$
2. $u_i[j] \leftarrow x_i \cdot u_i$ for all $i : 1 \leq i \leq n$ and $j : 1 \leq j \leq m$

In other words, $u_i = (z_i \cdot x_1, \ldots z_i \cdot x_m)$ for $i = 1, \ldots m$. Note that we can think of this as a linear transformation $u = Az$ where A is a matrix with random and independent entries in $\left\{\sqrt{\frac{1+\epsilon}{m}}, -\sqrt{\frac{1+\epsilon}{m}}\right\}$.

2.1 Analysis

We now do a "sketch" of the analysis. The following lemma shows that things work out well in expectation.

Lemma 1. *For any $1 \leq i < j \leq n$, $E(|u_i - u_j|^2) = (1+\epsilon)|z_i - z_j|^2$*

Proof: According to the projection, we have the following for any $1 \le i < j \le n$:

$$|u_i - u_j|^2 = \sum_{k=1}^{m} \left(\sum_{\ell=1}^{n} (z_i[\ell] - z_j[\ell]) x_k[\ell] \right)^2$$

Fix i and j. Let $z = z_i - z_j$ and let $u = u_i - u_j$. Then for any $1 \le k \le m$, we have

$$\begin{aligned} E(|u \cdot x_k|^2) &= E\left(\left(\sum_{\ell=1}^{n} (z[\ell] x_k[\ell] \right)^2 \right) \\ &= \sum_{\ell} \sum_{\ell'} E\left(z[\ell] x_k[\ell] z[\ell'] x_k[\ell'] \right) \\ &= \sum_{\ell=1}^{n} E\left((z[\ell] x_k[\ell])^2 \right) \\ &= \frac{1+\epsilon}{m} |z|^2 \end{aligned}$$

Hence, by linearity of expectation $E(|u|^2)) = (1+\epsilon)|z|^2$. □

The rest of the analysis follows similar to that in Theorem 2. First, one establishes a (harder) tail-bound around this expectation and then does a union bound over all pairs of points. In this way, we can get the same result as Theorem 2.

3 Applications of JL Projection

- Approximate all-pairs distances in $O(n \log n + nd)$ vs $O(n^2 d)$ time
- Approximate distance-based clustering
- Approximate support vector machine (SVM) classification
- Approximate Linear Regression

Note: For some of these Machine Learning type applications, we need it to be the case that distances are approximately preserved across *all* (infinite) vectors in the vector space. Thus, a simple union bound won't work and instead we need to make use of a technique called ϵ-*nets*. We discuss this technique below.

4 Linear Regression and ϵ-Nets

The following is the classic lear-squares regression problem.

Given: A set of n data vectors $a_1, \ldots a_n \in \mathbb{R}^d$, and n response values $y_1, \ldots, y_n \in \mathbb{R}$. Let A be a $n \times d$ matrix with rows $a_1, \ldots, a_n$; let y be a length n vector with entries $y_1, \ldots, y_n$.

Goal: Find $x \in \mathbb{R}^d$ to minimize

$$\sum_{i=1}^{n}(a_i \cdot x - y_i)^2 = |Ax - y|^2$$

Usually, this problem requires $O(nd^2)$ time to solve (for example, by using singular value decomposition). We now show how to speed it up by reducing n using Johnson-Lidenstrauss.[1]

Let Π be chosen from the family of matrices from Theorem 2. To obtain an approximate solution, we solve the "sketched" problem where we find $x \in \mathbb{R}^d$ to minimize:

$$|\Pi Ax - \Pi y|^2.$$

This can be solved in $O(md^2)$ time (once ΠA and Πy are computed - we haven't discussed this but there are JL transforms which are also fast, since they are sparse). We want to prove that a solution to this smaller problem is a good approximation to the big problem. Note that the following lemma is a direct consequence of Theorem 1, applied to the vector $Ax - y$:

Lemma 2. *Let $m = O(\log(1/\delta)/\epsilon^2)$. Then, for any particular vector x, with probability $1 - \delta$,*

$$(1-\epsilon)|Ax - y|^2 \le |\Pi Ax - \Pi y|^2 \le (1+\epsilon)|Ax - y|^2$$

Now if we could show this was true *for all* x, we'd be done. In particular, let x^* be the optimal solution for the original problem, and let $\tilde{x}^*$ be the solution for the sketched problem. Then we've have, with probability $1 - \delta$.

