CS-GY 6763 Lecture 5: Dimensionality Reduction

NYU, Prof. Ainesh Bakshi

Dimensionality Reduction

- Despite all our warnings from last class 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.**

Dimensionality Reduction

- Despite all our warnings from last class 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.

Dimensionality Reduction

- Despite all our warnings from last class 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 <u>too</u> far.
- An extremely simple method known as Johnson-Lindenstrauss Random Projection pushes right up to the edge of how much compression is possible.

Euclidean Dimensionality Reduction

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,

Euclidean Dimensionality Reduction

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,*

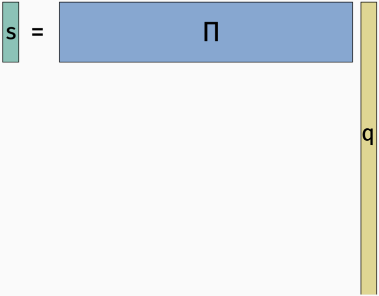
$$(1-\epsilon)\|\mathbf{q}_i - \mathbf{q}_j\|_2 \leq \|\mathbf{\Pi}\mathbf{q}_i - \mathbf{\Pi}\mathbf{q}_j\|_2 \leq (1+\epsilon)\|\mathbf{q}_i - \mathbf{q}_j\|_2.$$

Euclidean Dimensionality Reduction

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,

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



Euclidean Dimensionality Reduction

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,

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

Euclidean Dimensionality Reduction

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,

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

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

Tons of Applications

Make pretty much any computation involving vectors faster and more space efficient.

- Faster vector search (used in image search, AI-based web search, Retrieval Augmented Generation (RAG), etc.).
- Faster machine learning (today we will see an application to speeding up clustering).
- Faster numerical linear algebra.

Tons of Applications

Make pretty much any computation involving vectors faster and more space efficient.

- Faster vector search (used in image search, AI-based web search, Retrieval Augmented Generation (RAG), etc.).
- Faster machine learning (today we will see an application to speeding up clustering).
- Faster numerical linear algebra.

Only useful if we can explicity construct a JL map Π and apply efficiently to vectors.

Euclidean Dimensionality Reduction

Remarkably, Π can be chosen completely at random!

Euclidean Dimensionality Reduction

Remarkably, Π can be chosen completely at random!

One possible construction: Random Gaussian.

$$\mathbf{\Pi}_{i,j} = \frac{1}{\sqrt{k}}\mathcal{N}(0,1) \quad \text{where } k \text{ is reduced dimension}$$

Euclidean Dimensionality Reduction

Remarkably, Π can be chosen completely at random!

One possible construction: Random Gaussian.

$$\mathbf{\Pi}_{i,j} = \frac{1}{\sqrt{k}}\mathcal{N}(0,1) \quad \text{where } k \text{ is reduced dimension}$$

The map $\mathbf{\Pi}$ is **oblivious to the data set**. This stands in contrast to other vector compression methods you might know like PCA.

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

Euclidean Dimensionality Reduction

Remarkably, Π can be chosen completely at random!

One possible construction: Random Gaussian.

$$\mathbf{\Pi}_{i,j} = \frac{1}{\sqrt{k}}\mathcal{N}(0,1) \quad \text{where } k \text{ is reduced dimension}$$

The map $\mathbf{\Pi}$ is **oblivious to the data set**. This stands in contrast to other vector compression methods you might know like PCA.

