# CS-GY 9223 D: Lecture 3 Supplemental
# The Johnson-Lindenstrauss Lemma

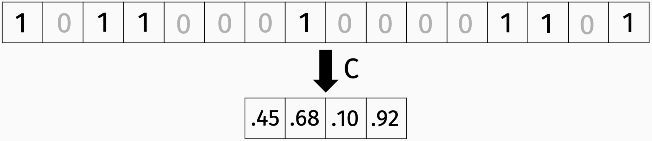
NYU Tandon School of Engineering, Prof. Christopher Musco

Abstract architecture of a sketching algorithm:

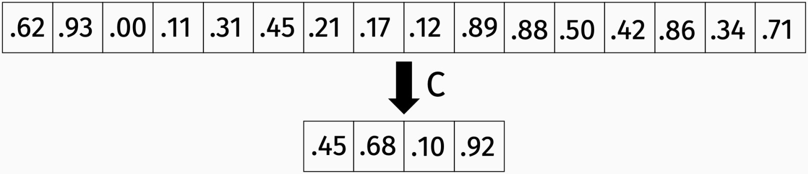
- Given a dataset $D = d_1, \ldots, d_n$ with $n$ pieces of data, we want to output $f(D)$ for some function $f$.
- **Sketch phase:** For each $i \in 1, \ldots, n$, compute $s_i = C(d_i)$, where $C$ is some compression function and $|s_i| \ll d_i$.
- **Process phase:** Using (lower dimensional) dataset $s_1, \ldots, s_n$, compute an approximation to $f(D)$.

Better space complexity, communication complexity, runtime, all at once.

We already saw a powerful application of sketching (the MinHash algorithm) to compressing binary vectors.




Let us estimate the Jaccard similarity between any two binary vectors $\mathbf{q}$ and $\mathbf{y}$ using the information in $C(\mathbf{q})$ and $C(\mathbf{y})$ alone.




Euclidean norm / distance:

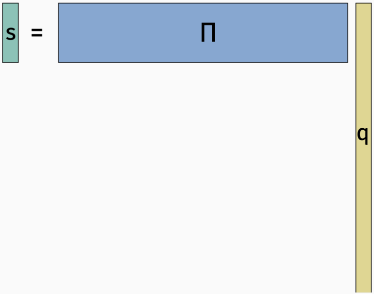
- Given $\mathbf{q} \in \mathbb{R}^d$, $\|\mathbf{q}\|_2 = \sqrt{\sum_{i=1}^{d} q(i)^2}$.
- Given $\mathbf{q}, \mathbf{y} \in \mathbb{R}^d$, distance defined as $\|\mathbf{q} - \mathbf{y}\|_2$.

Can we find compact sketches that preserve Euclidean distance, just as we did for Jaccard similarity?

**Lemma (Johnson-Lindenstrauss, 1984)**

*For any set of n data points* $\mathbf{q}_1, \ldots, \mathbf{q}_n \in \mathbb{R}^d$ *there exists a* *linear map* $\mathbf{\Pi} : \mathbb{R}^d \to \mathbb{R}^k$ *where* $k = O\left(\frac{\log n}{\epsilon^2}\right)$ *such that* *for all* $i, j$,

$$(1-\epsilon)\|\mathbf{q}_i - \mathbf{q}_j\|_2 \leq \|\mathbf{\Pi}\mathbf{q}_i - \mathbf{\Pi}\mathbf{q}_j\|_2 \leq (1+\epsilon)\|\mathbf{q}_i - \mathbf{q}_j\|_2.$$



**Please remember:** This is equivalent to:

**Lemma (Johnson-Lindenstrauss, 1984)**

*For any set of $n$ data points $\mathbf{q}_1, \ldots, \mathbf{q}_n \in \mathbb{R}^d$ there exists a linear map $\mathbf{\Pi} : \mathbb{R}^d \to \mathbb{R}^k$ where $k = O\left(\frac{\log n}{\epsilon^2}\right)$ such that for all $i, j$,*

$$(1-\epsilon)\|\mathbf{q}_i - \mathbf{q}_j\|_2^2 \leq \|\mathbf{\Pi}\mathbf{q}_i - \mathbf{\Pi}\mathbf{q}_j\|_2^2 \leq (1+\epsilon)\|\mathbf{q}_i - \mathbf{q}_j\|_2^2.$$

because for small $\epsilon$, $(1+\epsilon)^2 = 1 + O(\epsilon)$ and $(1-\epsilon)^2 = 1 - O(\epsilon)$.

