CS-GY 6763: Lecture 3
High Dimensional Geometry, the Johnson-Lindenstrauss Lemma, MinHash

NYU Tandon School of Engineering, Prof. Christopher Musco

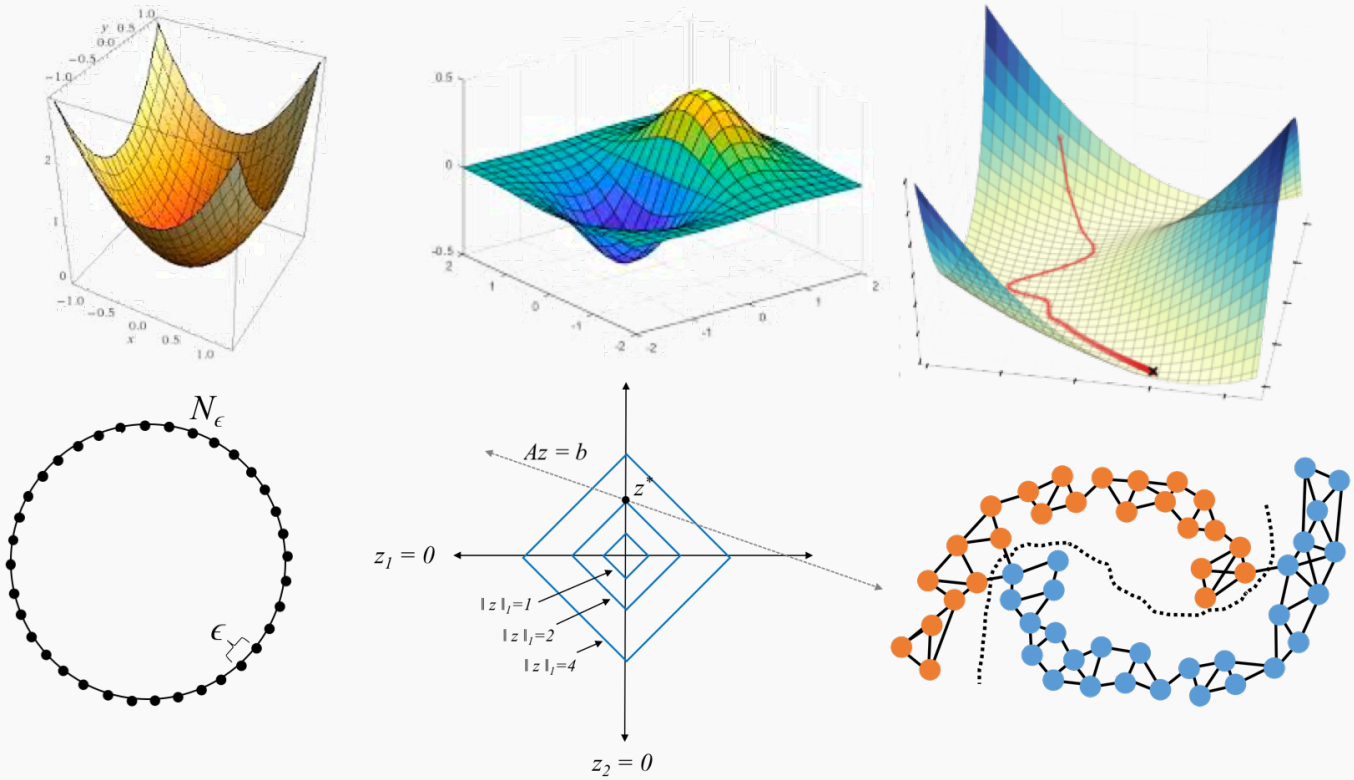
UNIFYING THEME OF THE COURSE

How do we deal with data (vectors) in high dimensions?

- Locality sensitive hashing for similarity search.
- Iterative methods for optimizing functions that depend on many variables.
- SVD + low-rank approximation to find and visualize low-dimensional structure.
- Convert large graphs to high dimensional vector data.

HIGH DIMENSIONAL IS NOT LIKE LOW DIMENSIONAL

Often visualize data and algorithms in 1,2, or 3 dimensions.

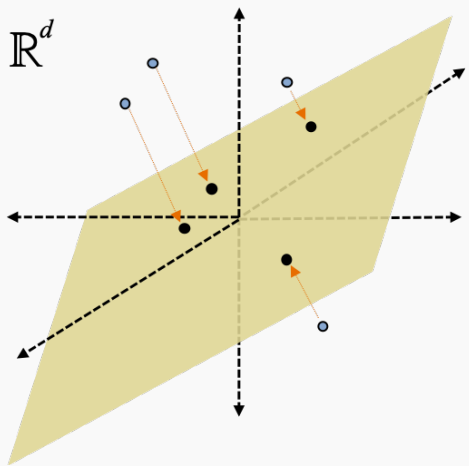


First part of lecture: Prove that high-dimensional space looks very different from low-dimensional space. These images are rarely very informative!

Second part of lecture: Ignore our own advice.

Learn about sketching, aka dimensionality reduction techniques that seek to approximate high-dimensional vectors with much lower dimensional vectors.

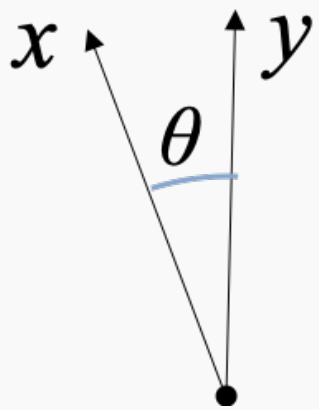
- Johnson-Lindenstrauss lemma for ℓ_2 space.
- MinHash for binary vectors.

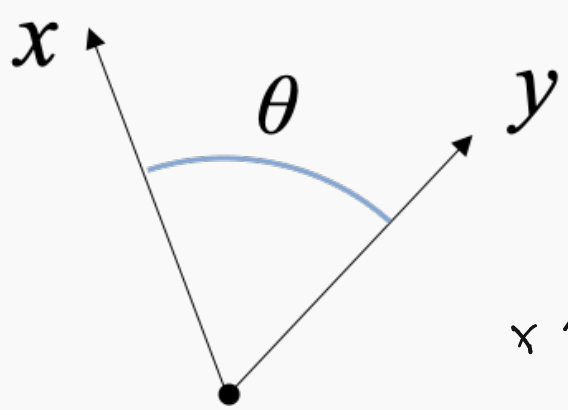


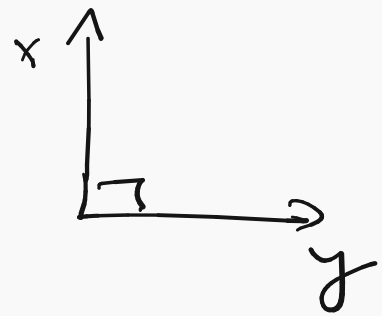
First part of lecture should help you understand the potential and limitations of these methods.

Recall the inner product between two d dimensional vectors:

$$\langle x, y \rangle = x^T y = y^T x = \sum_{i=1}^{d} x_i y_i$$





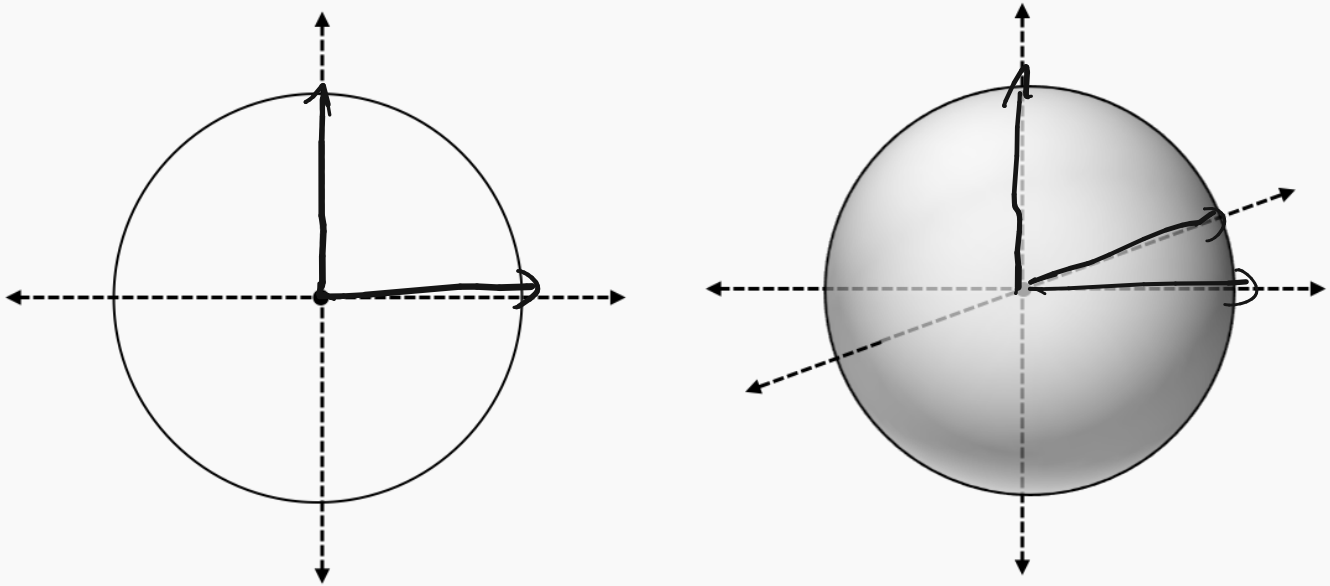


$$\langle x, y \rangle = \cos(\theta) \cdot \|x\|_2 \cdot \|y\|_2$$

ORTHOGONAL VECTORS

What is the largest set of **mutually orthogonal** unit vectors $\mathbf{x}_1, \ldots, \mathbf{x}_t$ in d-dimensional space? I.e. with inner product $|\mathbf{x}_i^T \mathbf{x}_j| = 0$ for all i, j.

$= d$

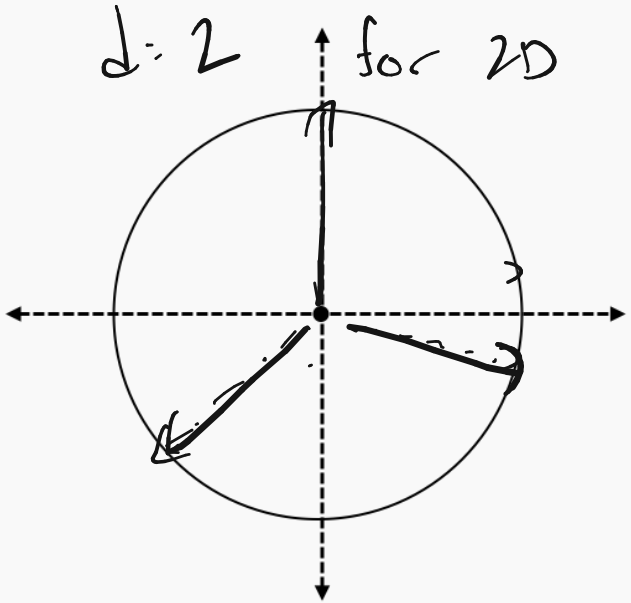


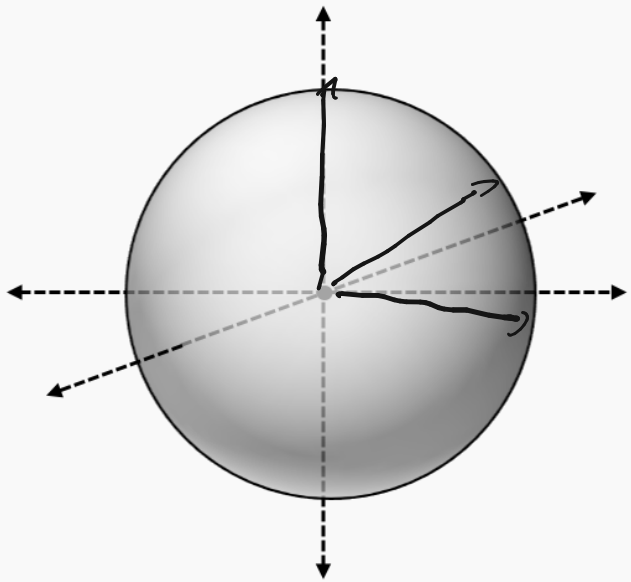
ORTHOGONAL VECTORS

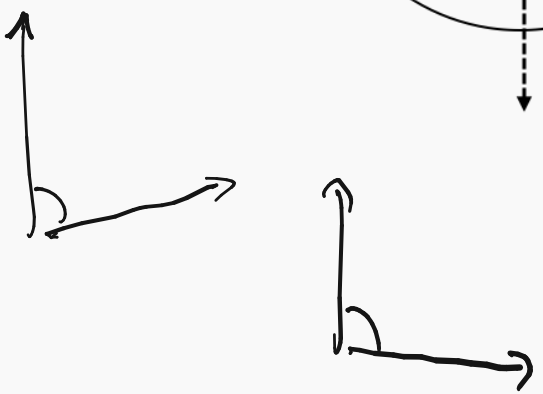
$\epsilon = .1$

$(1+\epsilon)2 = 2.2$

What is the largest set **nearly orthogonal** unit vectors $\mathbf{x}_1, \ldots, \mathbf{x}_t$ in d-dimensional space. I.e., with inner product $|\mathbf{x}_i^T \mathbf{x}_j| \leq \epsilon$ for all i, j.