$$|A\tilde{x}^* - y|^2 \le \frac{1}{1-\epsilon}|\Pi A\tilde{x}^* - \Pi y|^2 \le \frac{1}{1-\epsilon}|\Pi Ax^* - \Pi y|^2 \le \frac{1+\epsilon}{1-\epsilon}|Ax^* - y|^2.$$

In the above the first and last inequalities hold via Lemma 2, and the middle inequality holds by noting that $\tilde{x}^*$ minimizes $|\Pi Ax - \Pi y|$ over all vectors x.

If $\epsilon \le .25$, then $\frac{1+\epsilon}{1-\epsilon} \le 1 + 3\epsilon$, so we can get an approximation to the original regression problem. Q: Why do we need a bound for all x above??? The main problem is that $\tilde{x}^*$ depends on the projection π, and so it's not fixed ahead of time. How do we extend Lemma 2 to all x? We can't use union bounds since there are an infinite number of possible vectors x.

5 Beyond Union Bounds

Recall that we have $A \in \mathbb{R}^{n\times d}$ and want to approximately find x to minimize $|Ax - y|^2$, by instead solving the sketched problem $|\Pi Ax - \Pi y|^2$. We want to argue that for all $x \in \mathbb{R}^d$,

$$(1-\epsilon)|Ax - y|^2 \le |\Pi Ax - \Pi y|^2 \le (1+\epsilon)|Ax - y|^2 \tag{1}$$

But proving this requires establishing a JL-bound for an infinity of possible vectors, which clearly can't be shown via union bounds. Instead, we use a different approach.

5.1 Subspace Embeddings

We will prove a more general statement that implies equation 1, and is useful in other applications.

[1] Note that we are reducing n (number of vectors) and not d (dimension). Since we only care about the matrix A, you could think of n as the dimension and d as the number of vectors.

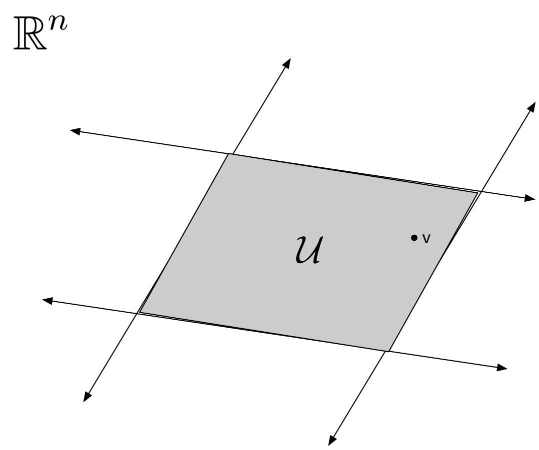

Figure 1. JL approximately preserves distances over any subspace $\mathcal{U}$ of dimension d contained in $\mathbb{R}^n$

Theorem 3. *Let $\mathcal{U} \subset \mathbb{R}^n$ be a d-dimensional linear subspace in $\mathbb{R}^n$. If $\Pi \in \mathbb{R}^{m \times n}$ is chosen from any distribution $\mathcal{D}$ satisfying Theorem 1, then with probability $1 - \delta$,*

$$(1 - \epsilon)|v| \leq |\Pi v| \leq (1 + \epsilon)|v| \tag{2}$$

for all $v \in \mathcal{U}$, as long as $m = O\left(\frac{d \log(1/\epsilon) + \log(1/\delta)}{\epsilon^2}\right)$

(Note that it's possible to prove a slightly tighter bound of $m = O(\frac{d + \log(1/\delta)}{\epsilon^2})$ that we won't discuss here.)

How does this theorem imply equation 1? We can apply it to the $d + 1$ dimensional subspace spanned by the d columns of A and the vector y. Every vector formed by inputting some vector x into the linear equation $Ax - y$ lies in this $d + 1$ dimensional subspace. So for the regression problem, we require dimension $m = O\left(\frac{(d+1)\log(1/\epsilon)}{\epsilon^2}\right)$. In particular, we can approximately solve linear regression over $n >> d$ examples for the same amount of work as $O(d)$ examples, for fixed ϵ.