And this is equivalent to:

**Lemma (Johnson-Lindenstrauss, 1984)**

*For any set of n data points $\mathbf{q}_1, \ldots, \mathbf{q}_n \in \mathbb{R}^d$ there exists a linear map $\mathbf{\Pi} : \mathbb{R}^d \to \mathbb{R}^k$ where $k = O\left(\frac{\log n}{\epsilon^2}\right)$ such that for all $i, j$,*

$$(1-\epsilon)\|\mathbf{\Pi}\mathbf{q}_i - \mathbf{\Pi}\mathbf{q}_j\|_2^2 \leq \|\mathbf{q}_i - \mathbf{q}_j\|_2^2 \leq (1+\epsilon)\|\mathbf{\Pi}\mathbf{q}_i - \mathbf{\Pi}\mathbf{q}_j\|_2^2.$$

because for small $\epsilon$, $\frac{1}{1+\epsilon} = 1 - O(\epsilon)$ and $\frac{1}{1-\epsilon} = 1 + O(\epsilon)$.

Remarkably, $\Pi$ can be chosen completely at random!

**One possible construction:** Random Gaussian.

$$\mathbf{\Pi}_{i,j} = \frac{1}{\sqrt{k}}\mathcal{N}(0,1)$$

The map $\mathbf{\Pi}$ is **oblivious to the data set**. This stands in contrast to e.g. PCA, amoung other differences.

[Indyk, Motwani 1998] [Arriage, Vempala 1999] [Achlioptas 2001] [Dasgupta, Gupta 2003].

Many other possible choices suffice – you can use random $\{+1, -1\}$ variables, sparse random matrices, pseudorandom $\mathbf{\Pi}$. Each with different advantages.

Let $\mathbf{\Pi} \in \mathbb{R}^{k\times d}$ be chosen so that each entry equals $\frac{1}{\sqrt{k}}\mathcal{N}(0,1)$.

... or each entry equals $\frac{1}{\sqrt{k}} \pm 1$ with equal probability.

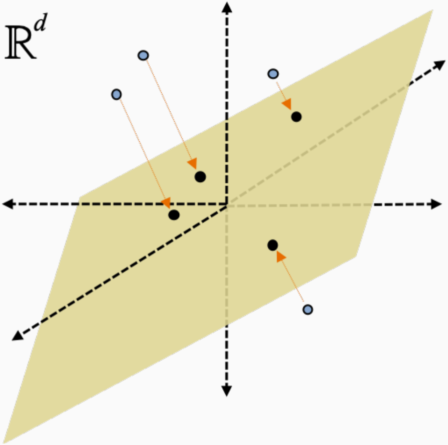
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| -2.1384 | 2.9080 | -0.3538 | 0.0229 | 0.5201 | -0.2938 | -1.3320 | -1.3617 | -0.1952 |
| -0.8396 | 0.8252 | -0.8236 | -0.2620 | -0.0200 | -0.8479 | -2.3299 | 0.4550 | -0.2176 |
| 1.3546 | 1.3790 | -1.5771 | -1.7502 | -0.0348 | -1.1201 | -1.4491 | -0.8487 | -0.3031 |
| -1.0722 | -1.0582 | 0.5080 | -0.2857 | -0.7982 | 2.5260 | 0.3335 | -0.3349 | 0.0230 |
| 0.9610 | -0.4686 | 0.2820 | -0.8314 | 1.0187 | 1.6555 | 0.3914 | 0.5528 | 0.0513 |
| 0.1240 | -0.2725 | 0.0335 | -0.9792 | -0.1332 | 0.3075 | 0.4517 | 1.0391 | 0.8261 |
| 1.4367 | 1.0984 | -1.3337 | -1.1564 | -0.7145 | -1.2571 | -0.1303 | -1.1176 | 1.5270 |
| -1.9609 | -0.2779 | 1.1275 | -0.5336 | 1.3514 | -0.8655 | 0.1837 | 1.2607 | 0.4669 |
| -0.1977 | 0.7015 | 0.3502 | -2.0026 | -0.2248 | -0.1765 | -0.4762 | 0.6601 | -0.2097 |
| -1.2078 | -2.0518 | -0.2991 | 0.9642 | -0.5890 | 0.7914 | 0.8620 | -0.0679 | 0.6252 |

```
>> Pi = randn(m,d);
>> s = (1/sqrt(m))*Pi*q;
```

