COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Cameron Musco

University of Massachusetts Amherst. Fall 2019.

Lecture 10

LOGISTICS

- Problem Set 2 is due next Friday 10/11, although we will allow submissions until Sunday 10/13 at midnight with no penalty.
- Midterm on Thursday 10/17. Will cover material through today.

Last Class: Dimensionality Reduction

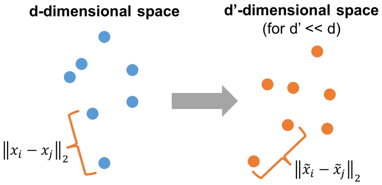
- Applications and examples of dimensionality reduction in data science.
- Low-distortion embeddings (MinHash as an example).
- Low-distortion embeddings for Euclidean space and the Johnson-Lindenstrauss Lemma.

This Class: Finish the JL Lemma.

- Prove the Johnson-Lindenstrauss Lemma.
- Discuss algorithmic considerations, connections to other methods, etc.

Low Distortion Embedding for Euclidean Space: Given $x_1, \ldots, x_n \in \mathbb{R}^d$ and error parameter $\epsilon \geq 0$, find $\tilde{x}_1, \ldots, \tilde{x}_n \in \mathbb{R}^{d'}$ (where $d' \ll d$) such that for all $i, j \in [n]$:

$$(1-\epsilon)\|x_i - x_j\|_2 \leq \|\tilde{x}_i - \tilde{x}_j\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2$$



If $x_1, \ldots, x_n$ lie in a k-dimensional subspace of $\mathbb{R}^d$ can project to $d' = k$ dimensions with no distortion.

If close to a k-dimensional space, can project to k dimensions without much distortion (the idea behind PCA).

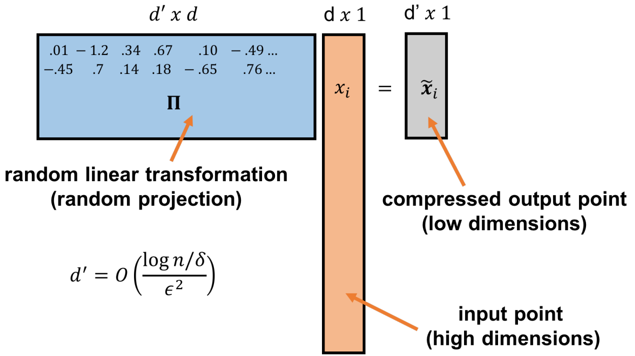
Johnson-Lindenstrauss Lemma: Let $\mathbf{\Pi} \in \mathbb{R}^{d' \times d}$ have each entry chosen i.i.d. as $\frac{1}{\sqrt{d'}} \cdot \mathcal{N}(0,1)$. For any set of points $x_1, \ldots, x_n \in \mathbb{R}^d$, $\epsilon, \delta > 0$, and $d' = O\left(\frac{\log(n/\delta)}{\epsilon^2}\right)$, letting $\tilde{\mathbf{x}}_\mathbf{i} = \mathbf{\Pi} x_i$, with probability $\geq 1 - \delta$ we have:

For all i, j : $(1-\epsilon)\|x_i - x_j\|_2 \leq \|\tilde{\mathbf{x}}_\mathbf{i} - \tilde{\mathbf{x}}_\mathbf{j}\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2$.

Surprising and powerful result.

- Construction of $\mathbf{\Pi}$ is simple, random and data oblivious.

$x_1, \ldots, x_n$: original data points (d dimensions), $\tilde{\mathbf{x}}_\mathbf{1}, \ldots, \tilde{\mathbf{x}}_\mathbf{n}$: compressed data points ($d' < d$ dimensions), $\mathbf{\Pi} \in \mathbb{R}^{d' \times d}$: random projection matrix (embedding function), ϵ: error of embedding, δ: failure probability.



