COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Cameron Musco

University of Massachusetts Amherst. Fall 2021.

Lecture 12

LOGISTICS

- Problem Set 2 is due Friday, 11:59pm.
- Quiz 6 is due today at 8pm.
- The exam will be held next Tuesday in class. Let me know ASAP if you need accommodations (e.g., extended time).
- We will do some midterm review in class on Thursday. I will also hold additional office hours for midterm prep, **next Monday, 4-6pm**, and potentially Friday afternoon as well.

Last Class: The Johnson-Lindenstrauss Lemma

- Low-distortion embeddings for any set of points via random projection.
- Started on proof of the JL Lemma via the Distributional JL Lemma.

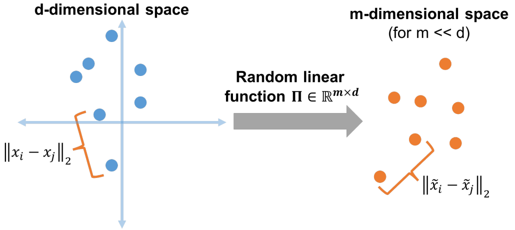
This Class:

- Finish Up proof of the JL lemma.
- Example applications to classification and clustering.
- Discuss connections to high dimensional geometry.

Johnson-Lindenstrauss Lemma: For any set of points $\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$ and $\epsilon > 0$ there exists a linear map $\mathbf{\Pi} : \mathbb{R}^d \to R^m$ such that $m = O\left(\frac{\log n}{\epsilon^2}\right)$ and letting $\tilde{x}_i = \mathbf{\Pi}\vec{x}_i$:

$$\text{For all } i,j: \ (1-\epsilon)\|\vec{x}_i - \vec{x}_j\|_2 \leq \|\tilde{x}_i - \tilde{x}_j\|_2 \leq (1+\epsilon)\|\vec{x}_i - \vec{x}_j\|_2.$$

Further, if $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$ has each entry chosen i.i.d. from $\mathcal{N}(0, 1/m)$ and $m = O\left(\frac{\log n/\delta}{\epsilon^2}\right)$, $\mathbf{\Pi}$ satisfies the guarantee with probability $\geq 1 - \delta$.

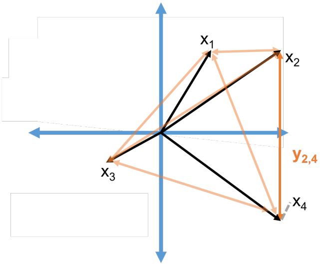


We showed that the Johnson-Lindenstrauss Lemma follows from:

Distributional JL Lemma: Let $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$ have each entry chosen i.i.d. as $\mathcal{N}(0, 1/m)$. If we set $m = O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$, then for any $\vec{y} \in \mathbb{R}^d$, with probability $\geq 1 - \delta$

$$(1-\epsilon)\|\vec{y}\|_2 \leq \|\mathbf{\Pi}\vec{y}\|_2 \leq (1+\epsilon)\|\vec{y}\|_2.$$

Main Idea: Union bound over $\binom{n}{2}$ difference vectors $\vec{y}_{ij} = \vec{x}_i - \vec{x}_j$.

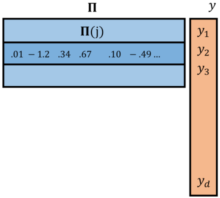


DISTRIBUTIONAL JL PROOF

Distributional JL Lemma: Let $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$ have each entry chosen i.i.d. as $\mathcal{N}(0, 1/m)$. If we set $m = O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$, then for any $\vec{y} \in \mathbb{R}^d$, with probability $\geq 1 - \delta$

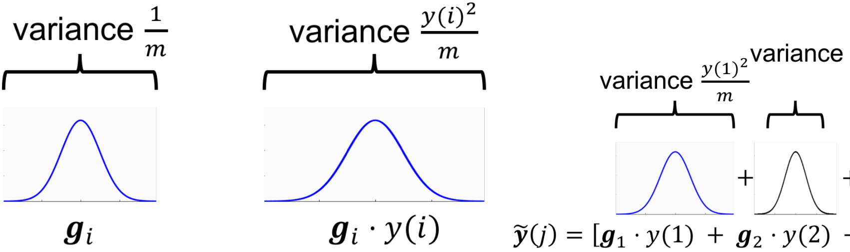
$$(1-\epsilon)\|\vec{y}\|_2 \leq \|\mathbf{\Pi}\vec{y}\|_2 \leq (1+\epsilon)\|\vec{y}\|_2$$

- Let $\tilde{\mathbf{y}}$ denote $\mathbf{\Pi}\vec{y}$ and let $\mathbf{\Pi}(j)$ denote the j^{th} row of $\mathbf{\Pi}$.
- For any j, $\tilde{\mathbf{y}}(j) = \langle \mathbf{\Pi}(j), \vec{y} \rangle = \sum_{i=1}^{d} \mathbf{g}_i \cdot \vec{y}(i)$ where $\mathbf{g}_i \sim \mathcal{N}(0, 1/m)$.