[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;
```

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	-1	1	-1	1

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

A scaled random orthogonal matrix $\sqrt{\frac{d}{k}} \cdot \mathbf{Q}$ also works. I.e. with $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}_{k \times k}$.

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 scaled random orthogonal matrix $\sqrt{\frac{d}{k}} \cdot \mathbf{Q}$ also works. I.e. with $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}_{k\times k}$.

For this reason, the JL operation is often called a “random projection”, even though it technically is not a projection when $\mathbf{\Pi}'s$ entries are i.i.d.

Random Projection

Can anyone see why $\mathbf{\Pi}$ is similar to a projection matrix? I.e., a matrix satisfying $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}_{k \times k}$.

Random Projection

Can anyone see why $\boldsymbol{\Pi}$ is similar to a projection matrix? I.e., a matrix satisfying $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}_{k \times k}$.

Idea: Π is "close" to a projection matrix.

Consider $\Pi_{i,j} = \frac{1}{\sqrt{k}} \cdot \mathrm{Rad}(0.5)$ i.e. uniformly ± 1.

Random Projection

Can anyone see why $\mathbf{\Pi}$ is similar to a projection matrix? I.e., a matrix satisfying $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}_{k \times k}$.

Idea: Π is "close" to a projection matrix.

Consider $\Pi_{i,j} = \frac{1}{\sqrt{k}} \cdot \mathrm{Rad}(0.5)$ i.e. uniformly ± 1.

Diagonal Entries of $\Pi\Pi^T$:

$$(\Pi\Pi^T)_{i,i} = \langle \Pi_{i,:}, \Pi_{i,:} \rangle = \frac{1}{k}\sum_{j=1}^{d} \mathrm{Rad}(0.5)^2 = \frac{d}{k}$$

Can anyone see why $\mathbf{\Pi}$ is similar to a projection matrix? I.e., a matrix satisfying $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}_{k \times k}$.

Idea: Π is "close" to a projection matrix.

Consider $\Pi_{i,j} = \frac{1}{\sqrt{k}} \cdot \mathrm{Rad}(0.5)$ i.e. uniformly ± 1.

Diagonal Entries of $\Pi\Pi^T$:

$$(\Pi\Pi^T)_{i,i} = \langle \Pi_{i,:}, \Pi_{i,:} \rangle = \frac{1}{k} \sum_{j=1}^{d} \mathrm{Rad}(0.5)^2 = \frac{d}{k}$$

Random Projection

Can anyone see why $\mathbf{\Pi}$ is similar to a projection matrix? I.e., a matrix satisfying $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}_{k \times k}$.

Idea: Π is "close" to a projection matrix.

Consider $\Pi_{i,j} = \frac{1}{\sqrt{k}} \cdot \mathrm{Rad}(0.5)$ i.e. uniformly ± 1.

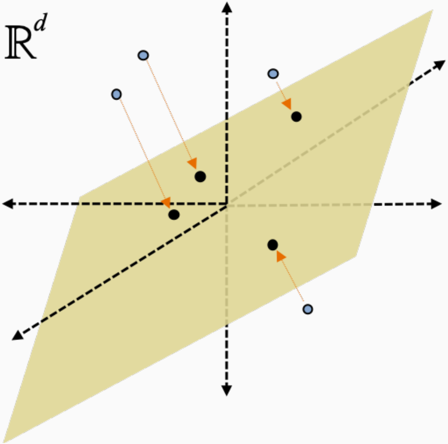
Diagonal Entries of $\Pi\Pi^T$:

$$(\Pi\Pi^T)_{i,i} = \langle \Pi_{i,:}, \Pi_{i,:} \rangle = \frac{1}{k} \sum_{j=1}^{d} \mathrm{Rad}(0.5)^2 = \frac{d}{k}$$

Off-diagonal Entries of $\Pi\Pi^T$:

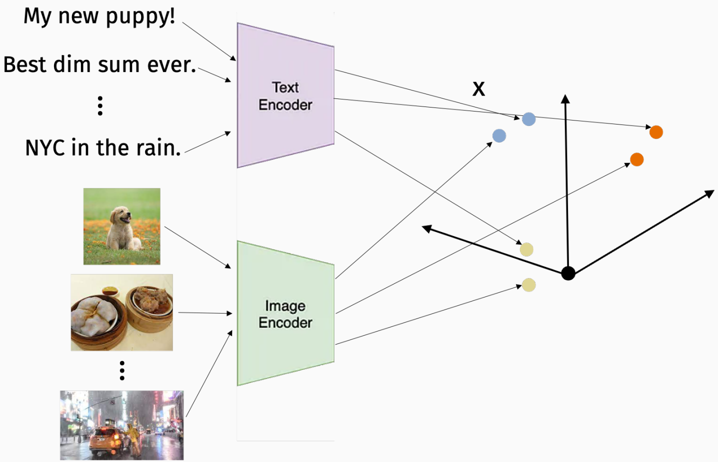
$$\mathbb{E}(\Pi\Pi^T)_{i,j} = \mathbb{E}\langle \Pi_{i,:}, \Pi_{j,:} \rangle = \frac{1}{k} \sum_{l=1}^{d} \mathbb{E}\mathrm{Rad}(0.5) \cdot \mathbb{E}\mathrm{Rad}(0.5) = 0$$

Random Projection



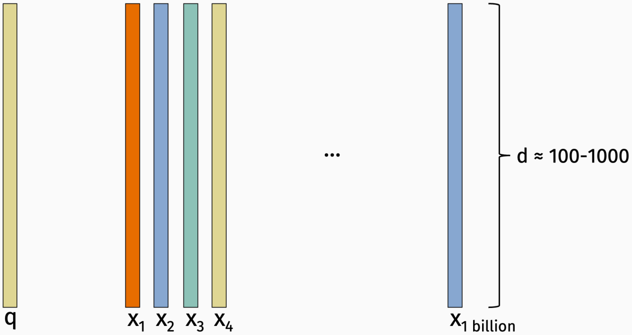