ORTHOGONAL VECTORS

What is the largest set **nearly orthogonal** unit vectors $\mathbf{x}_1, \ldots, \mathbf{x}_t$ in d-dimensional space. I.e., with inner product $|\mathbf{x}_i^T \mathbf{x}_j| \leq \epsilon$ for all i, j.

1. d
2. $\Theta(d)$
3. $\Theta(d^2)$
4. $2^{\Theta(d)}$

ORTHOGONAL VECTORS

Claim: There is an exponential number (i.e., $\sim 2^d$) of nearly orthogonal unit vectors in d dimensional space.

Proof strategy: Use the Probabilistic Method! For $t = 2^{o(d)}$ define a random process which generates random vectors $\mathbf{x}_1, \ldots, \mathbf{x}_t$ that are unlikely to have large inner product.

1. Claim that, with non-zero probability, $|\mathbf{x}_i^T \mathbf{x}_j| \leq \epsilon$ for all i, j.
2. Conclude that there must exists some set of t unit vectors with all pairwise inner-products bounded by ϵ.

Claim: There is an exponential number (i.e., $\sim 2^d$) of nearly orthogonal unit vectors in d dimensional space.

Proof: Let $\mathbf{x}_1, \ldots, \mathbf{x}_t$ all have independent random entries, each set to $\pm\frac{1}{\sqrt{d}}$ with equal probability.

- $\|\mathbf{x}_i\|_2 = \left(\sum_{h=1}^{d} x_i(h)^2\right)^{1/2} = \left(\sum_{i=1}^{d} 1/d\right)^{1/2} = 1$
- $\mathbb{E}[\mathbf{x}_i^T \mathbf{x}_j] = \mathbb{E}\left[\sum_{h=1}^{d} x_i(h) \cdot x_j(h)\right] = \sum_{h=1}^{d} \mathbb{E}[x_i(h)\, x_j(h)] = 0$ (each $\mathbb{E}[x_i(h)\, x_j(h)] = 0$)
- $\mathrm{Var}[\mathbf{x}_i^T \mathbf{x}_j] = \sum_{h=1}^{d} \mathrm{Var}[x_i(h) \cdot x_j(h)] = 1/d$

pairwise independence

$x_i(h)$: $\begin{cases} 1/d & \text{w/ prob } 1/2 \\ -1/d & \text{w/ prob } 1/2 \end{cases}$

$x_j(h)$: $1/d^2$

Let $Z = \mathbf{x}_i^T \mathbf{x}_j = \sum_{i=1}^{d} C_i$ where each C_i is $+\frac{1}{d}$ or $-\frac{1}{d}$ with equal probability.

Z is a sum of many i.i.d. random variables, so looks approximately Gaussian. Roughly, we expect that:

$$\Pr[|Z - \mathbb{E}Z| \geq \alpha \cdot \sigma] \leq O(e^{-\alpha^2})$$

$\geq \varepsilon$; $\alpha = \varepsilon\sqrt{d}$; $\sigma = 1/\sqrt{d}$; $e^{-\varepsilon^2 d}$

Note that we can transform to binary random variable:

$$Z = \sum_{i=1}^{d} C_i = \frac{2}{d} \sum_{i=1}^{d} \frac{d}{2} \cdot C_i$$

$$= \frac{2}{d} \cdot \left(-\frac{d}{2} + \sum_{i=1}^{d} B_i \right)$$

where each B_i is uniform in $\{0, 1\}$.

Theorem (Chernoff Bound)

Let $X_1, X_2, \ldots, X_k$ be independent $\{0,1\}$-valued random variables and let $S = \sum_{i=1}^{k} X_i$. We have for any $\epsilon < 1$:

$$\Pr[|S - \mathbb{E}[S]| \geq \epsilon \mathbb{E}[S]] \leq 2e^{\frac{-\epsilon^2 \mathbb{E}[S]}{3}}.$$

$$\Pr[|B - \mathbb{E}[B]| \geq \qquad\qquad] \leq$$

Formally, using a Chernoff bound: $X_1, \ldots, X_t$

$$\Pr[|Z - \mathbb{E}Z| \geq \epsilon] \leq 2e^{-\epsilon^2 d/6}$$

For any i, j pair, $\Pr[|\mathbf{x}_i^T \mathbf{x}_j| < \epsilon] \geq 1 - 2e^{-\epsilon^2 d/6}$.

By a union bound:

For all i, j pairs simultaneously, $\Pr[|\mathbf{x}_i^T \mathbf{x}_j| < \epsilon] \geq 1 - \binom{t}{2} \cdot 2e^{-\epsilon^2 d/6}$.

N, t^2

$\binom{t}{2} \cdot 2e^{-\epsilon^2 d/6} = 1/2$ $\geq 1/2$

$t^2 = O(e^{\epsilon^2 d/6})$

$t = O(e^{\epsilon^2 d/12})$ $t = e^{O(\epsilon^2 d)}$.

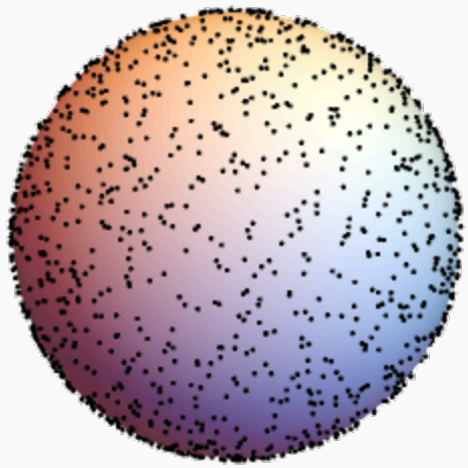
Final result: In d-dimensional space, there are $2^{\theta(\epsilon^2 d)}$ unit vectors with all pairwise inner products $\leq \epsilon$.

Corollary of proof: Random vectors tend to be far apart in high-dimensions.

$\|x - y\|_2^2 = \|x\|_2^2 + \|y\|_2^2 - 2\langle x, y \rangle$ ($\leq \epsilon$)

$\approx 2 \pm \epsilon$

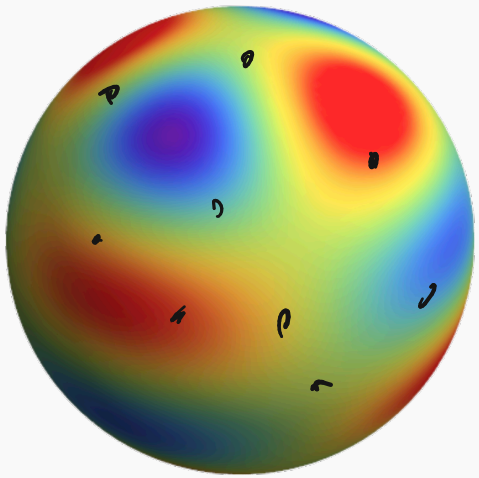
$\|x - y\|_2 \approx \sqrt{2}$



CURSE OF DIMENSIONALITY

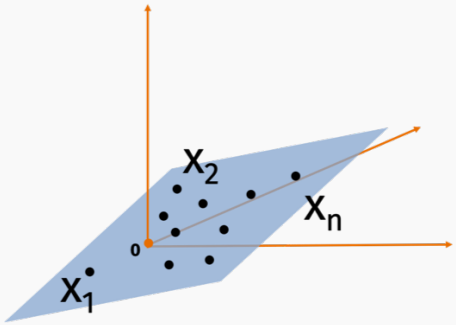
Curse of dimensionality: Suppose we want to use e.g. k-nearest neighbors to learn a function or classify points in $\mathbb{R}^d$. If our data distribution is truly random, we typically need an exponential amount of data.

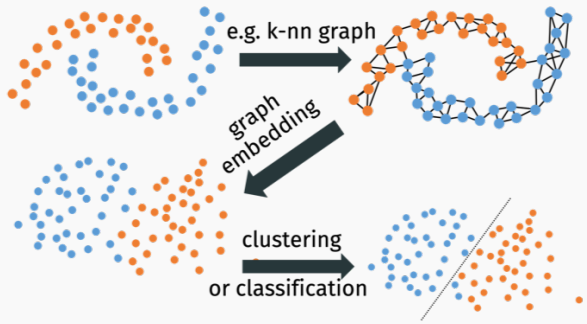
$2^{O(\epsilon^2 d)}$

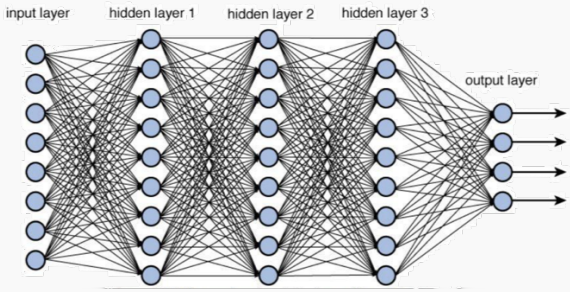


The existence of lower dimensional structure is our data is often the only reason we can hope to learn.

Low-dimensional structure.







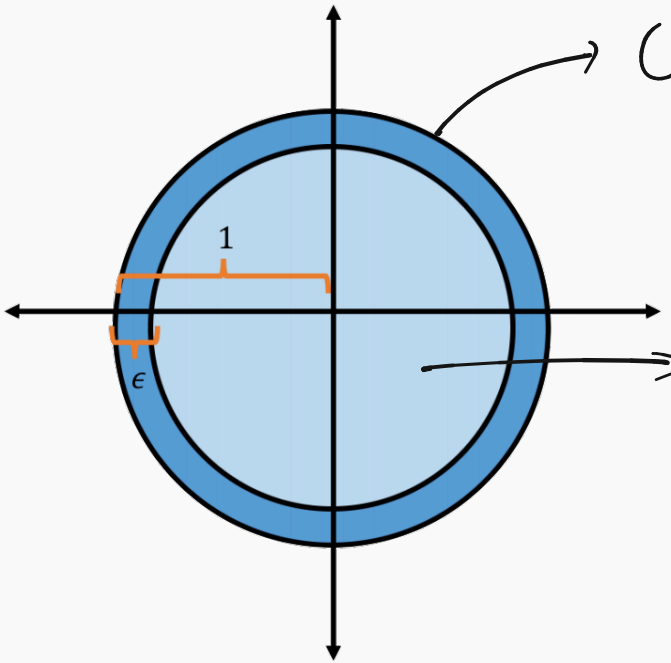
For example, data lies on low-dimensional subspace, or does so after transformation. Or function can be represented by a restricted class of functions, like neural net with specific structure.

UNIT BALL IN HIGH DIMENSIONS

Let $\mathcal{B}_d$ be the unit ball in d dimensions:

$$\mathcal{B}_d = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 \leq 1\}.$$

What percentage of volume of $\mathcal{B}_d$ falls with ϵ of its surface?



$C \cdot 1^d = C \cdot 1$

$C \cdot (1-\epsilon)^d = \left((1-\epsilon)^{1/\epsilon}\right)^{d\epsilon}$

$\approx \left(\frac{1}{e}\right)^{d\epsilon}$

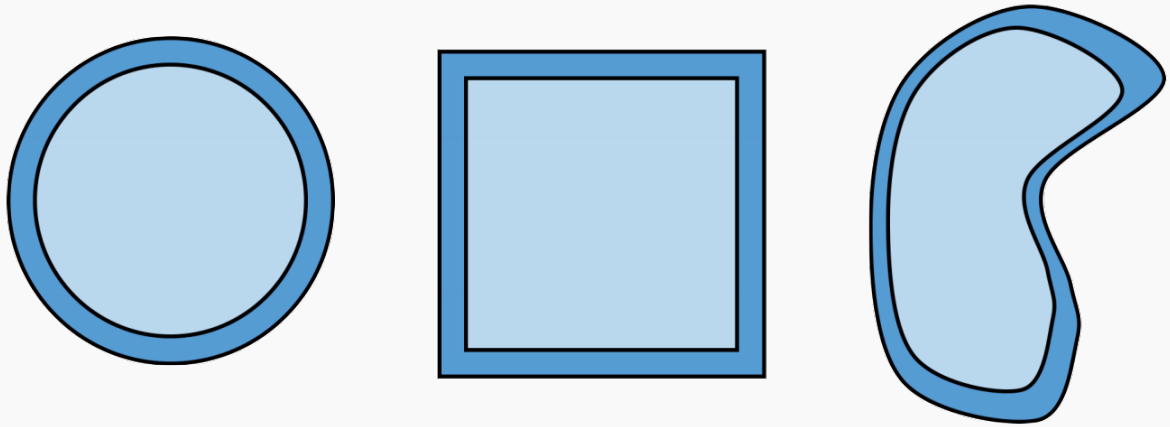
$= \left(\frac{1}{2}\right)^{\Theta(d\epsilon)}$

Volume of radius R ball is $\frac{\pi^{d/2}}{(d/2)!} \cdot R^d$.