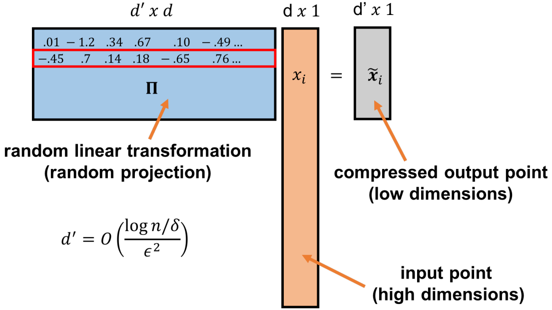
$\mathbf{\Pi} \in \mathbb{R}^{d' \times d}$ is a random matrix. I.e., a random function mapping length d vectors to length d' vectors.

$x_1, \ldots, x_n$: original points (d dims.), $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n$: compressed points ($d' < d$ dims.), $\mathbf{\Pi}$: random projection (embedding function), ϵ: error of embedding.

Compression operation is $\tilde{\mathbf{x}}_i = \mathbf{\Pi} x_i$, so for any j,

$$\tilde{\mathbf{x}}_i(j) = \langle \mathbf{\Pi}(j), x_i \rangle = \sum_{k=1}^{d} \mathbf{\Pi}(j,k) \cdot x_i(k).$$

$\mathbf{\Pi}(j)$ is a vector with independent random Gaussian entries.

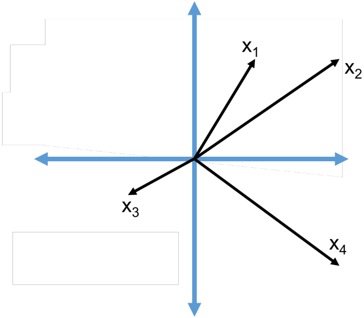


$x_1, \ldots, x_n$: original points (d dims.), $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n$: compressed points ($d' < d$ dims.), $\mathbf{\Pi} \in \mathbb{R}^{d' \times d}$: random projection (embedding function)

Compression operation is $\tilde{\mathbf{x}}_{\mathbf{i}} = \mathbf{\Pi} x_i$, so for any j,

$$\tilde{\mathbf{x}}_{\mathbf{i}}(j) = \langle \mathbf{\Pi}(j), x_i \rangle = \sum_{k=1}^{d} \mathbf{\Pi}(j,k) \cdot x_i(k).$$

$\mathbf{\Pi}(j)$ is a vector with independent random Gaussian entries.

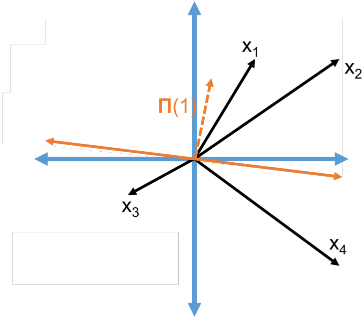


$x_1, \ldots, x_n$: original points (d dims.), $\tilde{\mathbf{x}}_{\mathbf{1}}, \ldots, \tilde{\mathbf{x}}_{\mathbf{n}}$: compressed points ($d' < d$ dims.), $\mathbf{\Pi} \in \mathbb{R}^{d' \times d}$: random projection (embedding function)

Compression operation is $\tilde{\mathbf{x}}_\mathbf{i} = \mathbf{\Pi} x_i$, so for any j,

$$\tilde{\mathbf{x}}_\mathbf{i}(j) = \langle \mathbf{\Pi}(j), x_i \rangle = \sum_{k=1}^{d} \mathbf{\Pi}(j,k) \cdot x_i(k).$$

$\mathbf{\Pi}(j)$ is a vector with independent random Gaussian entries.