5.2 An Example

Let $n = 7$ and $\mathcal{U}$ could be the 2 dimensional subspace spanned by $(1, -1, 1, 1, 1, 1, 1)$ and $(1, 1, -1, 1, 1, 1, 1)$. JL will basically find a low-dimensional sub-space that is not much higher than the dimensionality of $\mathcal{U}$.

5.3 Reduction to a Sphere

We first note that Theorem 3 holds so long as equation 2 holds for all points on the unit sphere in $\mathcal{U}$. This is a consequence of linearity of the Euclidean norm. In particular, denote the sphere $\mathcal{S}_{\mathcal{U}}$ as

$$\mathcal{S}_{\mathcal{U}} = \{v \mid v \in \mathcal{U} \text{ and } |v| = 1\}.$$

Now any point $v \in \mathcal{U}$ can be written as cx for some scalar c and some point $x \in \mathcal{S}_{\mathcal{U}}$. If $(1 - \epsilon)|x| \leq |\Pi x| \leq (1 + \epsilon)|x|$, then $c(1 - \epsilon)|x| \leq c|\Pi x| \leq c(1 + \epsilon)|x|$ and so $(1 - \epsilon)|cx| \leq |\Pi cx| \leq (1 + \epsilon)|cx|$.

Note that the last inequality holds since $|cx| = \sqrt{\sum_i (cx)_i^2} = c\sqrt{\sum_i x_i^2} = c|x|$.

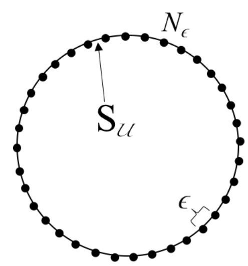

Figure 2. An ϵ-net N_ϵ for a sphere in a 2-dimensional subspace of $\mathcal{U}$

5.4 Constructing a Net

We prove Theorem 3 by showing that there is a large but finite set of points $N_\epsilon \subset \mathcal{S}_{\mathcal{U}}$ such that if equation 2 holds for all $v \in N_\epsilon$, then it holds for all $v \in \mathcal{S}_{\mathcal{U}}$. The set N_ϵ is called an ϵ-net. In particular, we will show the following.

Lemma 3. *For any positive $\epsilon < 1$, there exists a set $N_\epsilon \subset \mathcal{S}_{\mathcal{U}}$ with $|N_\epsilon| = \left(\frac{4}{\epsilon}\right)^d$ such that $\forall v \in \mathcal{S}_{\mathcal{U}}$,*

$$\min_{x \in N_\epsilon} |v - x| \le \epsilon.$$

Proof: We use the following greedy procedure to construct N_ϵ (note that this construction is just for proof of existence, our algorithms do not need to implement this). Initially $N_\epsilon \leftarrow \{\}$. Then:

- While there is a point $v \in \mathcal{S}_{\mathcal{U}}$ with distance greater than ϵ from any point in N_ϵ, add v to N_ϵ.

After running this procedure, we have $|N_\epsilon|$ points such that $\min_{x \in N_\epsilon} |v - x| \le \epsilon$ for all $v \in \mathcal{S}_{\mathcal{U}}$. So we just need to bound $|N_\epsilon|$.

To do so, we first lower bound the volume taken up by balls around points in $N_\epsilon = \{x_1, x_2, \ldots, x_{|N_\epsilon|}\}$. In particular, note that for all $i \ne j$, $|x_i - x_j| \ge \epsilon$. If not, then either x_i or x_j would not have been added to N_ϵ by our greedy algorithm. So if we place balls of radius $\epsilon/2$ around each x_i:

$$B(x_1, \epsilon/2) \ldots B(x_{|N_\epsilon|}, \epsilon/2)$$

then for all $i \ne j$, $B(x_i, \epsilon/2)$ does not intersect $B(x_j, \epsilon/2)$.

So how do we now set up an inequality to bound $|N_\epsilon|$??? The volume of a d dimensional ball of radius r is cr^d for some fixed constant c. Thus, the amount of space taken up by all the balls surrounding points in N_ϵ is $c|N_\epsilon|(\epsilon/2)^d$.

Next note that the amount of space that these balls can exist in is at most the volume of a d dimensional sphere with radius $1 + \epsilon/2$. This volume is $c(1 + \epsilon/2)^d$.