ISOPERIMETRIC INEQUALITY

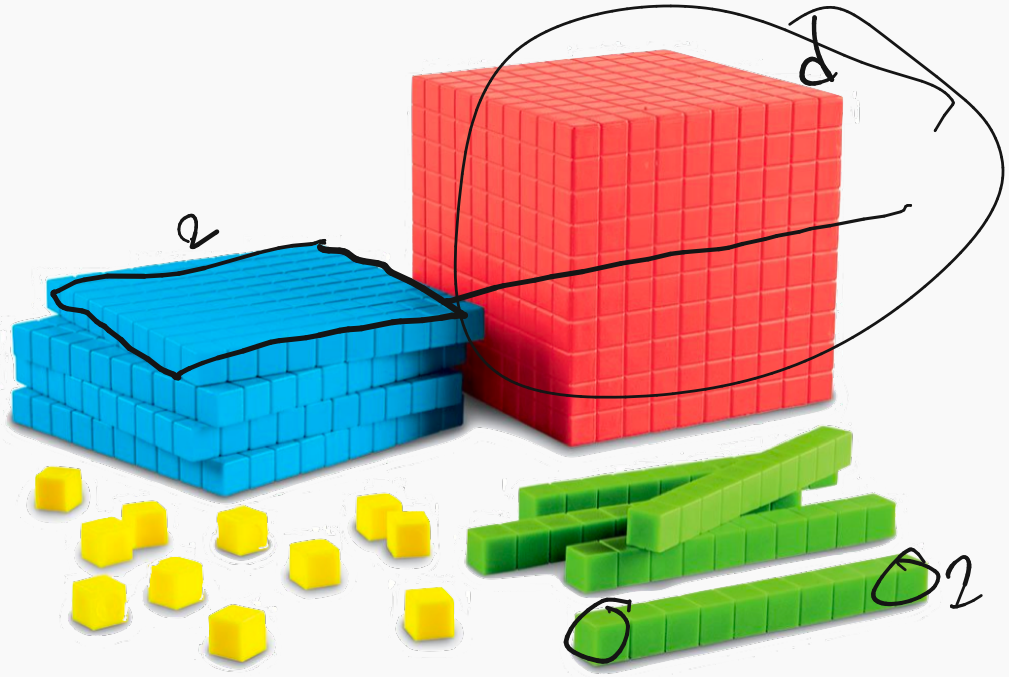
All but an $\frac{1}{2}^{\Theta(\epsilon d)}$ fraction of a unit ball's volume is within ϵ of its surface.

Isoperimetric Inequality: the ball has the maximum surface area/volume ratio of any shape.



- If we randomly sample points from any high-dimensional shape, nearly all will fall near its surface.
- 'All points are outliers.'

INTUITION



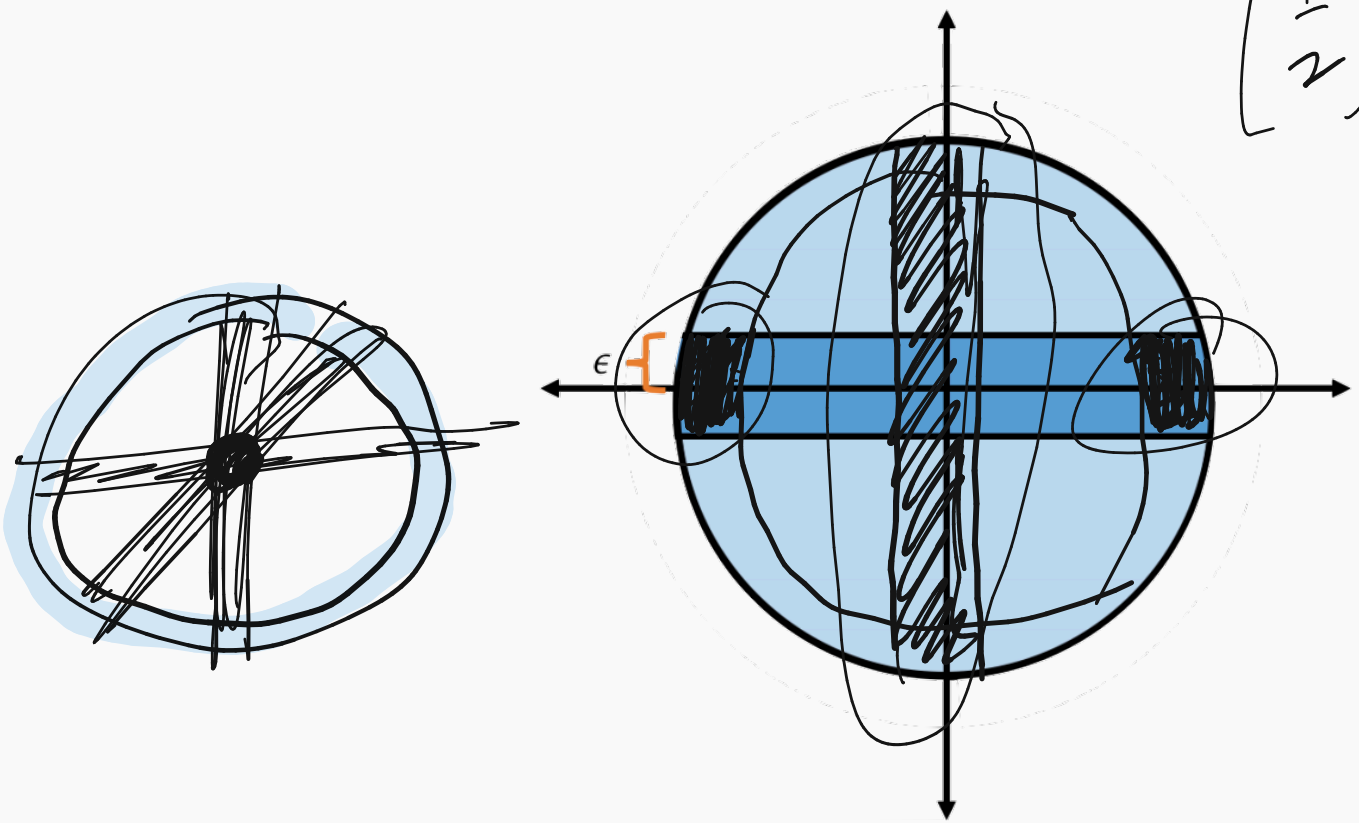
1D: $\frac{\text{surface cubes}}{\text{total cubes}} = \frac{2}{10} = .2$

2D: $\frac{\text{surface cubes}}{\text{total cubes}} = \frac{10^2 - 8^2}{10^2} = .36$

3D: $\frac{\text{surface cubes}}{\text{total cubes}} = \frac{10^3 - 8^3}{10^3} = \frac{1000 - 512}{1000} \approx .5$

SLICES OF THE UNIT BALL

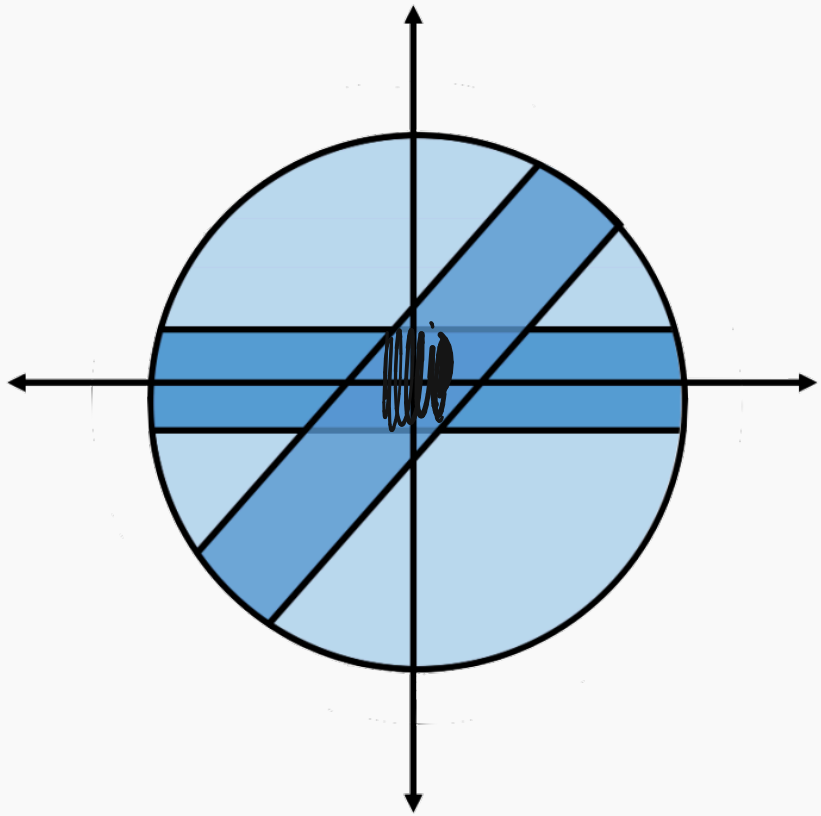
What percentage of the volume of $\mathcal{B}_d$ falls within ϵ of its equator?



$$S = \{\mathbf{x} \in \mathcal{B}_d : |x_1| \leq \epsilon\}$$

SLICES OF THE UNIT BALL

What percentage of the volume of $\mathcal{B}_d$ falls within ϵ of its equator? **Answer**: all but a $\frac{1}{2}^{\Theta(\epsilon^2 d)}$ fraction.

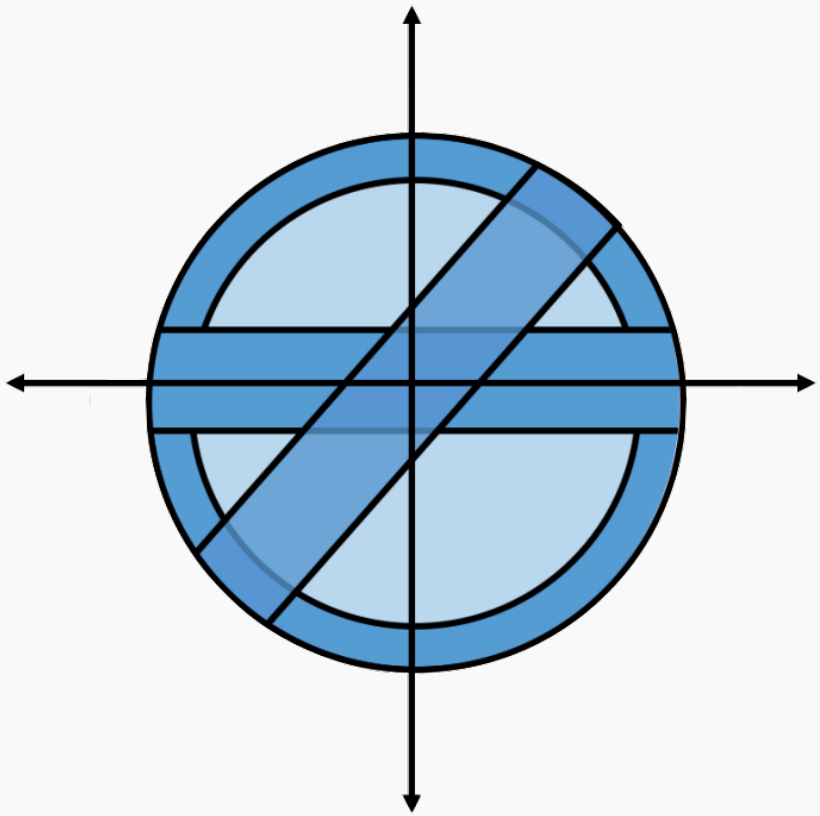


By symmetry, this is true for any equator:

$$S_t = \{x \in \mathcal{B}_d : x^T t \leq \epsilon\}.$$

BIZARRE SHAPE OF UNIT BALL

1. $(1 - \frac{1}{2}^{\Theta(\epsilon d)})$ fraction of volume lies ϵ close to surface.
2. $(1 - \frac{1}{2}^{\Theta(\epsilon^2 d)})$ fraction of volume lies ϵ close to any equator.



High-dimensional ball looks nothing like 2D ball!