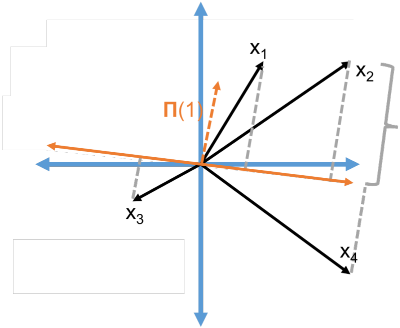


$x_1, \ldots, x_n$: original points (d dims.), $\tilde{\mathbf{x}}_\mathbf{1}, \ldots, \tilde{\mathbf{x}}_\mathbf{n}$: compressed points ($d' < d$ dims.), $\mathbf{\Pi} \in \mathbb{R}^{d' \times d}$: random projection (embedding function)

Compression operation is $\tilde{\mathbf{x}}_\mathbf{i} = \mathbf{\Pi} x_i$, so for any j,

$$\tilde{\mathbf{x}}_\mathbf{i}(j) = \langle \mathbf{\Pi}(j), x_i \rangle = \sum_{k=1}^{d} \mathbf{\Pi}(j,k) \cdot x_i(k).$$

$\mathbf{\Pi}(j)$ is a vector with independent random Gaussian entries.



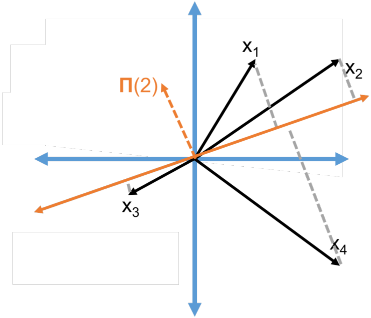
$$\frac{\langle x_2, \mathbf{\Pi}(1) \rangle}{\|\mathbf{\Pi}(1)\|} = \frac{\tilde{\mathbf{x}}_2(1)}{\|\mathbf{\Pi}(1)\|}$$

$x_1, \ldots, x_n$: original points (d dims.), $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_\mathbf{n}$: compressed points ($d' < d$ dims.), $\mathbf{\Pi} \in \mathbb{R}^{d' \times d}$: random projection (embedding function)

Compression operation is $\tilde{\mathbf{x}}_\mathbf{i} = \mathbf{\Pi} x_i$, so for any j,

$$\tilde{\mathbf{x}}_\mathbf{i}(j) = \langle \mathbf{\Pi}(j), x_i \rangle = \sum_{k=1}^{d} \mathbf{\Pi}(j,k) \cdot x_i(k).$$

$\mathbf{\Pi}(j)$ is a vector with independent random Gaussian entries.

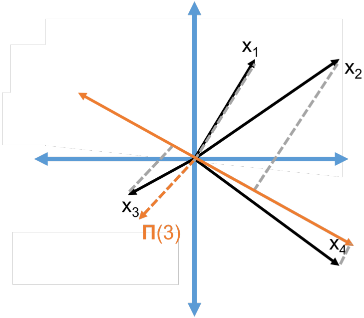


$x_1, \ldots, x_n$: original points (d dims.), $\tilde{\mathbf{x}}_\mathbf{1}, \ldots, \tilde{\mathbf{x}}_\mathbf{n}$: compressed points ($d' < d$ dims.), $\mathbf{\Pi} \in \mathbb{R}^{d' \times d}$: random projection (embedding function)

Compression operation is $\tilde{\mathbf{x}}_{\mathbf{i}} = \mathbf{\Pi} x_i$, so for any j,

$$\tilde{\mathbf{x}}_{\mathbf{i}}(j) = \langle \mathbf{\Pi}(j), x_i \rangle = \sum_{k=1}^{d} \mathbf{\Pi}(j,k) \cdot x_i(k).$$

$\mathbf{\Pi}(j)$ is a vector with independent random Gaussian entries.

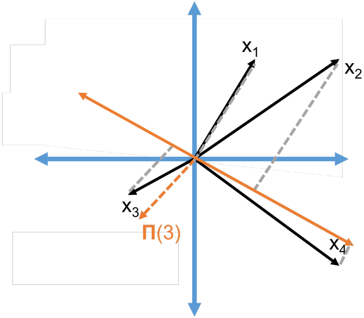


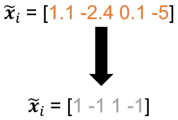
$x_1, \ldots, x_n$: original points (d dims.), $\tilde{\mathbf{x}}_{\mathbf{1}}, \ldots, \tilde{\mathbf{x}}_{\mathbf{n}}$: compressed points ($d' < d$ dims.), $\mathbf{\Pi} \in \mathbb{R}^{d' \times d}$: random projection (embedding function)

Compression operation is $\tilde{\mathbf{x}}_\mathbf{i} = \mathbf{\Pi} x_i$, so for any j,

$$\tilde{\mathbf{x}}_\mathbf{i}(j) = \langle \mathbf{\Pi}(j), x_i \rangle = \sum_{k=1}^{d} \mathbf{\Pi}(j,k) \cdot x_i(k).$$

$\mathbf{\Pi}(j)$ is a vector with independent random Gaussian entries.