Thus we have that

$$|N_\epsilon| c(\epsilon/2)^d \le c(1 + \epsilon/2)^d$$

Solving for $|N_\epsilon|$, we have that

$$\begin{aligned} |N_\epsilon| &\le \frac{c(1+\epsilon/2)^d}{c(\epsilon/2)^d} \\ &\le \frac{(1+\epsilon/2)^d}{(\epsilon/2)^d} \\ &\le \left(\frac{4}{\epsilon}\right)^d \end{aligned}$$

□

5.5 Proving Theorem 3

We can now prove Theorem 3 by extending to all vectors in the subspace.

Proof: Choose $m = O\left(\frac{\log(|N_\epsilon|/\delta)}{\epsilon^2}\right) = O\left(\frac{d\log(1/\epsilon)+\log(1/\delta}{\epsilon^2}\right)$ so that Equation 2 holds for all $x \in N_\epsilon$ (via Theorem 2 and a union bound).

Now consider any $v \in \mathcal{S}_{\mathcal{U}}$. We claim that for some $x_0, x_1, x_2, \ldots \in N_\epsilon$ that we can write v as:

$$v = x_0 + c_1x_1 + c_2x_2 + \ldots$$

for constants $c_1, c_2, \ldots$ where $|c_i| \leq \epsilon^i$. To see this, note that there is some point x_0 within distance ϵ of v. Next we need to represent $v - x_0$, which has norm at most ϵ. So instead, we can represent the point $\frac{v-x_0}{|v-x_0|}$, which has norm 1 and multiply the resulting coefficients by ϵ. Again there is some point x_1 within distance ϵ of *this* point. Continuing this process ad infinitum gives the claim.

Now, we can consider $|\Pi v|$ and make use of the triangle inequality in order to complete the proof.

$$\begin{aligned}
|\Pi v| &= |\Pi x_0 + \Pi c_1x_1 + \Pi c_2x_2 + \ldots| \\
&\leq |\Pi x_0| + |\Pi c_1x_1| + |\Pi c_2x_2| + \ldots \\
&\leq (1+\epsilon)|x_0| + (1+\epsilon)c_1|x_1| + (1+\epsilon)c_2|x_2| + \ldots \\
&\leq (1+\epsilon)(|x_0| + c_1|x_1| + c_2|x_2| + \ldots) \\
&\leq (1+\epsilon)(1+\epsilon+\epsilon^2+\ldots) \\
&= 1 + O(\epsilon)
\end{aligned}$$

[**Jared:** *EXERCISE: Show that the above is $1+O(\epsilon)$. Some hints are below.*] In the above, the third step follows by the triangle inequality. The fourth step follows by the fact that each $x_i \in N_\epsilon$, and so by a Union bound their norms are all approximately preserved. The last line follows since $\epsilon + \epsilon^2 + \epsilon^3 + \ldots$ is a geometric summation, which has value that is a constant times its largest term.

The other direction of the proof is symmetric. It is included below for completeness.

$$\begin{aligned}
|\Pi v| &= |\Pi(x_0 + c_1x_1 + c_2x_2 + \ldots)| \\
&\geq |\Pi x_0| - \epsilon|\Pi x_1| - \epsilon^2|\Pi x_2| - \ldots \\
&\geq (1-\epsilon) - \epsilon(1+\epsilon) - \epsilon^2(1+\epsilon) + \ldots \\
&= 1 - O(\epsilon)
\end{aligned}$$

□

5.6 Other Applications of JL

Note Winnow and Boosting are ML algorithms we'll discuss soon.

Speed up Winnow by projecting "training data"??? Yes

Speed up Boosting by projecting "training data"??? Yes

Speed up Winnow by projecting attributes??? Not necessarily

Approximate solutions to System of Linear equations? Sometimes

Finding an ϵ-approximate convex hull?? Sometimes

References

[1] Sanjeev Arora. Advanced Algorithm Design Class, Princeton University, 2013. https://www.cs.princeton.edu/courses/archive/fall15/cos521/.

[2] Matt Weinberg. Dimensionality Reduction and the Johnson-Lindenstrauss Lemma, 2019. https://www.cs.princeton.edu/~smattw/Teaching/Fa19Lectures/lec9/lec9.pdf.