CONCENTRATION AT EQUATOR

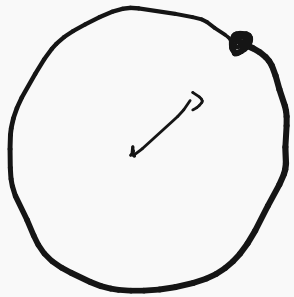
Claim: All but a $\frac{1}{2}^{\Theta(\epsilon^2 d)}$ fraction of the volume of the ball falls within ϵ of its equator.

Equivalent: If we draw a point $\mathbf{x}$ randomly from the unit ball, $|x_1| \leq \epsilon$ with probability $\geq 1 - \frac{1}{2}^{\Theta(\epsilon^2 d)}$.

CONCENTRATION AT EQUATOR

Let $\mathbf{w} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$. Because $\|\mathbf{x}\|_2 \leq 1$,

$$\Pr\left[|x_1| \leq \epsilon\right] \geq \Pr\left[|w_1| \leq \epsilon\right].$$



Claim: $|w_1| \leq \epsilon$ with probability $\geq 1 - \frac{1}{2}^{\Theta(\epsilon^2 d)}$, which then proves our statement from the previous slide.

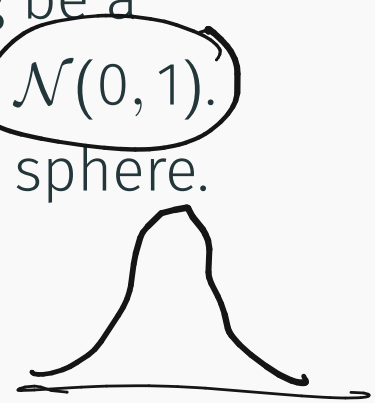
How can we generate $\mathbf{w}$, which is a random vector taken from the unit sphere (the surface of the ball)?

$$\|x\|_2^2 = x_1^2 + x_2^2 + \cdots + x_d^2$$

$$\|x\|_p = \left(\sum x_i^p\right)^{1/p}$$

IMPORTANT FACT IN HIGH DIMENSIONAL GEOMETRY

Rotational Invariance of Gaussian distribution: Let $\mathbf{g}$ be a random Gaussian vector, with each entry drawn from $\mathcal{N}(0,1)$. Then $\mathbf{w} = \mathbf{g}/\|\mathbf{g}\|_2$ is distributed uniformly on the unit sphere.



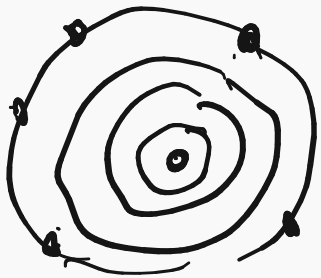
Proof:

$g_1 \sim N(0,1)$ $\quad c \cdot e^{-g_1^2}$

$g_2 \sim N(0,1)$ $\quad c^d e^{-g_1^2} e^{-g_2^2} \cdots e^{-g_d^2}$

$\vdots$ $\quad = c^d e^{-(g_1^2 + \cdots + g_d^2)}$

$g_d \sim N(0,1)$ $\quad = c^d e^{-\|g\|_2^2}$



$g = [\_\_\_\_\_]$

CONCENTRATION AT EQUATOR

Let $\mathbf{g}$ be a random Gaussian vector and $\mathbf{w} = \mathbf{g}/\|\mathbf{g}\|_2$.

every entry is $N(0,1)$

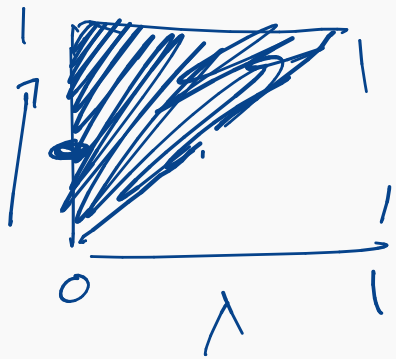
- $\mathbb{E}[\|\mathbf{g}\|_2^2] = \mathbb{E}\left[\sum_{i=1}^{d} g_i^2\right] = \sum_{i=1}^{d} \mathbb{E}[g_i^2] = \sum_{i=1}^{d} \mathrm{Var}[g_i] = d$

- $\Pr\left[\|\mathbf{g}\|_2^2 \le \frac{1}{2}\mathbb{E}[\|\mathbf{g}\|_2^2]\right] \le \frac{1}{2}^{\theta(d)}$

$\le \frac{d}{2}\ ] \le 1/2^{\theta(d)}$

$X : 0 \to 1$

$\lambda : 0 \to X$



$$\int_{\lambda=0}^{1} \Pr[s \ge \lambda]\, d\lambda$$

$$\int_{\lambda=0}^{1} \int_{x=\lambda}^{1} p(s=x)\, dx\, d\lambda$$

$=$

CONCENTRATION AT EQUATOR

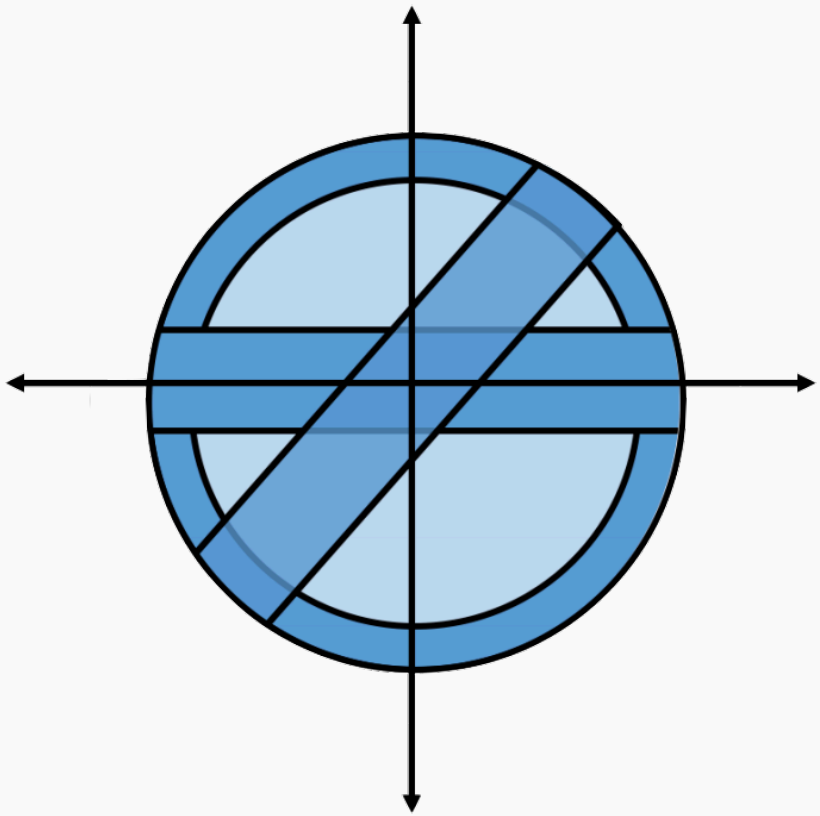
For $1 - \frac{1}{2}^{\theta(d)}$ fraction of vectors $\mathbf{g}$, $\|\mathbf{g}\|_2 \geq \sqrt{d/2}$. Condition on the event that we get a random vector in this set.

$$\begin{aligned}
\Pr\left[|w_1| \leq \epsilon\right] &= \Pr\left[|w_1| \cdot \sqrt{d/2} \leq \epsilon \cdot \sqrt{d/2}\right] \\
&\geq \Pr\left[|g_1| \leq \epsilon \cdot \sqrt{d/2}\right] \\
&\geq 1 - \frac{1}{2}^{\theta\left((\epsilon \cdot \sqrt{d/2})^2\right)}
\end{aligned}$$

Recall: $\mathbf{w} = \frac{\mathbf{g}}{\|\mathbf{g}\|_2}$. So after conditioning, we have $\mathbf{w} \leq \frac{\mathbf{g}}{\sqrt{d/2}}$.

BIZARRE SHAPE OF UNIT BALL

1. $(1 - \frac{1}{2}^{\Theta(\epsilon d)})$ fraction of volume lies ϵ close to surface.
2. $(1 - \frac{1}{2}^{\Theta(\epsilon^2 d)})$ fraction of volume lies ϵ close to any equator.

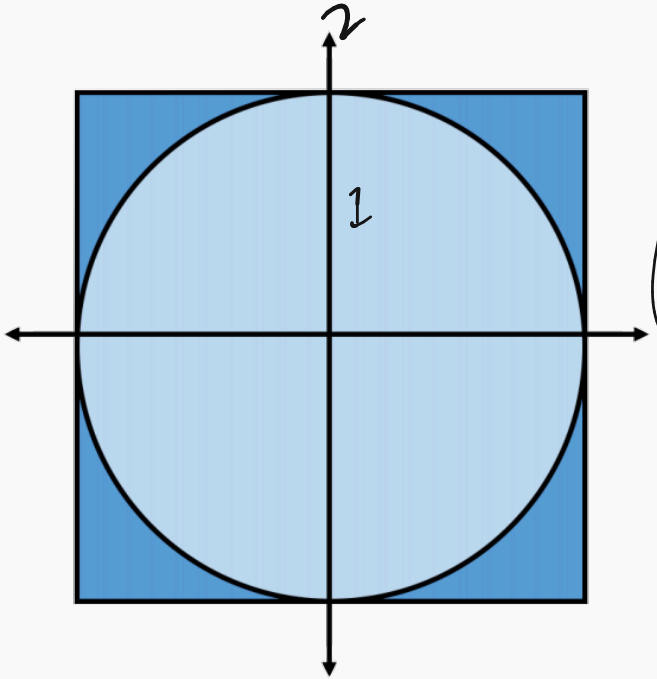


High-dimensional ball looks nothing like 2D ball!

HIGH DIMENSIONAL CUBE

Let $\mathcal{C}_d$ be the d-dimensional cube:

$$\mathcal{C}_d = \{\mathbf{x} \in \mathbb{R}^d : |\mathbf{x}(i)| \leq 1 \ \forall i\}.$$



In two dimensions, the cube is pretty similar to the ball.

But volume of $\mathcal{C}_d$ is 2^d while volume of unit ball is $\frac{\sqrt{\pi}^d}{(d/2)!}$.

This is a huge gap! Cube has $O(d)^{O(d)}$ more volume.

HIGH DIMENSIONAL CUBE

Some other ways to see these shapes are very different:

- $\max_{\mathbf{x} \in \mathcal{B}_d} \|\mathbf{x}\|_2^2 =$ 1
- $\max_{\mathbf{x} \in \mathcal{C}_d} \|\mathbf{x}\|_2^2 =$ d

$\|x\|_\infty = \sqrt{d}$

$x = (1, 1, 1 \ldots, 1)$

HIGH DIMENSIONAL CUBE

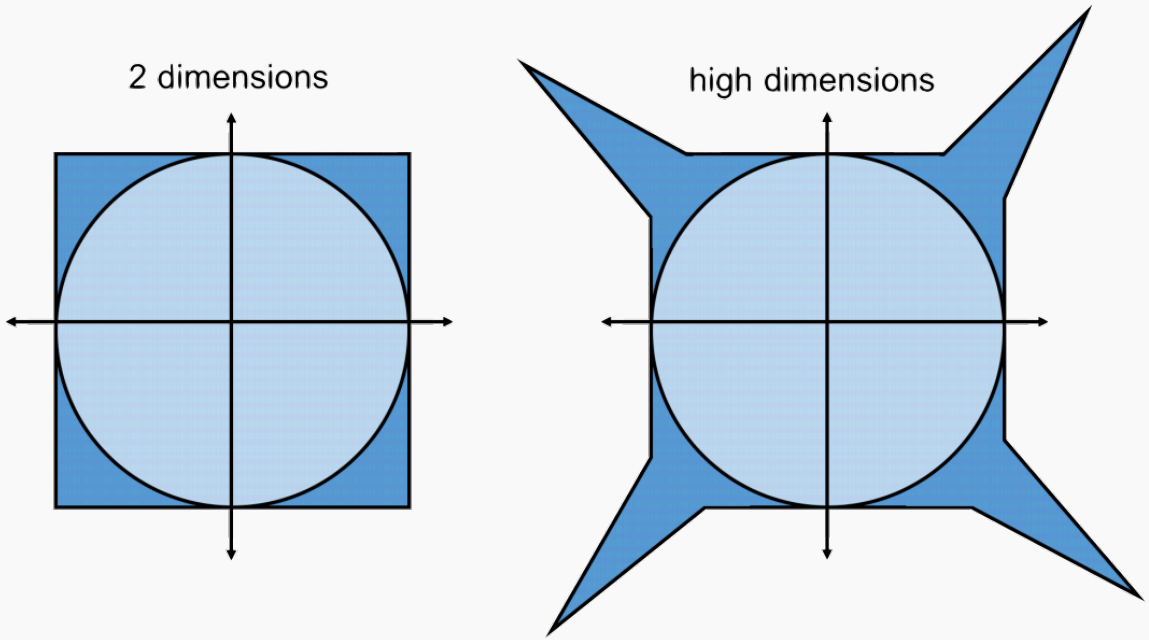
Some other ways to see these shapes are very different:

- $\mathbb{E}_{\mathbf{x} \sim \mathcal{B}_d} \|\mathbf{x}\|_2^2 \leq 1$
- $\mathbb{E}_{\mathbf{x} \sim \mathcal{C}_d} \|\mathbf{x}\|_2^2 = \sum_{i=1}^{d} \mathbb{E}[x_i^2] = d/3$ where $x_i \sim \text{Unif}[-1, 1]$

$$\int_{-1}^{1} \frac{1}{2} \cdot x^2 \, dx = \frac{1}{6} x^3 \Big|_{-1}^{1} = \frac{2}{6} = 1/3$$

HIGH DIMENSIONAL CUBE

Almost all of the volume of the unit cube falls in its corners, and these corners lie far outside the unit ball.