$\vec{y} \in \mathbb{R}^d$: arbitrary vector, $\tilde{\mathbf{y}} \in \mathbb{R}^m$: compressed vector, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection. d: original dim. m: compressed dim, ϵ: error, δ: failure prob.

- Let $\tilde{\mathbf{y}}$ denote $\mathbf{\Pi}\vec{y}$ and let $\mathbf{\Pi}(j)$ denote the j^{th} row of $\mathbf{\Pi}$.
- For any j, $\tilde{\mathbf{y}}(j) = \langle \mathbf{\Pi}(j), \vec{y} \rangle = \sum_{i=1}^{d} \mathbf{g}_i \cdot \vec{y}(i)$ where $\mathbf{g}_i \sim \mathcal{N}(0, 1/m)$.
- $\mathbf{g}_i \cdot \vec{y}(i) \sim \mathcal{N}(0, \frac{\vec{y}(i)^2}{m})$: normally distributed with variance $\frac{\vec{y}(i)^2}{m}$.



What is the distribution of $\tilde{\mathbf{y}}(j)$? Also Gaussian!

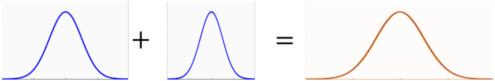
$\vec{y} \in \mathbb{R}^d$: arbitrary vector, $\tilde{\mathbf{y}} \in \mathbb{R}^m$: compressed vector, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection mapping $\vec{y} \to \tilde{\mathbf{y}}$. $\mathbf{\Pi}(j)$: j^{th} row of $\mathbf{\Pi}$, d: original dimension. m: compressed dimension. $\mathbf{g}_i$: normally distributed random variable

Letting $\tilde{\mathbf{y}} = \mathbf{\Pi}\vec{y}$, we have $\tilde{\mathbf{y}}(j) = \langle \mathbf{\Pi}(j), \vec{y} \rangle$ and:

$$\tilde{\mathbf{y}}(j) = \sum_{i=1}^{d} \mathbf{g}_i \cdot \vec{y}(i) \text{ where } \mathbf{g}_i \cdot \vec{y}(i) \sim \mathcal{N}\left(0, \frac{\vec{y}(i)^2}{m}\right).$$

Stability of Gaussian Random Variables. For independent $a \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $b \sim \mathcal{N}(\mu_2, \sigma_2^2)$ we have:

$$a + b \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$$



Thus, $\tilde{\mathbf{y}}(j) \sim \mathcal{N}(0, \frac{\vec{y}(1)^2}{m} + \frac{\vec{y}(2)^2}{m} + \ldots + \frac{\vec{y}(d)^2}{m}\ \frac{\|\vec{y}\|_2^2}{m})$ I.e., $\tilde{\mathbf{y}}$ itself is a random Gaussian vector. Rotational invariance of the Gaussian distribution.

$\vec{y} \in \mathbb{R}^d$: arbitrary vector, $\tilde{\mathbf{y}} \in \mathbb{R}^m$: compressed vector, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random

So far: Letting $\mathbf{\Pi} \in \mathbb{R}^{d \times m}$ have each entry chosen i.i.d. as $\mathcal{N}(0, 1/m)$, for any $\vec{y} \in \mathbb{R}^d$, letting $\tilde{\mathbf{y}} = \mathbf{\Pi}\vec{y}$:

$$\tilde{\mathbf{y}}(j) \sim \mathcal{N}(0, \|\vec{y}\|_2^2/m).$$

What is $\mathbb{E}[\|\tilde{\mathbf{y}}\|_2^2]$?

$$\mathbb{E}[\|\tilde{\mathbf{y}}\|_2^2] = \mathbb{E}\left[\sum_{j=1}^m \tilde{\mathbf{y}}(j)^2\right] = \sum_{j=1}^m \mathbb{E}[\tilde{\mathbf{y}}(j)^2]$$

$$= \sum_{j=1}^m \frac{\|\vec{y}\|_2^2}{m} = \|\vec{y}\|_2^2$$

So $\tilde{\mathbf{y}}$ has the right norm in expectation.