Intuition: Multiplying by a random matrix mimics the process of projecting onto a random k dimensional subspace in d dimensions.

Application: The New Paradigm for Search



Use neural network (BERT, CLIP, etc.) to convert documents, images, etc. to high dimensional vectors. Results matching search should have similar vector embeddings.

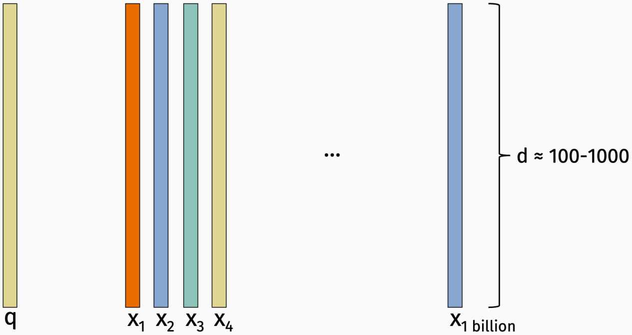
Application: The New Paradigm for Search



Finding results for a query reduces to finding the nearest vector in a vector database, with similarity typically measured by Euclidean distance.

$$\text{Find } \arg\min_i \|\mathbf{q} - \mathbf{x}_i\|_2$$

Application: The New Paradigm for Search



Finding results for a query reduces to finding the nearest vector in a vector database, with similarity typically measured by Euclidean distance.

$$\text{Find } \arg\min_i \|\mathbf{q} - \mathbf{x}_i\|_2$$

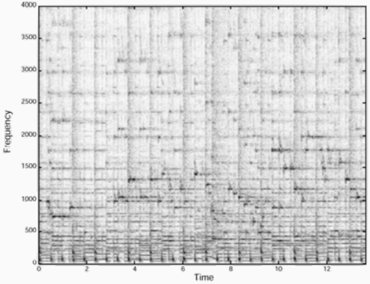
This is a massive algorithmic challenge!

Another Example of Vector Search

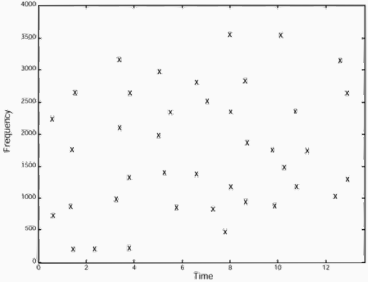
Shazam can match a song clip against a library of 20 million songs (TB of data) in a fraction of a second. Whole system based on vector embeddings + search.

Another Example of Vector Search

Shazam can match a song clip against a library of 20 million songs (TB of data) in a fraction of a second. Whole system based on vector embeddings + search.



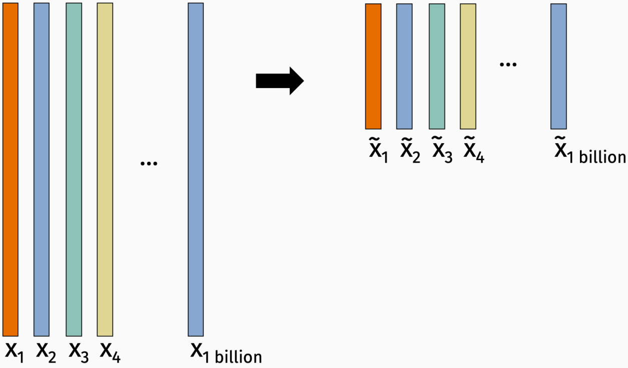
Spectrogram extracted from audio clip.



Processed spectrogram: used to construct audio “fingerprint” $\mathbf{x} \in \mathbb{R}^d$.

Application: The New Paradigm for Search

Main computational cost is repeatedly computing $\|\mathbf{q} - \mathbf{x}_i\|_2$ for candidate result $\mathbf{x}_i$.



Vector compression leads to faster distance computations. Not only is computational complexity reduced, but we can fit more database vectors in memory.

Lemma (Johnson-Lindenstrauss, 1984)

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 $k = O\left(\frac{\log(n)}{\epsilon^2}\right)$*, then with probability* $99/100$*, for for all* i, j*,*

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

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* <u>*any vector*</u> $\mathbf{x}$*, 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?

JL from Distributional JL

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

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

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 $1 - \delta$,

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

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 $\delta = \frac{1}{100n^2}$. Since there are $< n^2$ total i, j pairs, by a union bound we have that with probability 99/100, the above will hold <u>for all</u> i, j, as long as we compress to:

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 $\delta = \frac{1}{100n^2}$. Since there are $< n^2$ total i, j pairs, by a union bound we have that with probability 99/100, the above will hold <u>for all</u> i, j, as long as we compress to:

$$k = O\left(\frac{\log(1/(1/100n^2))}{\epsilon^2}\right) = O\left(\frac{\log n}{\epsilon^2}\right) \text{ dimensions.} \quad \square$$

Proof of Distributional JL

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

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

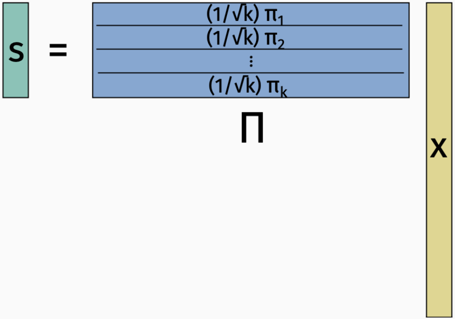
Proof of Distributional JL

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

Some notation:



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

Proof of Distributional JL

Intermediate Claim: Let $\boldsymbol{\pi}$ be a length d vector with $\mathcal{N}(0,1)$ entries.

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

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

Proof of Distributional JL

Intermediate Claim: Let $\boldsymbol{\pi}$ be a length d vector with $\mathcal{N}(0,1)$ entries.

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

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

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

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

Proof of Distributional JL

Intermediate Claim: Let $\boldsymbol{\pi}$ be a length d vector with $\mathcal{N}(0,1)$ entries.

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

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

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

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

Proof of Distributional JL

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

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

Proof of Distributional JL

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

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

We have that $Z_i \cdot x[i]$ is a normal $\mathcal{N}(0, x[i]^2)$ random variable.

Proof of Distributional JL

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

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

We have that $Z_i \cdot x[i]$ is a normal $\mathcal{N}(0, x[i]^2)$ random variable.

$$\mathbb{E}[\langle \pi, x \rangle^2] = \mathrm{Var}[\langle \pi, x \rangle^2] = \mathrm{Var}[\sum_{i=1}^{d} \mathcal{N}(0, x[i]^2)]$$

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

Proof of Distributional JL

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

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

We have that $Z_i \cdot x[i]$ is a normal $\mathcal{N}(0, x[i]^2)$ random variable.

$$\mathbb{E}[\langle \pi, x \rangle^2] = \mathrm{Var}[\langle \pi, x \rangle^2] = \mathrm{Var}[\sum_{i=1}^{d} \mathcal{N}(0, x[i]^2)]$$

$$= \sum_{i=1}^{d} \mathrm{Var}[\mathcal{N}(0, x[i]^2)] = \sum_{i=1}^{d} x[i]^2 = \|\mathbf{x}\|_2^2.$$

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

Stable Random Variables

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

Fact (Stability of Gaussian random variables)

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

Stable Random Variables

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