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | 1 | -1 |
| 1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 |
| 1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | 1 | -1 | 1 | -1 | 1 | -1 | 1 | -1 |
| -1 | -1 | -1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 |
| 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | -1 | 1 |
| 1 | -1 | -1 | 1 | -1 | 1 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 |
| 1 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | 1 | -1 | 1 | -1 | 1 | 1 | 1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 | -1 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | 1 |
| -1 | -1 | 1 | 1 | 1 | 1 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | -1 | 1 |
| -1 | 1 | -1 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 | 1 | -1 | -1 | 1 | -1 | 1 |

```
>> Pi = 2*randi(2,m,d)-3;
>> s = (1/sqrt(m))*Pi*q;
```

A random orthogonal matrix also works. I.e. with $\mathbf{\Pi}\mathbf{\Pi}^T = \mathbf{I}_{k\times k}$. For this reason, the JL operation is often called a "random projection", even though it technically isn't a projection when entries are i.i.d.




Intuitively, close points will remain close after projection, and far points will remain far.

Intermediate result:

**Lemma (Distributional JL Lemma)**

*Let* $\mathbf{\Pi} \in \mathbb{R}^{k \times d}$ *be chosen so that each entry equals* $\frac{1}{\sqrt{k}}\mathcal{N}(0,1)$, *where* $\mathcal{N}(0,1)$ *denotes a standard Gaussian random variable.*

*If we choose* $k = O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$, *then for any vector* $\mathbf{x}$, *with probability* $(1-\delta)$:

$$(1-\epsilon)\|\mathbf{x}\|_2^2 \leq \|\mathbf{\Pi}\mathbf{x}\|_2^2 \leq (1+\epsilon)\|\mathbf{x}\|_2^2$$

Given this lemma, how do we prove the traditional Johnson-Lindenstrauss lemma?

We have a set of vectors $\mathbf{q}_1, \ldots, \mathbf{q}_n$. Fix $i, j \in 1, \ldots, n$.

Let $\mathbf{x} = \mathbf{q}_i - \mathbf{q}_j$. By linearity, $\mathbf{\Pi x} = \mathbf{\Pi}(\mathbf{q}_i - \mathbf{q}_j) = \mathbf{\Pi q}_i - \mathbf{\Pi q}_j$.

By the Distributional JL Lemma, with probability $1 - \delta$,

$$(1-\epsilon)\|\mathbf{q}_i - \mathbf{q}_j\|_2 \leq \|\mathbf{\Pi q}_i - \mathbf{\Pi q}_j\|_2 \leq (1+\epsilon)\|\mathbf{q}_i - \mathbf{q}_j\|_2.$$

Finally, set $\delta = \frac{1}{n^2}$. Since there are $< n^2$ total $i, j$ pairs, by a union bound we have that with probability 9/10, the above will hold for all $i, j$, as long as we compress to:

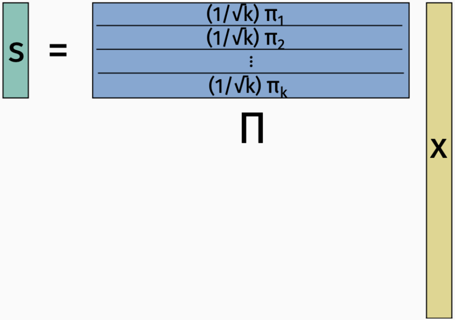
$$k = O\left(\frac{\log(1/(1/n^2))}{\epsilon^2}\right) = O\left(\frac{\log n}{\epsilon^2}\right) \text{ dimensions.} \quad \square$$

Want to argue that, with probability $(1-\delta)$,

$$(1-\epsilon)\|\mathbf{x}\|_2^2 \leq |\mathbf{\Pi x}\|_2^2 \leq (1+\epsilon)\|\mathbf{x}\|_2^2$$

**Claim**: $\mathbb{E}\|\mathbf{\Pi x}\|_2^2 = \|\mathbf{x}\|_2^2$.

Some notation:




So each $\boldsymbol{\pi}_i$ contains $\mathcal{N}(0,1)$ entries.

$$\|\boldsymbol{\Pi}\mathbf{x}\|_2^2 = \sum_i^k \mathbf{s}(i)^2 = \sum_i^k \left(\frac{1}{\sqrt{k}}\langle \boldsymbol{\pi}_i, \mathbf{x}\rangle\right)^2 = \frac{1}{k}\sum_i^k (\langle \boldsymbol{\pi}_i, \mathbf{x}\rangle)^2$$

$$\mathbb{E}\left[\|\boldsymbol{\Pi}\mathbf{x}\|_2^2\right] = \frac{1}{k}\sum_i^k \mathbb{E}\left[(\langle \boldsymbol{\pi}_i, \mathbf{x}\rangle)^2\right]$$
$$= \mathbb{E}\left[(\langle \boldsymbol{\pi}_i, \mathbf{x}\rangle)^2\right]$$

**Goal**: Prove $\mathbb{E}\|\boldsymbol{\Pi}\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2$.

$$\langle \boldsymbol{\pi}_i, \mathbf{x} \rangle = Z_1 \cdot \mathbf{x}(1) + Z_2 \cdot \mathbf{x}(2) + \ldots + Z_d \cdot \mathbf{x}(d)$$

where each $Z_1, \ldots, Z_d$ is a standard normal $\mathcal{N}(0, 1)$ random variable.

This implies that $Z_i \cdot \mathbf{x}(i)$ is a normal $\mathcal{N}(0, \mathbf{x}(i)^2)$ random variable.

**Goal**: Prove $\mathbb{E}\|\boldsymbol{\Pi}\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2$. Established: $\mathbb{E}\|\boldsymbol{\Pi}\mathbf{x}\|_2^2 = \mathbb{E}\left[(\langle \boldsymbol{\pi}_i, \mathbf{x} \rangle)^2\right]$

What type of random variable is $\langle \boldsymbol{\pi}_i, \mathbf{x} \rangle$?

**Fact (Stability of Gaussian random variables)**

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

$$\begin{aligned}\langle \boldsymbol{\pi}_i, \mathbf{x} \rangle &= \mathcal{N}(0, \mathbf{x}(1)^2) + \mathcal{N}(0, \mathbf{x}(2)^2) + \ldots + \mathcal{N}(0, \mathbf{x}(d)^2) \\ &= \mathcal{N}(0, \|\mathbf{x}\|_2^2).\end{aligned}$$

So $\mathbb{E}\|\boldsymbol{\Pi}\mathbf{x}\|_2^2 = \mathbb{E}\left[(\langle \boldsymbol{\pi}_i, \mathbf{x} \rangle)^2\right] = \|\mathbf{x}\|_2^2$, as desired.

Want to argue that, with probability $(1-\delta)$,

$$(1-\epsilon)\|\mathbf{x}\|_2^2 \leq \|\mathbf{\Pi x}\|_2^2 \leq (1+\epsilon)\|\mathbf{x}\|_2^2$$

1. $\mathbb{E}\|\mathbf{\Pi x}\|_2^2 = \|\mathbf{x}\|_2^2$.
2. Need to use a concentration bound.

$$\|\mathbf{\Pi x}\|_2^2 = \frac{1}{k}\sum_{i=1}^{k}\left(\langle \boldsymbol{\pi}_i, \mathbf{x}\rangle\right)^2 = \frac{1}{k}\sum_{i=1}^{k}\mathcal{N}(0, \|\mathbf{x}\|_2^2)$$

"Chi-squared random variable with $k$ degrees of freedom."