RECENT ARTICLE

See The Journey to Define Dimension from Quanta Magazine for another fun example comparing cubes to balls! Article posted last week.

Despite **all this** warning that low-dimensional space looks nothing like high-dimensional space, next we are going to learn about how to **compress high dimensional vectors to low dimensions.**

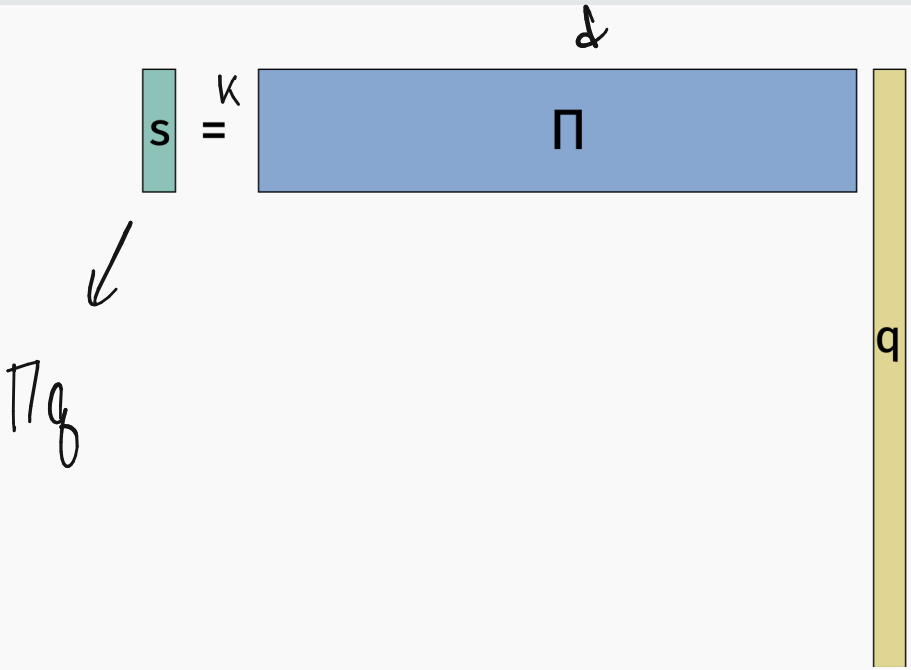
We will be very careful not to compress things too far. An extremely simple method known as Johnson-Lindenstrauss Random Projection pushes right up to the edge of how much compression is possible.

BREAK

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 $\Pi : \mathbb{R}^d \to \mathbb{R}^k$ where $k = O\left(\frac{\log n}{\epsilon^2}\right)$ such that for all i, j,

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



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 ϵ, $(1+\epsilon)^2 = 1 + O(\epsilon)$ and $(1-\epsilon)^2 = 1 - O(\epsilon)$.

And this is equivalent to:

Lemma (Johnson-Lindenstrauss, 1984)

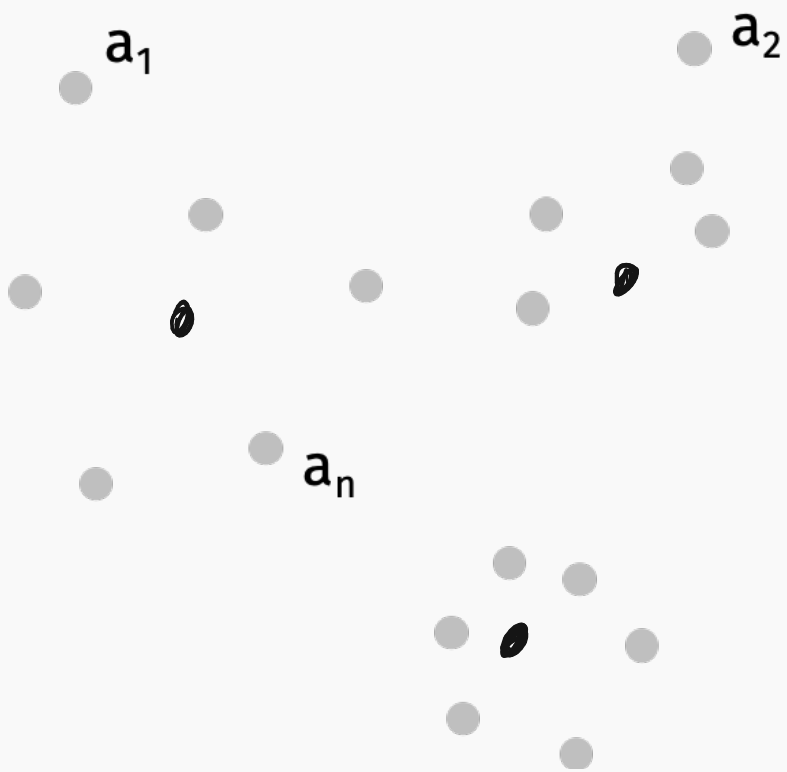
For any set of n data points $\mathbf{q}_1, \ldots, \mathbf{q}_n \in \mathbb{R}^d$ there exists a linear map $\Pi : \mathbb{R}^d \to \mathbb{R}^k$ where $k = O\left(\frac{\log n}{\epsilon^2}\right)$ such that for all i, j,

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

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

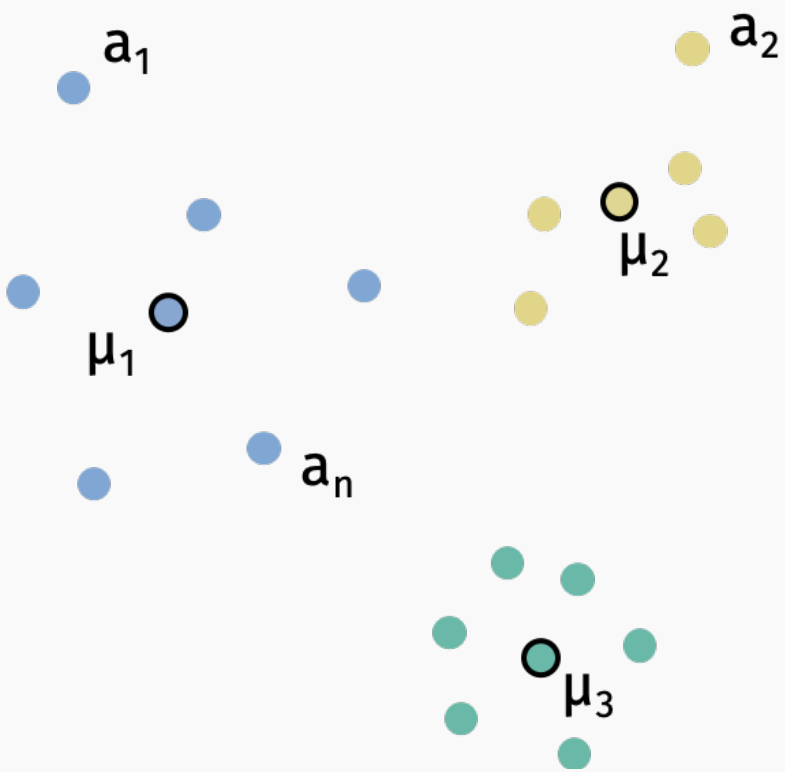
k-means clustering: Give data points $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^d$, find centers $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k \in \mathbb{R}^d$ to minimize:

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



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:

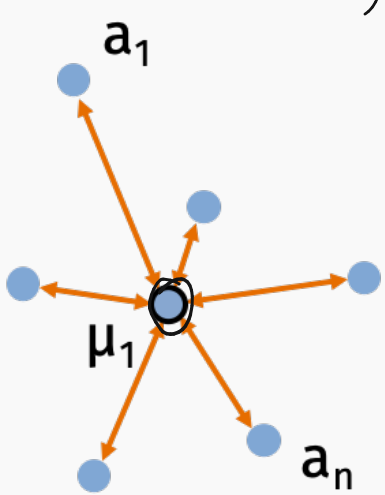
$$Cost(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k) = \sum_{i=1}^{n} \min_{j=1,\ldots,k} \|\boldsymbol{\mu}_j - \mathbf{a}_i\|_2^2$$

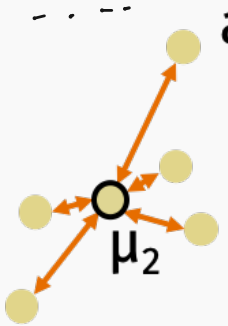


k-means clustering: Give data points $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^d$, find centers $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k \in \mathbb{R}^d$ to minimize:

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

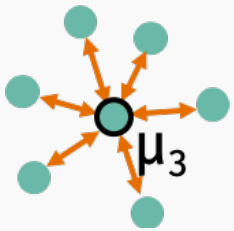
$\widetilde{cost}(\mu_1, \ldots \mu_k) = \sum \min \|\Pi \mu_j - \Pi a_i\|_2^2$





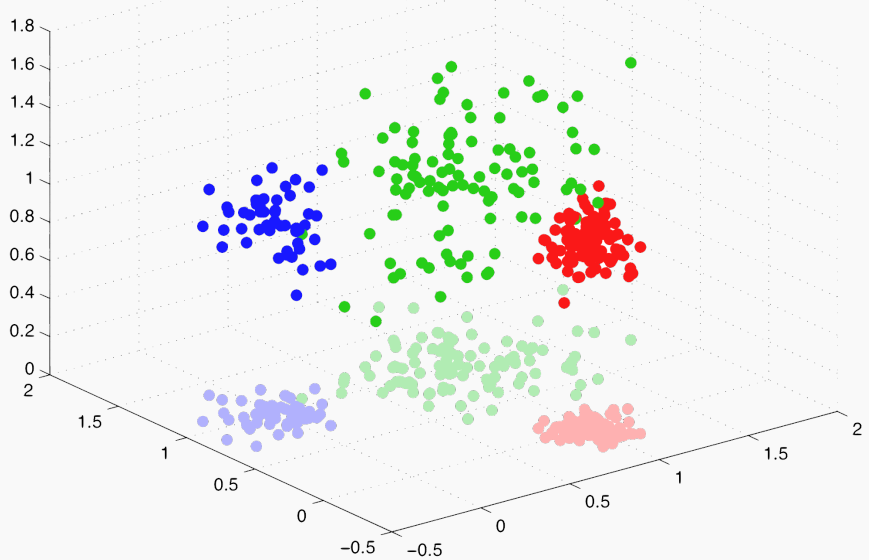
$\widetilde{cost}(\ldots)$

$= (1 \pm \varepsilon)\, cost(\ldots)$



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

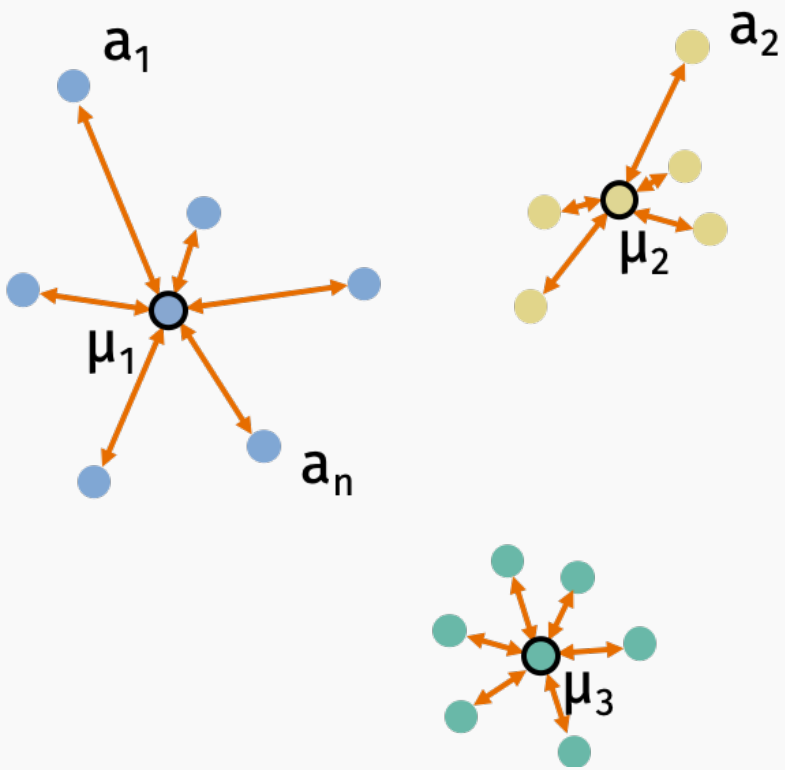
Approximation scheme: Find clusters $\tilde{C}_1, \ldots, \tilde{C}_k$ for the $k = O\left(\frac{\log n}{\epsilon^2}\right)$ dimension data set $\mathbf{\Pi a}_1, \ldots, \mathbf{\Pi a}_n$.