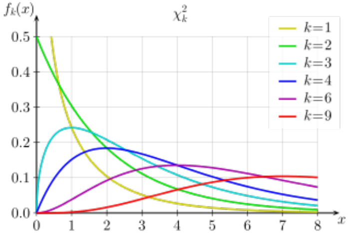
How is $\|\tilde{\mathbf{y}}\|_2^2$ distributed? Does it concentrate?

$\vec{y} \in \mathbb{R}^d$: arbitrary vector, $\tilde{\mathbf{y}} \in \mathbb{R}^m$: compressed vector, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection mapping $\vec{y} \to \tilde{\mathbf{y}}$. $\mathbf{\Pi}(j)$: j^{th} row of $\mathbf{\Pi}$, d: original dimension. m: compressed dimension, $\mathbf{g}_i$: normally distributed random variable

So far: Letting $\mathbf{\Pi} \in \mathbb{R}^{d \times m}$ have each entry chosen i.i.d. as $\mathcal{N}(0, 1/m)$, for any $\vec{y} \in \mathbb{R}^d$, letting $\tilde{\mathbf{y}} = \mathbf{\Pi}\vec{y}$:

$$\tilde{\mathbf{y}}(j) \sim \mathcal{N}(0, \|\vec{y}\|_2^2/m) \text{ and } \mathbb{E}[\|\tilde{\mathbf{y}}\|_2^2] = \|\vec{y}\|_2^2$$

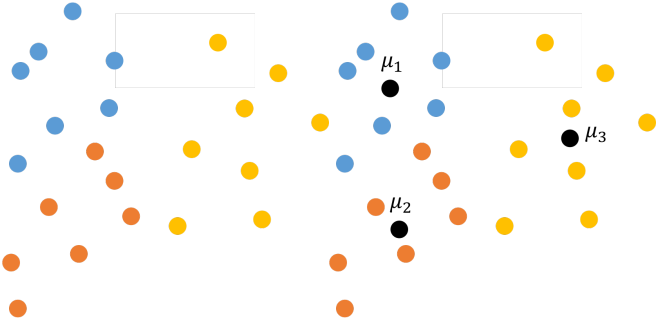
$\|\tilde{\mathbf{y}}\|_2^2 = \sum_{i=1}^m \tilde{\mathbf{y}}(j)^2$ a Chi-Squared random variable with m degrees of freedom (a sum of m squared independent Gaussians)



Lemma: (Chi-Squared Concentration) Letting $\mathbf{Z}$ be a Chi-Squared random variable with m degrees of freedom,

$$\Pr\left[|\mathbf{Z} - \mathbb{E}\mathbf{Z}| \geq \epsilon \mathbb{E}\mathbf{Z}\right] \leq 2e^{-m\epsilon^2/8}.$$

Goal: Separate n points in d dimensional space into k groups.



k-means Objective: $Cost(\mathcal{C}_1, \ldots, \mathcal{C}_k) = \min_{\mathcal{C}_1, \ldots \mathcal{C}_k} \sum_{j=1}^{k} \sum_{\vec{x} \in \mathcal{C}_k} \|\vec{x} - \mu_j\|_2^2.$

Write in terms of distances:

$$Cost(\mathcal{C}_1, \ldots, \mathcal{C}_k) = \min_{\mathcal{C}_1, \ldots \mathcal{C}_k} \sum_{j=1}^{k} \sum_{\vec{x}_1, \vec{x}_2 \in \mathcal{C}_k} \|\vec{x}_1 - \vec{x}_2\|_2^2$$

k-means Objective: $Cost(\mathcal{C}_1, \ldots, \mathcal{C}_k) = \min_{\mathcal{C}_1, \ldots \mathcal{C}_k} \sum_{j=1}^{k} \sum_{\vec{x}_1, \vec{x}_2 \in \mathcal{C}_k} \|\vec{x}_1 - \vec{x}_2\|_2^2$

If we randomly project to $m = O\left(\frac{\log n}{\epsilon^2}\right)$ dimensions, for all pairs $\vec{x}_1, \vec{x}_2$,

$$(1-\epsilon)\|\vec{x}_1 - \vec{x}_2\|_2^2 \leq \|\tilde{\mathbf{x}}_1 - \tilde{\mathbf{x}}_2\|_2^2 \leq (1+\epsilon)\|\vec{x}_1 - \vec{x}_2\|_2^2 \implies$$

