Today we'll talk about one of two general techniques we use for dimension reduction: random projections. Somewhat surprisingly, this will also be related to a central problem in streaming algorithms, that of computing the second moment F_2.

1 Dimension Reduction to Preserve Distances

Suppose we are given a set of n points $\{x_1, x_2, \ldots, x_n\}$ in $\mathbb{R}^D$. We want to reduce the dimension these points live in, and still maintain the Euclidean distances between the points? Say we want the distances to be maintained *exactly* for now. By basic linear algebra, any set of n points lies on a $(n-1)$-dimensional subspace (just like any two points define a line, three points a plane); we can always ensure that the dimension is no more than $n-1$. And this is (existentially) tight: e.g., the case when $x_2 - x_1, x_3 - x_1, \ldots, x_n - x_1$ are all orthogonal vectors.

But what if we were OK with the distances being *approximately preserved*? What can we do?

1.1 A Set with Unit Distances

Let's start with a simple case. Suppose you had n points in Euclidean space, all at distance 1 from each other. Could you map them to a smaller dimensional space and maintain their distances approximately.

I claim you've already solved this simple case. In HW#4, you saw that while there could only be D orthogonal unit vectors in $\mathbb{R}^D$, there could be as many as $\exp(c\varepsilon^2 D)$ unit vectors (for some constant $c > 0$) which are ε-orthogonal—i.e., whose mutual inner products all lie in $[-\varepsilon, \varepsilon]$. Near-orthogonality allows us to pack exponentially more vectors!

But observe that:

$$\|\vec{a} - \vec{b}\|_2^2 = \langle \vec{a} - \vec{b}, \vec{a} - \vec{b} \rangle = \langle \vec{a}, \vec{a} \rangle + \langle \vec{b}, \vec{b} \rangle - 2\langle \vec{a}, \vec{b} \rangle = \|\vec{a}\|_2^2 + \|\vec{b}\|_2^2 - 2\langle \vec{a}, \vec{b} \rangle. \tag{1}$$

And hence the squared Euclidean distance between any pair of the points defined by these ε-orthogonal vectors falls in $2(1 \pm \varepsilon)$. Which solves our problem of dimension reduction for the "uniform metric" with all unit distances.

To summarize, if we wanted n points exactly at unit (Euclidean) distance from each other, we would need $n-1$ dimensions. (Think of a triangle in 2-dims.) But if we wanted to pack in n points which were at distance $(1 \pm \varepsilon)$ from each other, we could pack them into

$$O\left(\frac{\log n}{\varepsilon^2}\right)$$

dimensions.

1.2 The Johnson Lindenstrauss lemma

The Johnson Lindenstrauss "flattening" lemma says that such a claim is true not just for equidistant points, but for any set of n points in Euclidean space:

Lemma 1 (JL Flattening Lemma) *Let $\varepsilon \in (0, 1/2)$. Given any set of points $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ in $\mathbb{R}^D$, there exists a map $S : \mathbb{R}^D \to \mathbb{R}^k$ with $k = O(\varepsilon^{-2} \log n)$ such that*

$$1 - \varepsilon \leq \frac{\|S\mathbf{x}_i - S\mathbf{x}_j\|_2^2}{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2} \leq 1 + \varepsilon.$$

Note that the target dimension k is independent of the original dimension D, and depends only on the number of points n and the accuracy parameter ε. (Aside: this lemma is tight up to the constant term: it is easy to see that we need at least $\Omega(\frac{1}{\varepsilon} \log n)$ using a packing argument. Noga Alon showed a lower bound of $\Omega(\frac{\log n}{\varepsilon^2 \log 1/\varepsilon})$.)

1.3 The construction

The JL lemma is pretty surprising, but the construction of the map is perhaps even more surprising: it is a super-simple random construction. Let M be a $k \times D$ matrix, such that every entry of M is filled with an i.i.d. draw from a standard normal $N(0, 1)$ distribution (a.k.a. the "Gaussian" distribution). Define

$$S := \frac{1}{\sqrt{k}} M.$$

The point $\mathbf{x} \in \mathbb{R}^D$ is mapped to $S\mathbf{x}$. That's it. You hit the vector $\mathbf{x}$ with a Gaussian matrix M, and scale it down by $\sqrt{k}$. That's the map S. Nice and easy.