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

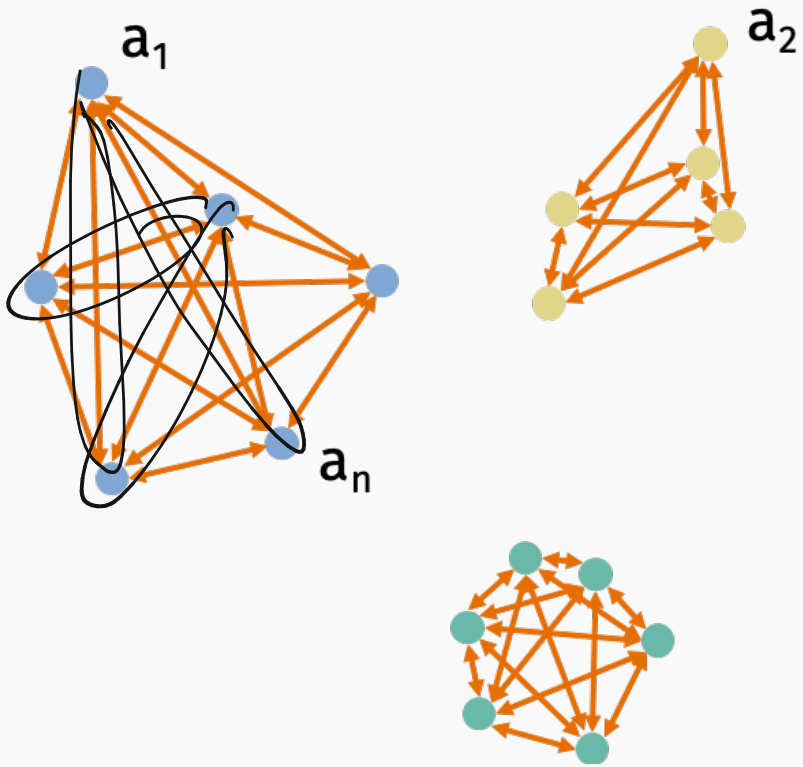
Equivalent formulation: Find clusters $C_1, \ldots, C_k \subseteq \{1, \ldots, n\}$ to minimize:

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



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

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



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

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

Claim: For any clusters $C_1, \ldots, C_k$:

$$(1-\epsilon)Cost(C_1, \ldots, C_k) \leq \widetilde{Cost}(C_1, \ldots, C_k) \leq (1+\epsilon)Cost(C_1, \ldots, C_k)$$

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:

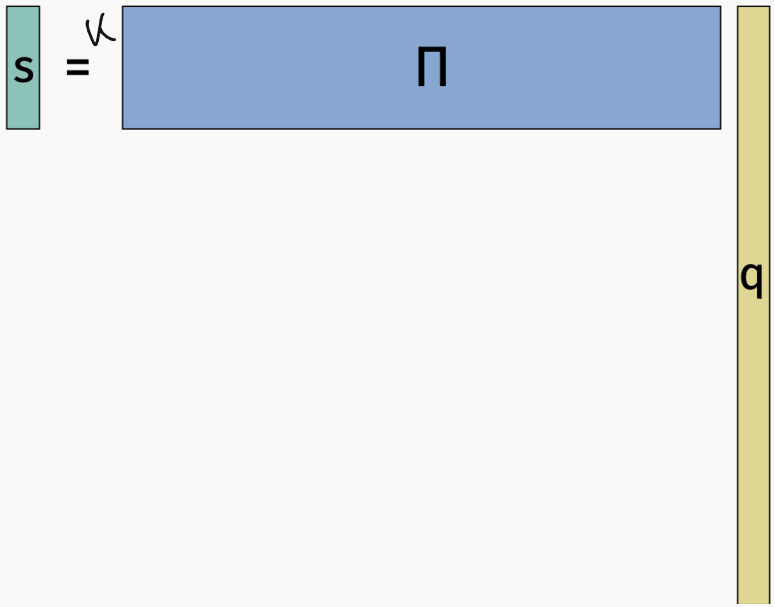
$$\begin{aligned} Cost(B_1, \ldots, B_k) &\leq \frac{1}{1-\epsilon}\widetilde{Cost}(B_1, \ldots, B_k) \\ &\leq (1+\alpha)(1+O(\epsilon))\widetilde{Cost}^* \\ &\leq (1+\alpha)(1+O(\epsilon))(1+\epsilon)Cost^* \\ &= (1+O(\alpha+\epsilon))Cost^* \end{aligned}$$

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

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 $\Pi : \mathbb{R}^d \to \mathbb{R}^k$ where $k = O\left(\frac{\log n}{\epsilon^2}\right)$ such that for all i, j,

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



Remarkably, Π can be chosen completely at random!

One possible construction: Random Gaussian.

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

The map Π 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 Π. Each with different advantages.