Points with high cosine similarity have similar random projections.

Computing a length d' SimHash signature $SH_1(x_i), \ldots, SH_{d'}(x_i)$ is identical to computing $\tilde{\mathbf{x}}_\mathbf{i} = \mathbf{\Pi} x_i$ and then taking $sign(\tilde{\mathbf{x}}_\mathbf{i})$.

The Johnson-Lindenstrauss Lemma is a direct consequence of a closely related lemma:

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

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

Applying a random matrix $\mathbf{\Pi}$ to any vector y preserves y's norm with high probability.

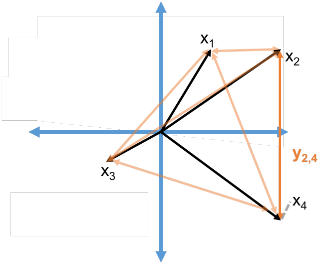
- Like a low-distortion embedding, but for the length of a compressed vector rather than distances between vectors.
- Can be proven from first principles. Will see next.

$\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection matrix. d: original dimension. m: compressed dimension (analogous to d'), ϵ: embedding error, δ: embedding failure prob.

Distributional JL Lemma $\implies$ JL Lemma: Distributional JL show that a random projection $\mathbf{\Pi}$ preserves the norm of any y. The main JL Lemma says that $\mathbf{\Pi}$ preserves distances between vectors.

Since $\mathbf{\Pi}$ is linear these are the same thing!

Proof: Given $x_1, \ldots, x_n$, define $\binom{n}{2}$ vectors y_{ij} where $y_{ij} = x_i - x_j$.



$x_1, \ldots, x_n$: original points, $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n$: compressed points, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection matrix. d: original dimension. m: compressed dimension (analogous to d'), ϵ: embedding error, δ: embedding failure prob.

Distributional JL Lemma $\implies$ JL Lemma: Distributional JL show that a random projection $\mathbf{\Pi}$ preserves the norm of any y. The main JL Lemma says that $\mathbf{\Pi}$ preserves distances between vectors.

Since $\mathbf{\Pi}$ is linear these are the same thing!

Proof: Given $x_1, \ldots, x_n$, define $\binom{n}{2}$ vectors y_{ij} where $y_{ij} = x_i - x_j$.

- If we choose $\mathbf{\Pi}$ with $m = O\left(\frac{\log 1/\delta}{\epsilon^2}\right)$, for each y_{ij} with probability $\geq 1 - \delta$ we have:

$$(1-\epsilon)\|y_{ij}\|_2 \leq \|\mathbf{\Pi} y_{ij}\|_2 \leq (1+\epsilon)\|y_{ij}\|_2$$

$x_1, \ldots, x_n$: original points, $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n$: compressed points, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection matrix. d: original dimension. m: compressed dimension (analogous to d'), ϵ: embedding error, δ: embedding failure prob.

Distributional JL Lemma $\Longrightarrow$ JL Lemma: Distributional JL show that a random projection $\mathbf{\Pi}$ preserves the norm of any y. The main JL Lemma says that $\mathbf{\Pi}$ preserves distances between vectors.

Since $\mathbf{\Pi}$ is linear these are the same thing!

Proof: Given $x_1, \ldots, x_n$, define $\binom{n}{2}$ vectors y_{ij} where $y_{ij} = x_i - x_j$.