## Lemma

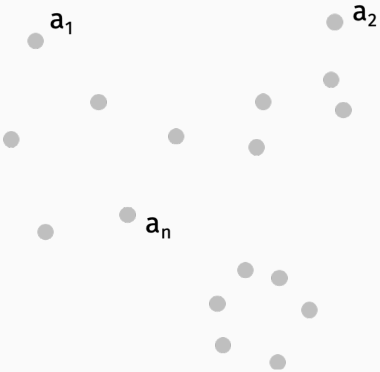
*Let $Z$ be a Chi-squared random variable with $k$ degrees of freedom.*

$$\Pr[|\mathbb{E}Z - Z| \geq \epsilon \mathbb{E}Z] \leq 2e^{-k\epsilon^2/8}$$

**Goal**: Prove $\|\mathbf{\Pi x}\|_2^2$ concentrates within $1 \pm \epsilon$ of its expectation, which equals $\|\mathbf{x}\|_2^2$.

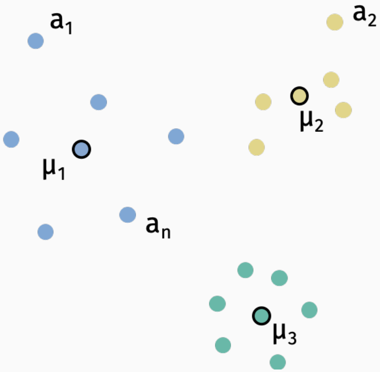
**k-means clustering**: Give data points $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^d$, find centers $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k \in \mathbb{R}^d$ to minimize:

$$Cost(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k) = \sum_{i=1}^{n} \min_{j=1,\ldots,k} \|\boldsymbol{\mu}_j - \mathbf{x}_i\|_2^2$$



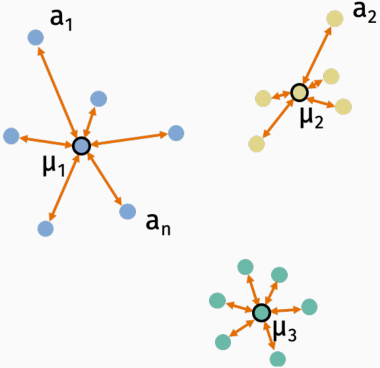
**k-means clustering**: Give data points $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^d$, find centers $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k \in \mathbb{R}^d$ to minimize:

$$Cost(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k) = \sum_{i=1}^{n} \min_{j=1,\ldots,k} \|\boldsymbol{\mu}_j - \mathbf{X}_i\|_2^2$$



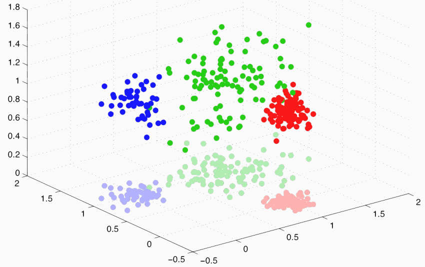
**k-means clustering**: Give data points $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^d$, find centers $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k \in \mathbb{R}^d$ to minimize:

$$Cost(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k) = \sum_{i=1}^{n} \min_{j=1,\ldots,k} \|\boldsymbol{\mu}_j - \mathbf{a}_i\|_2^2$$



NP hard to solve exactly, but there are many good approximation algorithms. All depend at least linearly on the dimension $d$.