Fact (Stability of Gaussian random variables)

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

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

Stable Random Variables

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

Fact (Stability of Gaussian random variables)

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

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

So $\mathbb{E}\|\mathbf{\Pi}\mathbf{x}\|_2^2 = \mathbb{E}\left[(\langle \boldsymbol{\pi}, \mathbf{x} \rangle)^2\right] = \mathbb{E}\left[\mathcal{N}(0, \|\mathbf{x}\|_2^2)^2\right] = \|\mathbf{x}\|_2^2$, as desired.

Proof of Distributional JL

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

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

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

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

Proof of Distributional JL

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

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

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

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

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

Concentration of Chi-Squared Random Variables

Lemma

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

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

Concentration of Chi-Squared Random Variables

Lemma

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

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

Proof: Recall, $H = \|\mathbf{\Pi x}\|_2^2 = \frac{1}{k}\sum_{i=1}^{k} \mathcal{N}(0, \|\mathbf{x}\|_2^2)^2$.

Concentration of Chi-Squared Random Variables

Lemma

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

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

Proof: Recall, $H = \|\mathbf{\Pi x}\|_2^2 = \frac{1}{k}\sum_{i=1}^{k} \mathcal{N}(0, \|\mathbf{x}\|_2^2)^2$.

Also, $\mathbb{E}[H] = \mathbb{E}[\|\mathbf{\Pi x}\|_2^2] = \|\mathbf{x}\|_2^2$. Plugging this in,

$$\Pr[|\|\mathbf{x}\|_2^2 - \|\mathbf{\Pi x}\|_2^2| \geq \epsilon \|\mathbf{x}\|_2^2] \leq 2e^{-k\epsilon^2/8} = \delta$$

Concentration of Chi-Squared Random Variables

Lemma

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

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

Proof: Recall, $H = \|\mathbf{\Pi x}\|_2^2 = \frac{1}{k}\sum_{i=1}^{k}\mathcal{N}(0, \|\mathbf{x}\|_2^2)^2$.

Also, $\mathbb{E}[H] = \mathbb{E}[\|\mathbf{\Pi x}\|_2^2] = \|\mathbf{x}\|_2^2$. Plugging this in,

$$\Pr[|\|\mathbf{x}\|_2^2 - \|\mathbf{\Pi x}\|_2^2| \geq \epsilon\|\mathbf{x}\|_2^2] \leq 2e^{-k\epsilon^2/8} = \delta$$

Concentration of Chi-Squared Random Variables

Lemma

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

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

Proof: Recall, $H = \|\mathbf{\Pi x}\|_2^2 = \frac{1}{k}\sum_{i=1}^{k} \mathcal{N}(0, \|\mathbf{x}\|_2^2)^2$.

Also, $\mathbb{E}[H] = \mathbb{E}[\|\mathbf{\Pi x}\|_2^2] = \|\mathbf{x}\|_2^2$. Plugging this in,