- If we choose $\mathbf{\Pi}$ with $m = O\left(\frac{\log 1/\delta}{\epsilon^2}\right)$, for each y_{ij} with probability $\geq 1 - \delta$ we have:

$$(1-\epsilon)\|x_i - x_j\|_2 \leq \|\mathbf{\Pi}(x_i - x_j)\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2$$

$x_1, \ldots, x_n$: original points, $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n$: compressed points, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection matrix. d: original dimension. m: compressed dimension (analogous to d'), ϵ: embedding error, δ: embedding failure prob.

Distributional JL Lemma $\implies$ JL Lemma: Distributional JL show that a random projection $\mathbf{\Pi}$ preserves the norm of any y. The main JL Lemma says that $\mathbf{\Pi}$ preserves distances between vectors.

Since $\mathbf{\Pi}$ is linear these are the same thing!

Proof: Given $x_1, \ldots, x_n$, define $\binom{n}{2}$ vectors y_{ij} where $y_{ij} = x_i - x_j$.

- If we choose $\mathbf{\Pi}$ with $m = O\left(\frac{\log 1/\delta}{\epsilon^2}\right)$, for each y_{ij} with probability $\geq 1 - \delta$ we have:

$$(1-\epsilon)\|x_i - x_j\|_2 \leq \|\tilde{\mathbf{x}}_\mathbf{i} - \tilde{\mathbf{x}}_\mathbf{j}\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2$$

$x_1, \ldots, x_n$: original points, $\tilde{\mathbf{x}}_\mathbf{1}, \ldots, \tilde{\mathbf{x}}_\mathbf{n}$: compressed points, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection matrix. d: original dimension. m: compressed dimension (analogous to d'), ϵ: embedding error, δ: embedding failure prob.

DISTRIBUTIONAL JL $\implies$ JL

Claim: If we choose $\mathbf{\Pi}$ with i.i.d. $\frac{1}{\sqrt{m}} \cdot \mathcal{N}(0,1)$ entries and $m = O\left(\frac{\log 1/\delta'}{\epsilon^2}\right)$, letting $\tilde{\mathbf{x}}_\mathbf{i} = \mathbf{\Pi} x_i$, for each pair x_i, x_j with probability $\geq 1 - \delta'$ we have:

$$(1-\epsilon)\|x_i - x_j\|_2 \leq \|\tilde{\mathbf{x}}_\mathbf{i} - \tilde{\mathbf{x}}_\mathbf{j}\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2.$$

With what probability are all pairwise distances preserved?

Union bound: With probability $\geq 1 - \binom{n}{2} \cdot \delta'$ all pairwise distances are preserved.

Apply the claim with $\delta' = \delta / \binom{n}{2}$. $\implies$ for $m = O\left(\frac{\log 1/\delta'}{\epsilon^2}\right)$, all pairwise distances are preserved with probability $\geq 1 - \delta$.

$$m = O\left(\frac{\log(1/\delta')}{\epsilon^2}\right) = O\left(\frac{\log(\binom{n}{2}/\delta)}{\epsilon^2}\right) = O\left(\frac{\log(n^2/\delta)}{\epsilon^2}\right) = O\left(\frac{\log(n/\delta)}{\epsilon^2}\right)$$

$x_1, \ldots, x_n$: original points, $\tilde{\mathbf{x}}_\mathbf{1}, \ldots, \tilde{\mathbf{x}}_\mathbf{n}$: compressed points, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection matrix. d: original dimension. m: compressed dimension (analogous to d'), ϵ: embedding error, δ: embedding failure prob.

Claim: If we choose $\mathbf{\Pi}$ with i.i.d. $\frac{1}{\sqrt{m}} \cdot \mathcal{N}(0,1)$ entries and $m = O\left(\frac{\log 1/\delta'}{\epsilon^2}\right)$, letting $\tilde{\mathbf{x}}_\mathbf{i} = \mathbf{\Pi} x_i$, for each pair x_i, x_j with probability $\geq 1 - \delta'$ we have:

$$(1-\epsilon)\|x_i - x_j\|_2 \leq \|\tilde{\mathbf{x}}_\mathbf{i} - \tilde{\mathbf{x}}_\mathbf{j}\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2.$$

With what probability are all pairwise distances preserved?