RANDOMIZED JL CONSTRUCTIONS

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;
```

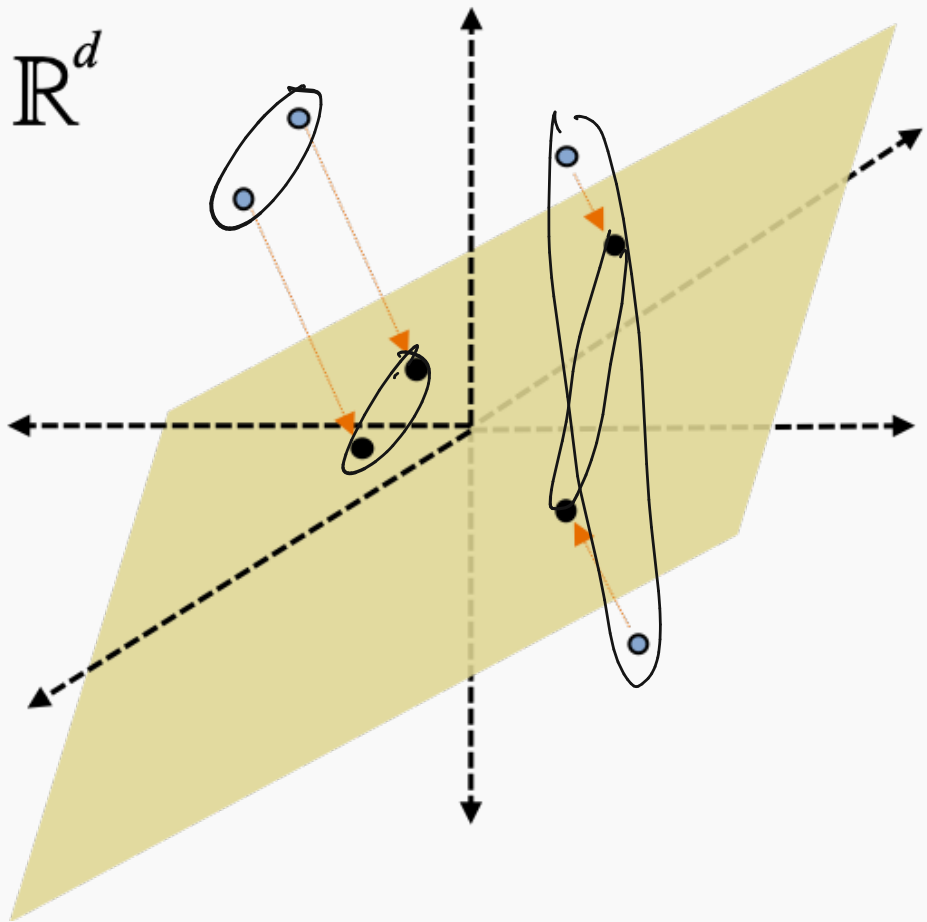
1	1	-1	-1	-1	-1	-1	-1	1	-1	-1	1	-1	-1	1	1	-1
1	1	1	-1	1	-1	-1	-1	1	1	1	1	-1	1	-1	-1	-1
1	1	-1	-1	-1	1	-1	-1	1	1	-1	1	-1	1	-1	1	-1
-1	-1	-1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1
1	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	-1	1
1	-1	-1	1	-1	1	1	-1	-1	-1	1	-1	-1	-1	1	1	1
1	1	-1	1	1	-1	1	-1	1	-1	1	-1	1	1	1	-1	-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.

RANDOM PROJECTION



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 $(1-\delta)$:

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

for i,j $\quad (1-\epsilon)\|q_i - q_j\|_2^2 \leq \|\Pi q_i - \Pi q_j\|_2^2 \leq (1+\epsilon)\|q_i - q_j\|_2^2$

$\|\Pi(q_i - q_j)\|_2^2$

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

$x = q_i - q_j$

JL FROM DISTRIBUTIONAL JL

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:

w.p. $1 - \frac{1}{10n^2}$ preserve distance for one pair of vectors

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

$1 - \frac{1}{10}$

PROOF OF DISTRIBUTIONAL JL

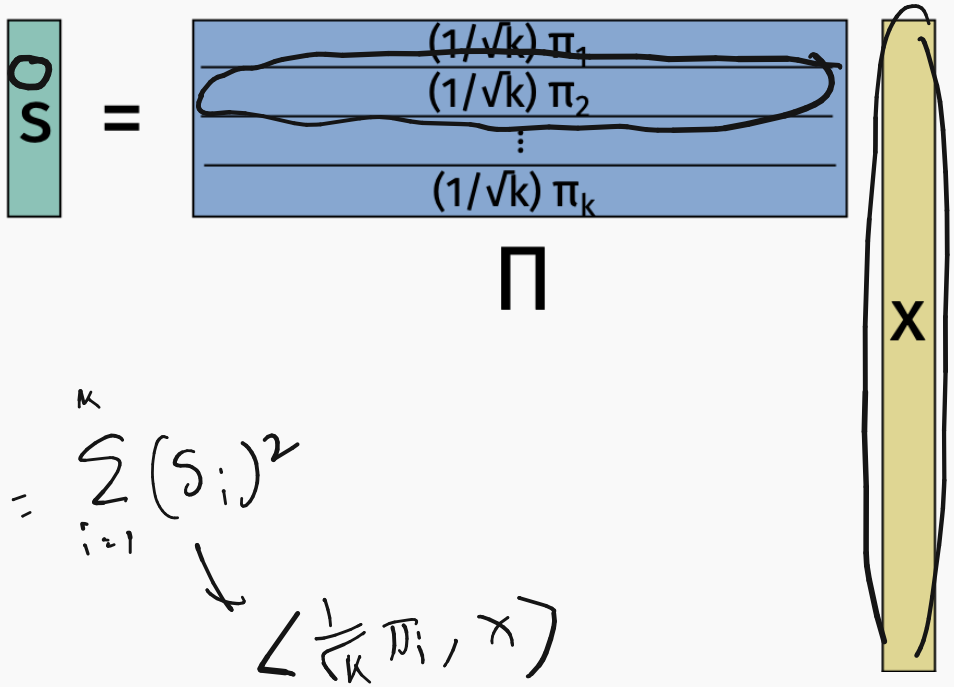
Want to argue that, with probability $(1-\delta)$,

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

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

$\mathbb{E}[\langle \pi_i, x\rangle^2] = \|x\|_2^2$

Some notation:



$S = \Pi x$

$\|\Pi x\|_2^2 = \|S\|_2^2 = \sum_{i=1}^{k} (S_i)^2$

$\langle \frac{1}{\sqrt{k}} \pi_i, x \rangle$

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

PROOF OF DISTRIBUTIONAL JL

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

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

$$= \mathbb{E}\left[(\langle \boldsymbol{\pi}_i, \mathbf{x}\rangle)^2\right]$$

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

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

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

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

Fact (Stability of Gaussian random variables)

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

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

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

$\mathrm{Var}[\langle \pi_i, x \rangle]$

PROOF OF DISTRIBUTIONAL JL

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

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

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

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

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

CONCENTRATION OF CHI-SQUARED RANDOM VARIABLES

Lemma

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

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

$\mathbb{E}Z = \|x\|_2^2$

$2e^{-k\epsilon^2/8} = \delta$

$\log(2) + -k\epsilon^2/8 = \log(\delta)$

$k\epsilon^2/8 = O(\log(1/\delta))$

$k = O(\log(1/\delta)/\epsilon^2)$

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

CONNECTION TO LAST LECTURE

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?

CONNECTION TO DIMENSIONALITY REDUCTION

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

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

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.

BREAK

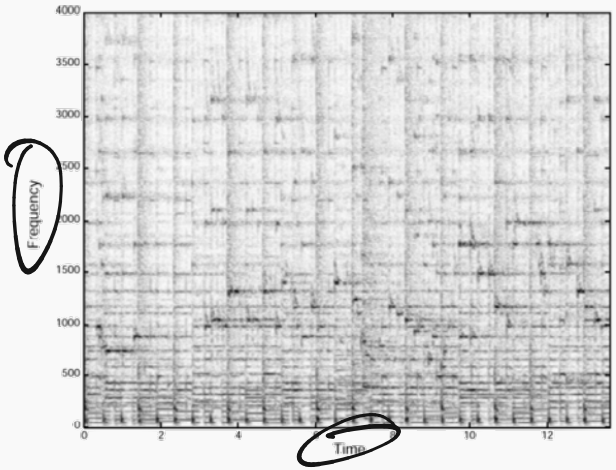
DIMENSIONALITY REDUCTION

The Johnson-Lindenstrauss Lemma let us sketch vectors and preserve their ℓ_2 Euclidean distance. We also have dimensionality reduction techniques that preserve alternative measures of similarity.

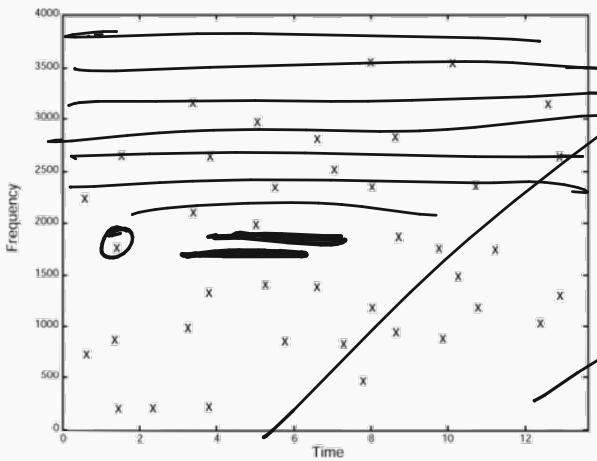
How does **Shazam** match a song clip against a library of 8 million songs (32 TB of data) in a fraction of a second?

$$\text{cost}(u_1, u_2, u_3) = \text{cost}(u_2, u_1, u_3)$$

How does **Shazam** match a song clip against a library of 8 million songs (32 TB of data) in a fraction of a second?



Spectrogram extracted from audio clip.



Processed spectrogram: used to construct audio "fingerprint" $\mathbf{q} \in \{0,1\}^d$.

Each clip is represented by a high dimensional binary vector $\mathbf{q}$.

1	0	1	1	0	0	0	1	0	0	0	0	1	1	0	1

Given $\mathbf{q}$, find any nearby "fingerprint" $\mathbf{y}$ in a database – i.e. any $\mathbf{y}$ with $\text{dist}(\mathbf{y}, \mathbf{q})$ small.

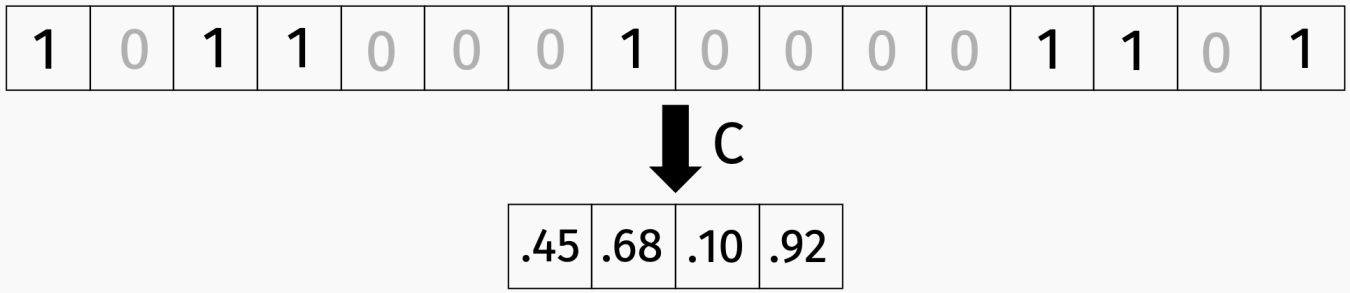