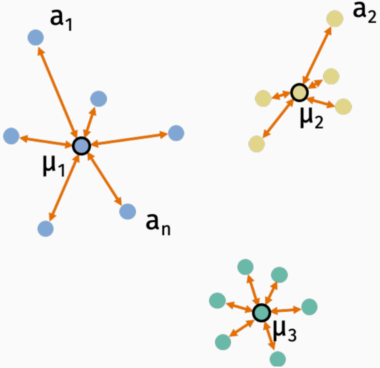
**Approximation scheme**: Find clusters $\tilde{C}_1, \ldots, \tilde{C}_k$ for the $k = O\left(\frac{\log n}{\epsilon^2}\right)$ dimension data set $\mathbf{\Pi a}_1, \ldots, \mathbf{\Pi a}_n$.




Argue these clusters are near optimal for $\mathbf{a}_1, \ldots, \mathbf{a}_n$.

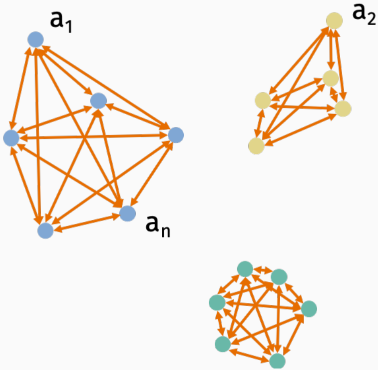
**Equivalent formulation**: Find clusters $C_1, \ldots, C_k \subseteq \{1, \ldots, n\}$ to minimize:

$$Cost(C_1, \ldots, C_k) = \sum_{j=1}^{k} \frac{1}{2|C_j|} \sum_{u,v \in C_j} \|\mathbf{a}_u - \mathbf{a}_v\|_2^2.$$



**Equivalent formulation**: Find clusters $C_1, \ldots, C_k \subseteq \{1, \ldots, n\}$ to minimize:

$$Cost(C_1, \ldots, C_k) = \sum_{j=1}^{k} \frac{1}{2|C_j|} \sum_{u,v \in C_j} \|\mathbf{a}_u - \mathbf{a}_v\|_2^2.$$



$$Cost(C_1, \ldots, C_k) = \sum_{j=1}^{k} \frac{1}{2|C_j|} \sum_{u,v \in C_j} \|\mathbf{a}_u - \mathbf{a}_v\|_2^2$$

$$\widetilde{Cost}(C_1, \ldots, C_k) = \sum_{j=1}^{k} \frac{1}{2|C_j|} \sum_{u,v \in C_j} \|\Pi \mathbf{a}_u - \Pi \mathbf{a}_v\|_2^2$$

Let $Cost^* = \min Cost(C_1, \ldots, C_k)$ and $\widetilde{Cost}^* = \min \widetilde{Cost}(C_1, \ldots, C_k)$.

**Claim:** $(1-\epsilon)Cost^* \leq \widetilde{Cost}^* \leq (1+\epsilon)Cost^*$.

Suppose we use an approximation algorithm to find clusters $B_1, \ldots, B_k$ such that:

$$\widetilde{Cost}(B_1, \ldots, B_k) \leq (1+\alpha)\widetilde{Cost}^*$$

Then:

$$\begin{aligned} Cost(B_1, \ldots, B_k) &\leq \frac{1}{1-\epsilon}\widetilde{Cost}(B_1, \ldots, B_k) \\ &\leq (1+\alpha)(1+O(\epsilon))\widetilde{Cost}^* \\ &\leq (1+\alpha)(1+O(\epsilon))(1+\epsilon)Cost^* \\ &= 1+O(\alpha+\epsilon)Cost^* \end{aligned}$$

If high dimensional geometry is so different from low-dimensional geometry, why is dimensionality reduction possible? Doesn't Johnson-Lindenstrauss tell us that high-dimensional geometry can be approximated in low dimensions?

**Hard case:** $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ are all mutually orthogonal unit vectors:

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 = 2 \qquad \text{for all } i, j.$$

From our result earlier, in $O(\log n/\epsilon^2)$ dimensions, there exists $2^{O(\epsilon^2 \cdot \log n/\epsilon^2)} \geq n$ unit vectors that are close to mutually orthogonal.

$O(\log n/\epsilon^2)$ = just enough dimensions.