Union bound: With probability $\geq 1 - \binom{n}{2} \cdot \delta'$ all pairwise distances are preserved.

Apply the claim with $\delta' = \delta / \binom{n}{2}$. $\implies$ for $m = O\left(\frac{\log 1/\delta'}{\epsilon^2}\right)$, all pairwise distances are preserved with probability $\geq 1 - \delta$.

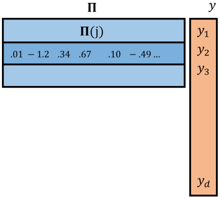
$$m = O\left(\frac{\log(1/\delta')}{\epsilon^2}\right) = O\left(\frac{\log(\binom{n}{2}/\delta)}{\epsilon^2}\right) = O\left(\frac{\log(n^2/\delta)}{\epsilon^2}\right) = O\left(\frac{\log(n/\delta)}{\epsilon^2}\right)$$

Yields the JL lemma.

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

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

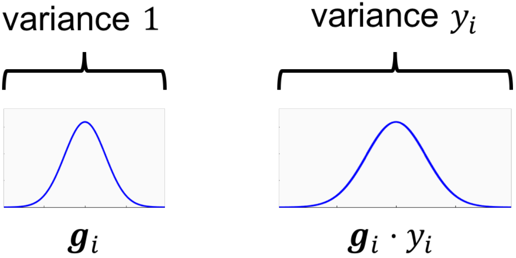
- Let $\tilde{\mathbf{y}}$ denote $\mathbf{\Pi} y$ and let $\mathbf{\Pi}(j)$ denote the j^{th} row of $\mathbf{\Pi}$.
- For any j, $\tilde{\mathbf{y}}(j) = \langle \mathbf{\Pi}(j), y \rangle = \frac{1}{\sqrt{m}} \sum_{i=1}^{d} \mathbf{g}_i \cdot y_i$ where $\mathbf{g}_i \sim \mathcal{N}(0,1)$.



$y \in \mathbb{R}^d$: 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.

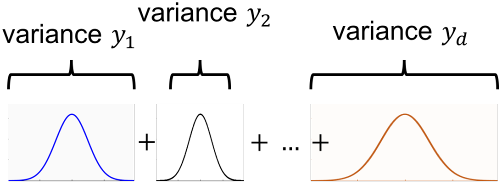
DISTRIBUTIONAL JL PROOF

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



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

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



$$\tilde{\boldsymbol{y}}(j) = \frac{1}{\sqrt{m}}(\boldsymbol{g}_1 \cdot y_1 + \boldsymbol{g}_2 \cdot y_2 + \ldots + \boldsymbol{g}_n \cdot y_d)$$

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

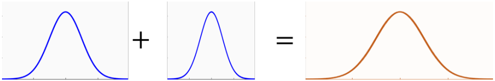
> $y \in \mathbb{R}^d$: arbitrary vector, $\tilde{\mathbf{y}} \in \mathbb{R}^m$: compressed vector, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection mapping $y \to \tilde{\mathbf{y}}$. $\mathbf{\Pi}(j)$: j^{th} row of $\mathbf{\Pi}$, d: original dimension. m: compressed dimension, $\mathbf{g}_i$: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

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

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

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:

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



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

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

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

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

Stability is another explanation for the central limit theorem.

$y \in \mathbb{R}^d$: arbitrary vector, $\tilde{\mathbf{y}} \in \mathbb{R}^m$: compressed vector, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection mapping $y \to \tilde{\mathbf{y}}$. $\mathbf{\Pi}(j)$: 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 $\frac{1}{\sqrt{m}} \cdot \mathcal{N}(0,1)$, for any $y \in \mathbb{R}^d$, letting $\tilde{\mathbf{y}} = \mathbf{\Pi} y$:

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

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

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

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

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

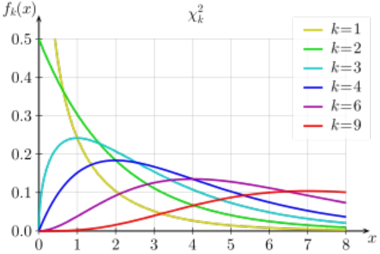
How is $\|\tilde{\mathbf{y}}\|_2^2$ distributed? Does it concentrate?