$$\Pr[|\|\mathbf{x}\|_2^2 - \|\mathbf{\Pi x}\|_2^2| \geq \epsilon \|\mathbf{x}\|_2^2] \leq 2e^{-k\epsilon^2/8} = \delta$$

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

Concentration of Chi-Squared Random Variables

Lemma

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

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

Proof: Recall, $H = \|\mathbf{\Pi x}\|_2^2 = \frac{1}{k}\sum_{i=1}^{k} \mathcal{N}(0, \|\mathbf{x}\|_2^2)^2$.

Also, $\mathbb{E}[H] = \mathbb{E}[\|\mathbf{\Pi x}\|_2^2] = \|\mathbf{x}\|_2^2$. Plugging this in,

$$\Pr[|\|\mathbf{x}\|_2^2 - \|\mathbf{\Pi x}\|_2^2| \geq \epsilon \|\mathbf{x}\|_2^2] \leq 2e^{-k\epsilon^2/8} = \delta$$

$$2e^{-k\epsilon^2/8} = \delta$$
$$\implies -k\epsilon^2/8 = \log(\delta/2)$$

Concentration of Chi-Squared Random Variables

Lemma

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

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

Proof: Recall, $H = \|\mathbf{\Pi x}\|_2^2 = \frac{1}{k}\sum_{i=1}^{k} \mathcal{N}(0, \|\mathbf{x}\|_2^2)^2$.

Also, $\mathbb{E}[H] = \mathbb{E}[\|\mathbf{\Pi x}\|_2^2] = \|\mathbf{x}\|_2^2$. Plugging this in,

$$\Pr[|\|\mathbf{x}\|_2^2 - \|\mathbf{\Pi x}\|_2^2| \geq \epsilon\|\mathbf{x}\|_2^2] \leq 2e^{-k\epsilon^2/8} = \delta$$

$$\begin{aligned} 2e^{-k\epsilon^2/8} &= \delta \\ \implies -k\epsilon^2/8 &= \log(\delta/2) \\ \implies k &= O\left(\frac{\log(1/\delta)}{\epsilon^2}\right) \end{aligned}$$

Connection to Earlier Part of Lecture

If high dimensional geometry is so different from low-dimensional geometry, why is <u>dimensionality reduction possible?</u>

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 nearly orthogonal unit vectors:

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

When we reduce to k dimensions with JL, we still expect these vectors to be nearly orthogonal. Why?

Connection to Dimensionality Reduction

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

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

When we reduce to k dimensions with JL, we still expect these vectors to be nearly orthogonal. Why?

$$\|\Pi\mathbf{x}_i - \Pi\mathbf{x}_j\|_2^2 = \|\Pi\mathbf{x}_i\|_2^2 + \|\Pi\mathbf{x}_j\|_2^2 - 2\langle \Pi\mathbf{x}_i, \Pi\mathbf{x}_j\rangle$$

Connection to Dimensionality Reduction

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

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

When we reduce to k dimensions with JL, we still expect these vectors to be nearly orthogonal. Why?

$$\|\Pi\mathbf{x}_i - \Pi\mathbf{x}_j\|_2^2 = \|\Pi\mathbf{x}_i\|_2^2 + \|\Pi\mathbf{x}_j\|_2^2 - 2\langle \Pi\mathbf{x}_i, \Pi\mathbf{x}_j\rangle$$
$$\implies \langle \Pi\mathbf{x}_i, \Pi\mathbf{x}_j\rangle = \frac{1}{2}\left(\|\Pi\mathbf{x}_i\|_2^2 + \|\Pi\mathbf{x}_j\|_2^2 - \|\Pi\mathbf{x}_i - \Pi\mathbf{x}_j\|_2^2\right)$$

Connection to Dimensionality Reduction

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

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