Letting $\overline{Cost}(\mathcal{C}_1, \ldots, \mathcal{C}_k) = \min_{\mathcal{C}_1, \ldots \mathcal{C}_k} \sum_{j=1}^{k} \sum_{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2 \in \mathcal{C}_k} \|\tilde{\mathbf{x}}_1 - \tilde{\mathbf{x}}_2\|_2^2$

$$(1-\epsilon)Cost(\mathcal{C}_1, \ldots, \mathcal{C}_k) \leq \overline{Cost}(\mathcal{C}_1, \ldots, \mathcal{C}_k) \leq (1+\epsilon)Cost(\mathcal{C}_1, \ldots, \mathcal{C}_k).$$

Upshot: Can cluster in m dimensional space (much more efficiently) and minimize $\overline{Cost}(\mathcal{C}_1, \ldots, \mathcal{C}_k)$. The optimal set of clusters will have true cost within $1 + c\epsilon$ times the true optimal. Good exercise to prove this.

The Johnson-Lindenstrauss Lemma and High Dimensional Geometry

- High-dimensional Euclidean space looks *very different* from low-dimensional space. So how can JL work?
- Is Euclidean distance in high-dimensional meaningless, making JL useless? (The curse of dimensionality)

What is the largest set of mutually orthogonal unit vectors in d-dimensional space?

a) 1 b) $\log d$ c) $\sqrt{d}$ d) d

What is the largest set of unit vectors in d-dimensional space that have all pairwise dot products $|\langle \vec{x}, \vec{y} \rangle| \leq \epsilon$? (think $\epsilon = .01$)

a) d b) $\Theta(d)$ c) $\Theta(d^2)$ d) $2^{\Theta(d)}$

In fact, an exponentially large set of random vectors will be nearly pairwise orthogonal with high probability!

Claim: $2^{\Theta(\epsilon^2 d)}$ random d-dimensional unit vectors will have all pairwise dot products $|\langle \vec{x}, \vec{y} \rangle| \leq \epsilon$ (be nearly orthogonal) with high probability.

Proof: Let $\vec{x}_1, \ldots, \vec{x}_t$ each have independent random entries set to $\pm 1/\sqrt{d}$.

- What is $\|\vec{x}_i\|_2$? Every $\vec{x}_i$ is always a unit vector.
- What is $\mathbb{E}[\langle \vec{x}_i, \vec{x}_j \rangle]$? $\mathbb{E}[\langle \vec{x}_i, \vec{x}_j \rangle] = 0$
- By a Chernoff bound, $\Pr[|\langle \vec{x}_i, \vec{x}_j \rangle| \geq \epsilon] \leq 2e^{-\epsilon^2 d/6}$ (great exercise).
- If we chose $t = \frac{1}{2} e^{\epsilon^2 d/12}$, using a union bound over all $\binom{t}{2} \leq \frac{1}{8} e^{\epsilon^2 d/6}$ possible pairs, with probability $\geq 3/4$ all will be nearly orthogonal.

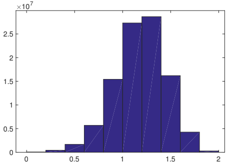
Up Shot: In d-dimensional space, a set of $2^{\Theta(\epsilon^2 d)}$ random unit vectors have all pairwise dot products at most ϵ (think $\epsilon = .01$)

$$\|\vec{x}_i - \vec{x}_j\|_2^2 = \|\vec{x}_i\|_2^2 + \|\vec{x}_j\|_2^2 - 2\vec{x}_i^T\vec{x}_j \in [1.98, 2.02].$$

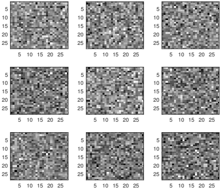
Even with an exponential number of random vector samples, we don't see any nearby vectors.

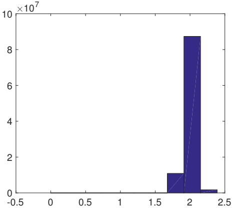
- One version of the 'curse of dimensionality'.
- If all your distances are roughly the same, distance based methods (k-means clustering, nearest neighbors, SVMs, etc.) aren't going to work well.
- Distances are only meaningful if we have lots of structure and our data isn't just independent random vectors.

Distances for MNIST Digits:



Distances for Random Images:





Another Interpretation: Tells us that random data can be a very bad model for actual input data.