$y \in \mathbb{R}^d$: arbitrary vector, $\tilde{\mathbf{y}} \in \mathbb{R}^m$: compressed vector, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection mapping $y \to \tilde{\mathbf{y}}$. $\mathbf{\Pi}(j)$: 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 $\frac{1}{\sqrt{m}} \cdot \mathcal{N}(0,1)$, for any $y \in \mathbb{R}^d$, letting $\tilde{\mathbf{y}} = \mathbf{\Pi} y$:

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

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



$y \in \mathbb{R}^d$: arbitrary vector, $\tilde{\mathbf{y}} \in \mathbb{R}^m$: compressed vector, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection mapping $y \to \tilde{\mathbf{y}}$. $\mathbf{\Pi}(j)$: j^{th} row of $\mathbf{\Pi}$, d: original dimension. m: compressed dimension, ϵ: embedding error, δ: embedding failure prob.

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

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

$\|\tilde{\mathbf{y}}\|_2^2 = \sum_{i=1}^m \tilde{\mathbf{y}}(j)^2$ 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,

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

If we set $m = O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$, with probability $1 - O(e^{-\log(1/\delta)}) \geq 1 - \delta$:

$$(1-\epsilon)\|y\|_2^2 \leq \|\tilde{\mathbf{y}}\|_2^2 \leq (1+\epsilon)\|y\|_2^2.$$

$y \in \mathbb{R}^d$: arbitrary vector, $\tilde{\mathbf{y}} \in \mathbb{R}^m$: compressed vector, $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$: random projection mapping $y \to \tilde{\mathbf{y}}$. $\mathbf{\Pi}(j)$: j^{th} row of $\mathbf{\Pi}$, d: original dimension. m: compressed dimension, ϵ: embedding error, δ: embedding failure prob.

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

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

$\|\tilde{\mathbf{y}}\|_2^2 = \sum_{i=1}^m \tilde{\mathbf{y}}(j)^2$ 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,

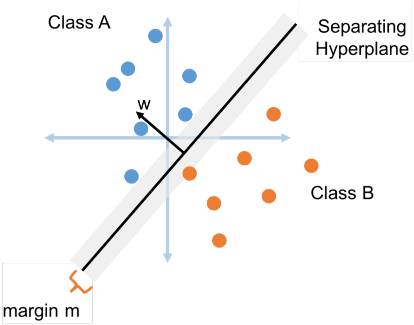
$$\Pr\left[|\mathbf{Z} - \mathbb{E}\mathbf{Z}| \geq \epsilon \mathbb{E}\mathbf{Z}\right] \leq 2e^{-m\epsilon^2/8}.$$

If we set $m = O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$, with probability $1 - O(e^{-\log(1/\delta)}) \geq 1 - \delta$:

$$(1-\epsilon)\|y\|_2^2 \leq \|\tilde{\mathbf{y}}\|_2^2 \leq (1+\epsilon)\|y\|_2^2.$$

Gives the distributional JL Lemma and thus the classic JL Lemma!

Support Vector Machines: A classic ML algorithm, where data is classified with a hyperplane.



- For any point a in A, $\langle a, w \rangle \geq c + m$
- For any point b in B $\langle b, w \rangle \leq c - m$.
- Assume all vectors have unit norm.

JL Lemma implies that after projection into $O\left(\frac{\log n}{m^2}\right)$ dimensions, still have $\langle \tilde{\mathbf{a}}, \tilde{\mathbf{w}} \rangle \geq c + m/4$ and $\langle \tilde{\mathbf{b}}, \tilde{\mathbf{w}} \rangle \leq c - m/4$.

Upshot: Can random project and run SVM (much more efficiently) in the lower dimensional space to find separator $\tilde{\mathbf{w}}$.

Questions?