Note that it is a *linear* map: $S\mathbf{x} + S\mathbf{y} = S(\mathbf{x} + \mathbf{y})$. Hence, I claim we'd get a proof of Lemma 1 if we could show the following "sketch" lemma:

Lemma 2 (The JL Sketch) *Let $\varepsilon \in (0, 1/2)$. If S is constructed as above with $k = O(\varepsilon^{-2} \log \delta^{-1})$, and $\mathbf{x} \in \mathbb{R}^D$ is a unit vector (i.e., $\|x\|_2 = 1$), then*

$$\Pr[\|S\mathbf{x}\|_2^2 \in (1 \pm \varepsilon)] \geq 1 - \delta.$$

Why? Set $\delta = 1/n^2$, and hence $k = O(\varepsilon^{-2} \log n)$. Now for each $\mathbf{x}_i, \mathbf{x}_j \in X$ we get that the squared length of the unit vector $\frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|_2}$ is maintained to within $1 \pm \varepsilon$ with probability at least $1 - 1/n^2$. Since the map is linear, we know that $S(\alpha\mathbf{x}) = \alpha S\mathbf{x}$, and hence the squared length of the non-unit vector $\mathbf{x}_i - \mathbf{x}_j$ is in $(1 \pm \varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$ with probability $1/n^2$. Now by a union bound, all $\binom{n}{2}$ pairs of squared lengths in $\binom{X}{2}$ are maintained with probability at least $1 - \binom{n}{2}\frac{1}{n^2} \geq 1/2$. This proves Lemma 1.

A few comments about this construction:

- The above proof shows not only the existence of a good map, we also get that a random map as above works with constant probability! In other words, a Monte-Carlo randomized algorithm for dimension reduction. (Since we can efficiently check that the distances are preserved to within the prescribed bounds, we can convert this into a Las Vegas algorithm.)
- The algorithm (at least the Monte Carlo version) *does not even look* at the set of points X: it works for any set X with high probability. Hence, we can pick this map A before the points in X arrive.

1.4 The proof

Now, on to the proof of Lemma 2. Here's the main idea. Imagine that the vector we're considering is just the elementary unit vector $\mathbf{e}_1 = (1, 0, \ldots, 0)$. Then $M\,\mathbf{e}_1$ is just a vector with independent and identical Gaussian values, and we're interested in its length—the sum of squares of these Gaussians. If these were bounded r.v.s, we'd be done—but they are not. However, their tails are very small, so things should work out.

What I mean is: look at a Gaussian $N(0, 1)$ r.v.: it's density looks like this:

Which is not too different from this (bounded) random variable, if you squint a bit (a lot):

And the square of this r.v. has constant mean. So, if we take a sum of a bunch of squares of such random variables, it should concentrate strongly around pretty much like its mean (which is $\propto k$). The concentration is because of a Hoeffding-like argument. And so the length is concentrated close to $\sqrt{k}$, which explains the division by $\sqrt{k}$.

Now we just need to make all this precise, and remove the assumption that the vector was just $\mathbf{e}_1$. That's what the rest of the formal proof does: it has a few steps, but each of them is fairly elementary. And the technique is very general.

1.5 The proof, this time for real

We'll be using basic facts about Gaussians, let's just recall them. The probability density function for the Gaussian $N(\mu, \sigma^2)$ is

$$f(x) = \tfrac{1}{\sqrt{2\pi}\sigma} e^{\frac{(x-\mu)^2}{2\sigma^2}}.$$

We also use the following; the proof just needs some elbow grease.

Proposition 3 *If $Y_1 \sim N(\mu_1, \sigma_1^2)$ and $Y_2 \sim N(\mu_2, \sigma_2^2)$ are independent, then*

$$Y_1 + Y_2 \sim N(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2).$$

Recall that we want to argue about the squared length of $S\mathbf{x} \in \mathbb{R}^k$. To start off, observe that each coordinate of the vector $M\mathbf{x}$ behaves like

$$Y \sim (G_1, G_2, \ldots, G_D) \cdot \mathbf{x} = \sum x_i\, G_i$$

where the G_i's are i.i.d. $N(0,1)$ r.v.s. But then the proposition tells us that $Y \sim N(0, x_1^2 + x_2^2 + \ldots + x_D^2)$. And since $\mathbf{x}$ is a unit length vector, this is simply $N(0,1)$. So, each of the k coordinates of $M\mathbf{x}$ behaves just like an independent Gaussian!