Challenges:

- Database is possibly huge: $O(nd)$ bits.
- Expensive to compute $\text{dist}(\mathbf{y}, \mathbf{q})$: $O(d)$ time.

Goal: Design a more compact sketch for comparing $\mathbf{q}, \mathbf{y} \in \{0,1\}^d$. Ideally $\ll d$ space/time complexity.

$$C(\mathbf{q}) \in \mathbb{R}^k$$

$$C(\mathbf{y}) \in \mathbb{R}^k$$



Homomorphic Compression:

$C(\mathbf{q})$ should be similar to $C(\mathbf{y})$ if $\mathbf{q}$ is similar to $\mathbf{y}$.

Definition (Jaccard Similarity)

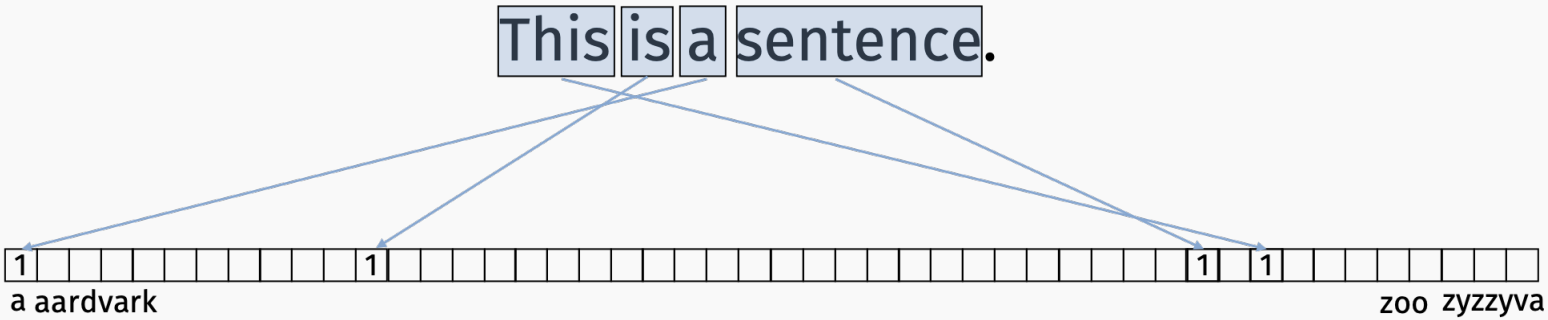
$$J(\mathbf{q}, \mathbf{y}) = \frac{|\mathbf{q} \cap \mathbf{y}|}{|\mathbf{q} \cup \mathbf{y}|} = \frac{\text{\# of non-zero entries in common}}{\text{total \# of non-zero entries}}$$

Natural similarity measure for binary vectors. $0 \leq J(\mathbf{q}, \mathbf{y}) \leq 1$.

Can be applied to any data which has a natural binary representation (more than you might think).

JACCARD SIMILARITY FOR DOCUMENT COMPARISON

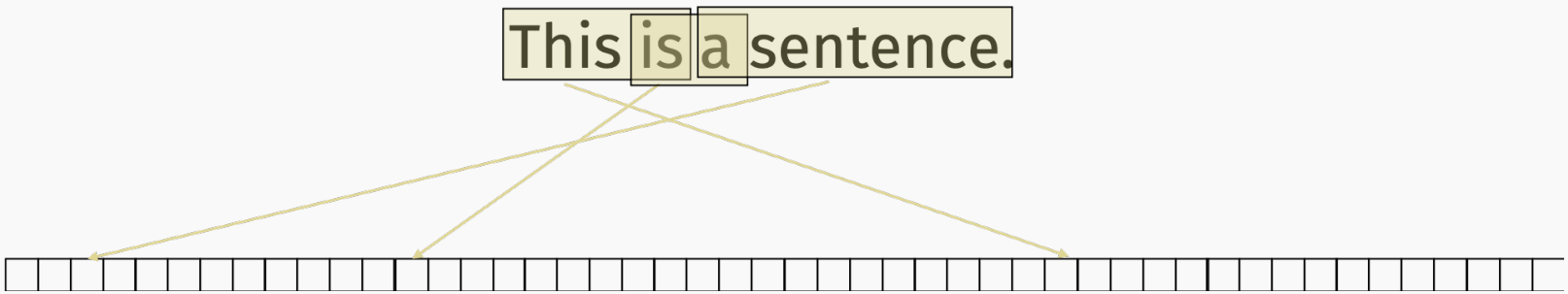
"Bag-of-words" model:



How many words do a pair of documents have in common?

JACCARD SIMILARITY FOR DOCUMENT COMPARISON

"Bag-of-words" model:

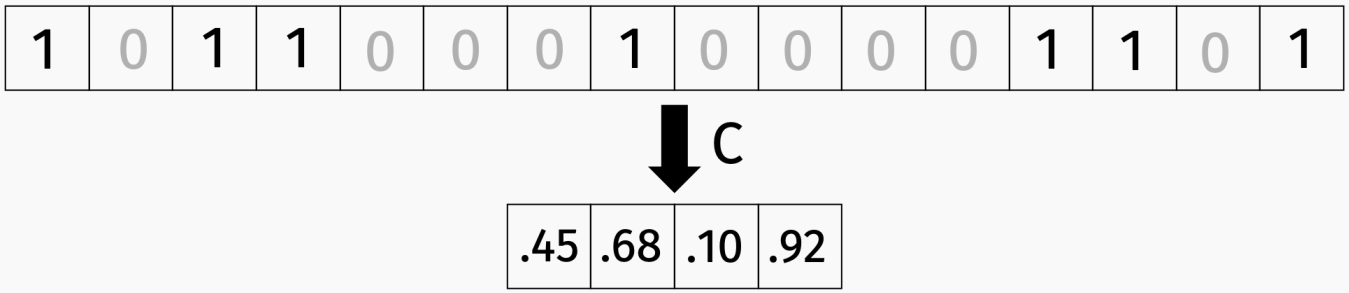


How many bigrams do a pair of documents have in common?

APPLICATIONS: DOCUMENT SIMILARITY

- Finding duplicate or new duplicate documents or webpages.
- Change detection for high-speed web caches.
- Finding near-duplicate emails or customer reviews which could indicate spam.

Goal: Design a compact sketch $C : \{0, 1\} \to \mathbb{R}^k$:



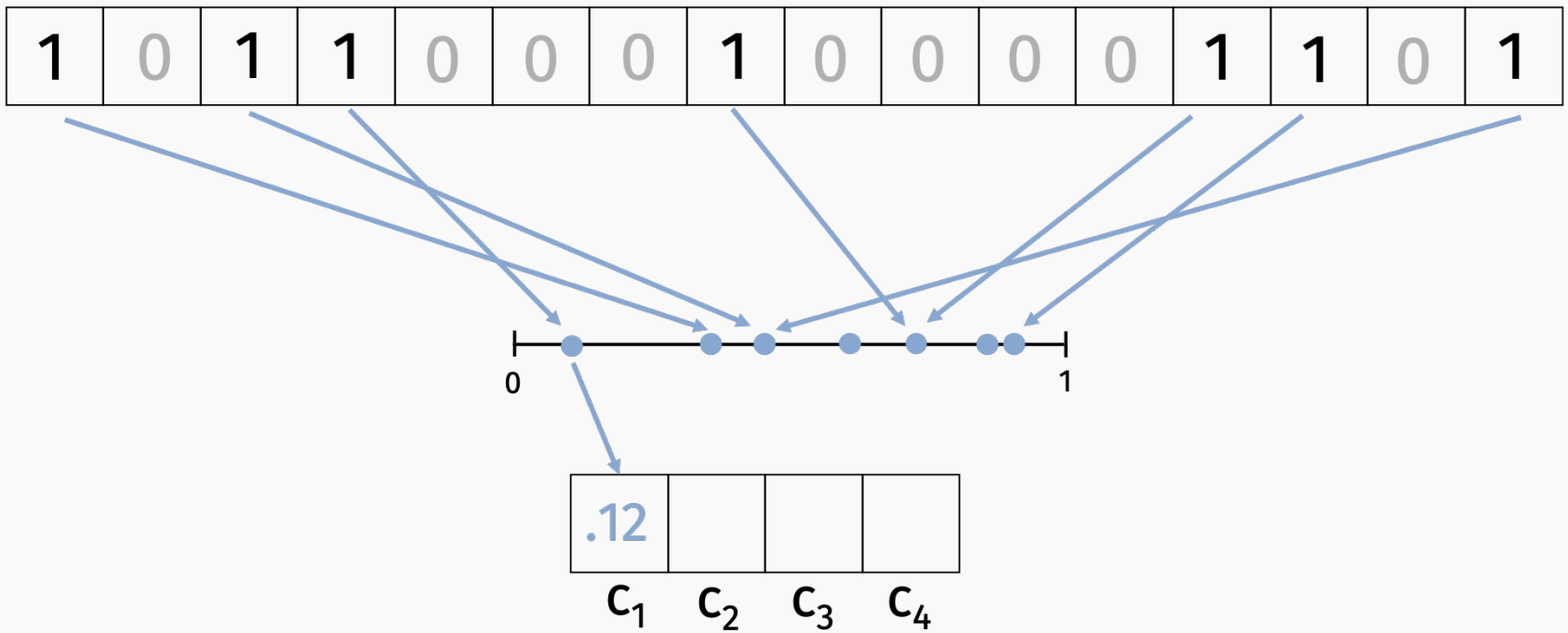
Homomorphic Compression: Want to use $C(\mathbf{q}), C(\mathbf{y})$ to approximately compute the Jaccard similarity $J(\mathbf{q}, \mathbf{y})$.

MinHash (Broder, '97):

- Choose k random hash functions $h_1, \ldots, h_k : \{1, \ldots, n\} \to [0, 1]$.
- For $i \in 1, \ldots, k$, let $c_i = \min_{j, \mathbf{q}_j = 1} h_i(j)$.
- $C(\mathbf{q}) = [c_1, \ldots, c_k]$.

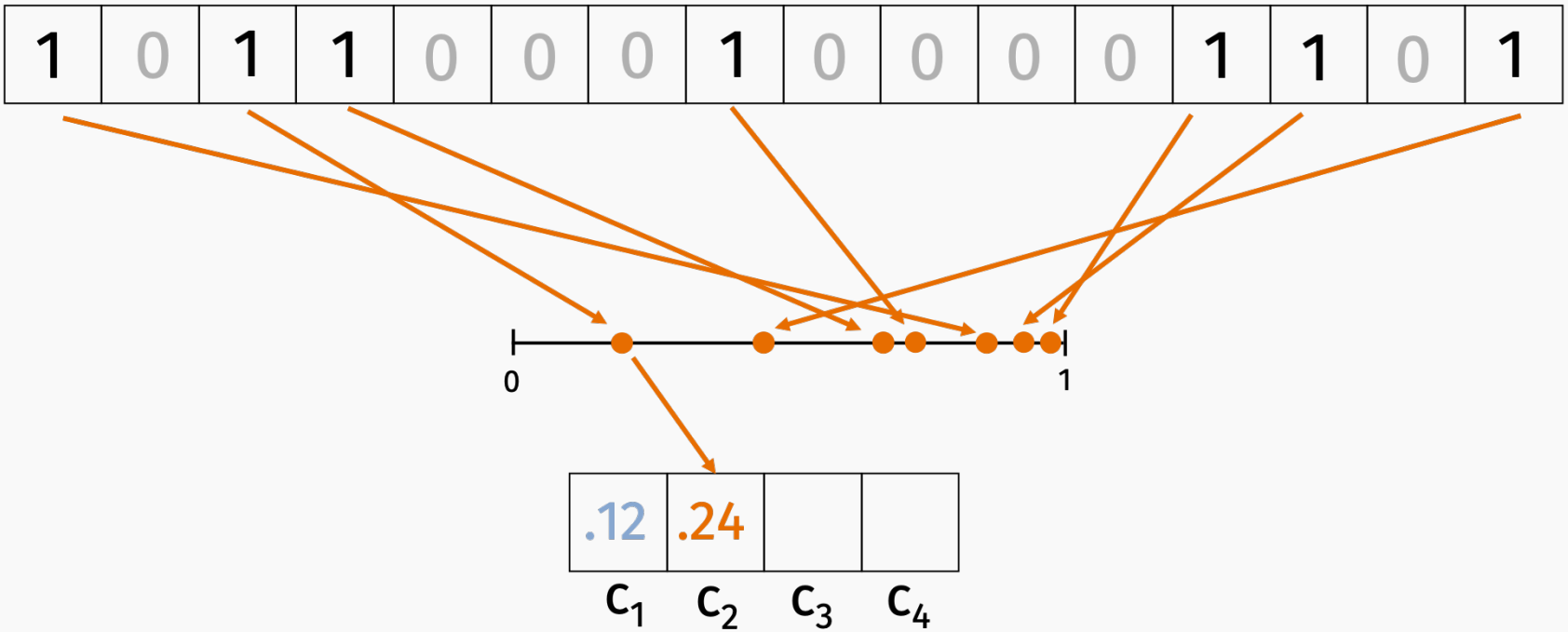
MinHash (Broder, '97):

- Choose k random hash functions $h_1, \ldots, h_k : \{1, \ldots, n\} \to [0, 1]$.
- For $i \in 1, \ldots, k$, let $c_i = \min_{j, \mathbf{q}_j = 1} h_i(j)$.
- $C(\mathbf{q}) = [c_1, \ldots, c_k]$.

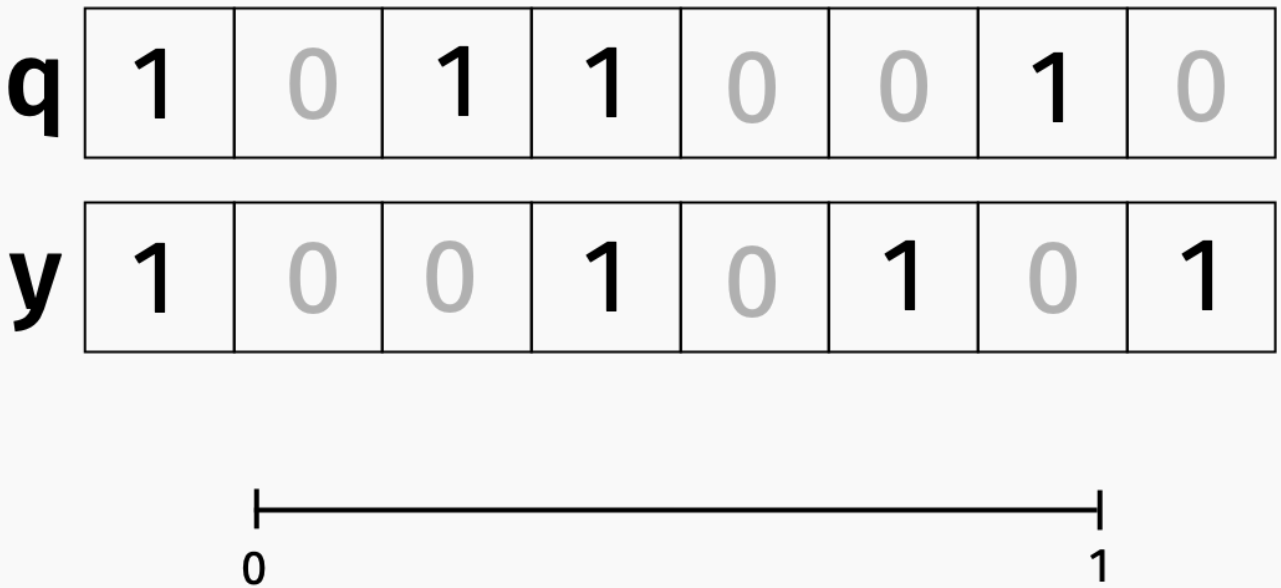


MINHASH

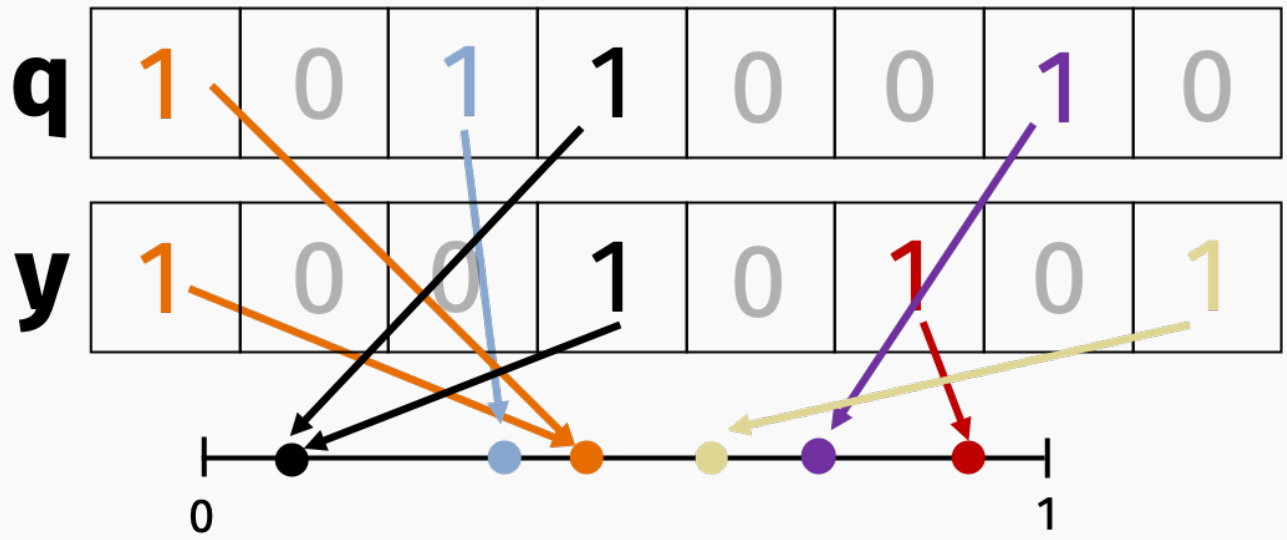
- Choose k random hash functions $h_1, \ldots, h_k : \{1, \ldots, n\} \to [0, 1]$.
- For $i \in 1, \ldots, k$, let $c_i = \min_{j, \mathbf{q}_j = 1} h_i(j)$.
- $C(\mathbf{q}) = [c_1, \ldots, c_k]$.



Claim: $\Pr[c_i(q) = c_i(y)] = J(q, y)$.



Claim: $\Pr[c_i(\mathbf{q}) = c_i(\mathbf{y})] = J(\mathbf{q}, \mathbf{y})$.



Every non-zero index in $\mathbf{q} \cup \mathbf{y}$ is equally likely to produce the lowest hash value. $c_i(\mathbf{q}) = c_i(\mathbf{y})$ only if this index is 1 in <u>both</u> $\mathbf{q}$ and $\mathbf{y}$. There are $\mathbf{q} \cap \mathbf{y}$ such indices. So:

$$\Pr[c_i(\mathbf{q}) = c_i(\mathbf{y})] = \frac{\mathbf{q} \cap \mathbf{y}}{\mathbf{q} \cup \mathbf{y}} = J(\mathbf{q}, \mathbf{y})$$

Return: $\tilde{J} = \frac{1}{k} \sum_{i=1}^{k} \mathbb{1}[c_i(\mathsf{q}) = c_i(\mathsf{y})]$.

Unbiased estimate for Jaccard similarity:

$$\mathbb{E}\tilde{J} =$$

C(q)	.12	.24	.76	.35

C(y)	.12	.98	.76	.11

The more repetitions, the lower the variance.

Let $J = J(\mathbf{q}, \mathbf{y})$ denote the true Jaccard similarity.

Estimator: $\tilde{J} = \frac{1}{k} \sum_{i=1}^{k} \mathbb{1}[c_i(\mathbf{q}) = c_i(\mathbf{y})]$.

$$\text{Var}[\tilde{J}] =$$

Plug into Chebyshev inequality. How large does k need to be so that with probability $> 1 - \delta$:

$$|J - \tilde{J}| \leq \epsilon?$$

Chebyshev inequality: As long as $k = O\left(\frac{1}{\epsilon^2 \delta}\right)$, then with prob. $1 - \delta$,

$$J(\mathbf{q}, \mathbf{y}) - \epsilon \leq \tilde{J}\left(C(\mathbf{q}), C(\mathbf{y})\right) \leq J(\mathbf{q}, \mathbf{y}) + \epsilon.$$

And $\tilde{J}$ only takes $O(k)$ time to compute! Independent of original fingerprint dimension d.

Linear dependence on $\frac{1}{\delta}$ is not good! Suppose we have a database of n songs slips, and Shazam wants to ensure the similarity between a query $\mathbf{q}$ and every song clip $\mathbf{y}$ is approximated well. Cam be improved to $\log(1/\delta)$ dependence using exponential concentration inequalities.