When we reduce to k dimensions with JL, we still expect these vectors to be nearly orthogonal. Why?

$$\begin{aligned}
\|\Pi\mathbf{x}_i - \Pi\mathbf{x}_j\|_2^2 &= \|\Pi\mathbf{x}_i\|_2^2 + \|\Pi\mathbf{x}_j\|_2^2 - 2\langle \Pi\mathbf{x}_i, \Pi\mathbf{x}_j\rangle \\
\implies \langle \Pi\mathbf{x}_i, \Pi\mathbf{x}_j\rangle &= \frac{1}{2}\left(\|\Pi\mathbf{x}_i\|_2^2 + \|\Pi\mathbf{x}_j\|_2^2 - \|\Pi\mathbf{x}_i - \Pi\mathbf{x}_j\|_2^2\right) \\
&\approx \frac{1}{2}\left(\|\mathbf{x}_i\|_2^2 + \|\mathbf{x}_j\|_2^2 - \|\mathbf{x}_i - \mathbf{x}_j\|_2^2\right)
\end{aligned}$$

Connection to Dimensionality Reduction

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

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

When we reduce to k dimensions with JL, we still expect these vectors to be nearly orthogonal. Why?

$$\begin{aligned}
\|\Pi\mathbf{x}_i - \Pi\mathbf{x}_j\|_2^2 &= \|\Pi\mathbf{x}_i\|_2^2 + \|\Pi\mathbf{x}_j\|_2^2 - 2\langle \Pi\mathbf{x}_i, \Pi\mathbf{x}_j\rangle \\
\implies \langle \Pi\mathbf{x}_i, \Pi\mathbf{x}_j\rangle &= \frac{1}{2}\left(\|\Pi\mathbf{x}_i\|_2^2 + \|\Pi\mathbf{x}_j\|_2^2 - \|\Pi\mathbf{x}_i - \Pi\mathbf{x}_j\|_2^2\right) \\
&\approx \frac{1}{2}\left(\|\mathbf{x}_i\|_2^2 + \|\mathbf{x}_j\|_2^2 - \|\mathbf{x}_i - \mathbf{x}_j\|_2^2\right) \\
&\approx \langle \mathbf{x}_i, \mathbf{x}_j\rangle
\end{aligned}$$

Connection to Dimensionality Reduction

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

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

From our result last class, in k dimensions, there exists $2^{O(\epsilon^2 \cdot k)}$ unit vectors that are almost orthogonal.

Connection to Dimensionality Reduction

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

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

From our result last class, in k dimensions, there exists $2^{O(\epsilon^2 \cdot k)}$ unit vectors that are almost orthogonal.

We set $k = O(\log n/\epsilon^2)$, so $2^{O(\epsilon^2 \cdot k)} = 2^{O(\epsilon^2 \cdot \log(n)/\epsilon^2)} \geq n$ nearly orthogonal vectors, which is sufficient since we started with n vectors to begin with.

Connection to Dimensionality Reduction

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

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

From our result last class, in k dimensions, there exists $2^{O(\epsilon^2 \cdot k)}$ unit vectors that are almost orthogonal.

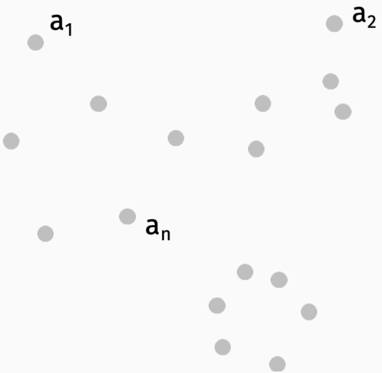
We set $k = O(\log n/\epsilon^2)$, so $2^{O(\epsilon^2 \cdot k)} = 2^{O(\epsilon^2 \cdot \log(n)/\epsilon^2)} \geq n$ nearly orthogonal vectors, which is sufficient since we started with n vectors to begin with.

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

Second Application

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:

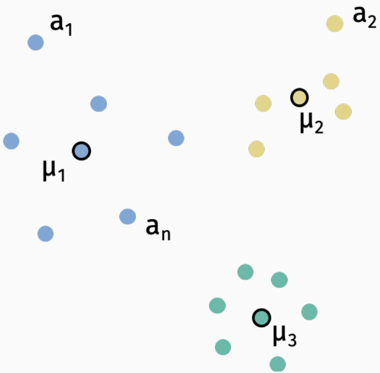
$$Cost(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k) = \sum_{i=1}^{n} \min_{j=1,\ldots,k} \|\boldsymbol{\mu}_j - \mathbf{a}_i\|_2^2$$



Sample Application