What is the squared length of $S\mathbf{x} = \frac{1}{\sqrt{k}} M\mathbf{x}$, then? It is

$$Z := \sum_{i=1}^{k} \frac{1}{k} \cdot Y_i^2$$

where each $Y_i \sim N(0,1)$, independent of the others. And since $E[Y_i^2] = \mathrm{Var}(Y_i) + E[Y_i]^2 = 1$, we get $E[Z] = 1$.

Now to show that Z does not deviate too much from 1. And Z is the sum of a bunch of independent and identical random variables. If only the Y_i's were all bounded, we could have used a Hoeffding bound and be done. But these are not bounded r.v.s, so we'll need to do a little work. The concentration bound we will prove in the next (optional) subsection is:

Theorem 4 *For Z defined above,*

$$\Pr[Z \geq (1+\varepsilon)] \leq e^{-k\varepsilon^2/8}.$$
$$\Pr[Z \leq (1-\varepsilon)] \leq e^{-k\varepsilon^2/8}.$$

To recap, we observed that $\|S\mathbf{x}\|_2^2$ is distributed like a sum of squares of Gaussians, and then the above concentration bound for such random variables shows that

$$\Pr[\|S\mathbf{x}\|_2^2 \notin 1 \pm \varepsilon] \leq \exp(-k\varepsilon^2/8) \leq \delta/2$$

for $k = \frac{8}{\varepsilon^2} \ln \frac{2}{\delta}$. This finishes the proof of Lemma 2.

1.5.1 Proof of Theorem 4⋆

For the "upper tail", we look at:

$$\Pr[Z \geq 1+\varepsilon] \leq \Pr[e^{tkZ} \geq e^{tk(1+\varepsilon)}] \leq E[e^{tkZ}]/e^{tk(1+\varepsilon)} = \prod_i \left(E[e^{tY_i^2}]/e^{t(1+\varepsilon)} \right) \tag{2}$$

for every $t > 0$. And what is $E[e^{tY^2}]$ for $Y \sim N(0,1)$? Let's calculate it:

$$\frac{1}{\sqrt{2\pi}} \int_y e^{ty^2} e^{-y^2/2} dy = \frac{1}{\sqrt{2\pi}} \int_z e^{-z^2/2} \frac{dz}{\sqrt{1-2t}} = \frac{1}{\sqrt{1-2t}}. \tag{3}$$

for $t < 1/2$. So our current bound on the upper tail is that for all $t \in (0, 1/2)$ we have

$$\Pr[Z \geq (1+\varepsilon)] \leq \left(\frac{1}{e^{t(1+\varepsilon)}\sqrt{1-2t}} \right)^k.$$

Let's just focus on part of this expression:

$$\begin{aligned}
\left(\frac{1}{e^t \sqrt{1-2t}} \right) &= \exp\left(-t - \frac{1}{2}\log(1-2t) \right) \\
&= \exp\left((2t)^2/4 + (2t)^3/6 + \cdots \right) \leq \exp\left(t^2(1 + 2t + 2t^2 + \cdots) \right) \\
&= \exp(t^2/(1-2t)).
\end{aligned}$$

Plugging this back, we get

$$\Pr[Z \geq (1+\varepsilon)] \leq \left(\frac{1}{e^{t(1+\varepsilon)}\sqrt{1-2t}}\right)^k$$
$$\leq \exp(kt^2/(1-2t) - kt\varepsilon) \leq e^{-k\varepsilon^2/8},$$

if we set $t = \varepsilon/4$ and use the fact that $1-2t \geq 1/2$ for $\varepsilon \leq 1/2$. (Note: this setting of t also satisfies $t \in (0, 1/2)$, which we needed from our previous calculations.) A similar calculation bounds the lower tail.

A different proof observes that the squares of Gaussians are chi-squared r.v.s, the sum of k of them is *chi-squared with k degrees of freedom*, and the Internet conveniently has concentration bounds for these things.)

Citations: The JL Lemma was first proved in this paper of Bill Johnson and Joram Lindenstrauss. There have been several proofs after theirs, usually trying to tighten their results, or simplify the algorithm/proof (see citations in some of the newer papers): the proof is some combination of the proofs in this STOC '98 paper of Piotr Indyk and Rajeev Motwani, and this paper by Sanjoy Dasgupta and myself.

1.6 Extensions

There is a lot of recent work on JL transforms and their variants: how do use less randomness, do them faster, get better parameters for well-behaved point sets, etc. Here are some pointers. For more recent work, look online, or come by and I can point you to more resources.

- **Fewer Random Bits.** Instead of the entries of the $k \times D$ matrix M being Gaussians, we could have chosen them to be unbiased $\{-1, +1\}$ r.v.s. The claim in Lemma 2 goes through almost unchanged! And one can do better: this is related to speeding up the process (by making the projection matrix S sparse and better behaved) — read the next bullet point.

- **Fast JL Transform.** Ideally we would like M to be very sparse. Then we would need to choose very few random variables. Also, if M were sparse, then we could compute $S\mathbf{x} = \frac{1}{\sqrt{k}}M\mathbf{x}$ in time much smaller than $O(kD)$. But sparsity in M creates a problem. In this case, if the vector $\mathbf{x}$ is also sparse, we start getting into trouble. Look at any entry $(M\mathbf{x})_i = \sum_j M_{ij}x_j$—most entries in the sum will be zero, so this sum will depend on very few entries. And hence the length $\|M\mathbf{x}\|_2^2$ will have high variance. To control this variance we will need to pick k much larger.

  The idea (due to Ailon and Chazelle) is to do a length-preserving transformation that converts sparse vectors into dense ones. This is the discrete Fourier (or Hadamard) transform. The idea is that due to the "uncertainty principle", a sparse vector will be dense after the Fourier transform. And hence one can preprocess sparse vectors to make them dense, and then hit them with a sparse M matrix.

  Of course, the problem is that dense vectors may become sparse due to the transform, so you need to be careful. It turns out a little randomness can take care of this. Indeed, to create one JL transform that handles all vectors (not just sparse or dense ones), they propose the following operation:

$$T\mathbf{x} := \frac{1}{\sqrt{kD}}PH_DR\mathbf{x}$$

where R is a random $\{+1, -1\}$ valued *diagonal* matrix, H_D is the Hadamard matrix of size $D \times D$, and P is a sparse Gaussian $k \times D$ matrix picked with the right parameters. Since (a) R is a diagonal matrix, (b) we can do the fast Fourier transform in $O(D \log D)$ time, and (c) P is a sparse matrix, the entire transformation $x \to T\mathbf{x}$ can be done in $O(D \log D + k \log^2 D)$ time, much faster than the naive $O(kD)$ time.

- **Manifolds.** As mentioned above, there is an (almost) matching lower bound on the trade-off between dimension and the stretch. If you want $(1 \pm \varepsilon)$ then you cannot do better than $\Omega(\frac{\log n}{\varepsilon^2 \log 1/\varepsilon})$. This is even true for n points all at distance 1 from each other.

  But if your data is "well-behaved" you might do better. E.g., if your data lies on a low-dimensional manifold whose curvature is not large, then Clarkson shows better parameters work for JL.

2 Concentration of Measure in High Dimensions

Last lecture we said that if we picked a point at random from the surface of the Euclidean unit ball in D dimensions, it would be very close to the equator. Let's give some intuition for this fact.

First of all, how do you pick a point at random from the surface of the D-dimensional unit ball? One way is to pick a vector $\mathbf{g}$ from the multivariate Gaussian distribution (with the identity I_D as the covariance matrix) and normalize it to have length 1. Because the distribution has spherical symmetry, you get a point $\mathbf{x}$ uniformly from the surface of the unit ball.

Actually, we claim that the length of the random vector is pretty close to $\sqrt{D}$, so just taking $\mathbf{x} = \frac{1}{\sqrt{D}}\mathbf{g}$ has length $1 \pm \varepsilon)$ with high probability. This is precisely what Theorem 4 shows. And this vector is still spherically symmetric. So it's not quite on the surface, but is pretty close. Equivalently, $\mathbf{g}$ is close to the surface of a radius $\sqrt{D}$ ball in $\mathbb{R}^D$. Scaling things up, we want to show it is very likely that $\mathbf{g}$ will be at distance at most $\sqrt{D} \cdot \frac{1}{\sqrt{D}} = 1$ from the equator.

Now, let's consider the equator given by the intersection of the ball with the normal subspace to the $\mathbf{e}_1 = (1, 0, 0, \ldots, 0)$ vector. So what random choices of $\mathbf{g}$ are at distance more than 1 from the equatorial plane? This distance is just the first coordinate $g_1 \sim N(0, 1)$. So we're just interested in $\Pr[g_1 \geq 1]$. This is the probability of a normal random variable lying beyond its standard deviation σ, which is small, and falls exponentially as we consider lying beyond $2\sigma, 3\sigma$, etc.