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:

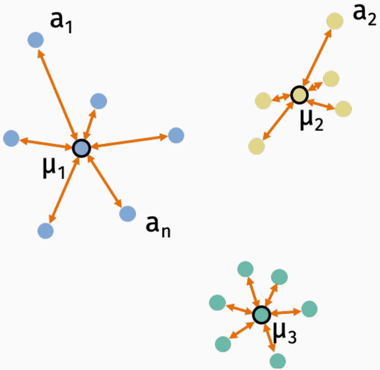
$$Cost(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k) = \sum_{i=1}^{n} \min_{j=1,\ldots,k} \|\boldsymbol{\mu}_j - \mathbf{a}_i\|_2^2$$



Sample Application

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:

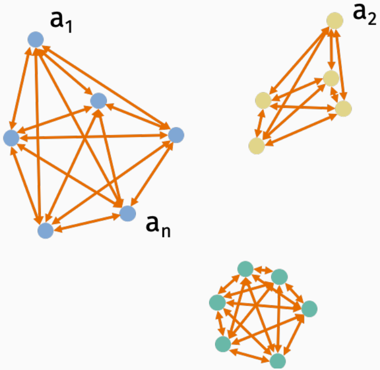
$$Cost(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k) = \sum_{i=1}^{n} \min_{j=1,\ldots,k} \|\boldsymbol{\mu}_j - \mathbf{a}_i\|_2^2$$



K-Means Clustering

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

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



Exercise: Prove this to your self.

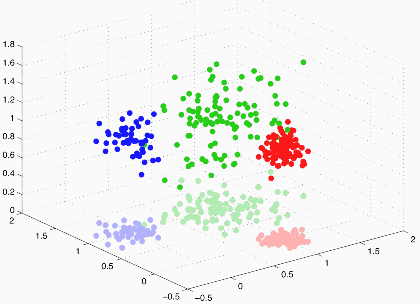
K-Means Clustering

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

K-Means Clustering

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 $\mathbf{a}_1, \ldots, \mathbf{a}_n$.

K-Means Clustering

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

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

K-Means Clustering

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

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

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

$$(1-\epsilon)Cost \leq \widetilde{Cost} \leq (1+\epsilon)Cost$$

Proof:

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

K-Means Clustering

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

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

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

$$(1-\epsilon) Cost \leq \widetilde{Cost} \leq (1+\epsilon) Cost$$

Proof:

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

$$\leq (1+\epsilon) \sum_{j=1}^{k} \frac{1}{2|C_j|} \sum_{u,v \in C_j} \|\mathbf{a}_u - \mathbf{a}_v\|_2^2$$

K-Means Clustering

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

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

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

$$(1-\epsilon)Cost \leq \widetilde{Cost} \leq (1+\epsilon)Cost$$

Proof:

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

K-Means Clustering

Suppose we find the optimal clustering $B_1, \ldots, B_k$ in the low-dimensional space, i.e. :

$$\widetilde{Cost}(B_1, \ldots, B_k) = \widetilde{Cost}^*$$

Then:

$$Cost(B_1, \ldots, B_k) \leq \frac{1}{1-\epsilon} \widetilde{Cost}(B_1, \ldots, B_k)$$

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

K-Means Clustering

Suppose we find the optimal clustering $B_1, \dots, B_k$ in the low-dimensional space, i.e. :

$$\widetilde{Cost}(B_1, \dots, B_k) = \widetilde{Cost}^*$$

Then:

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

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

K-Means Clustering

Suppose we find the optimal clustering $B_1, \ldots, B_k$ in the low-dimensional space, i.e. :

$$\widetilde{Cost}(B_1, \ldots, B_k) = \widetilde{Cost}^*$$

Then:

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

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

K-Means Clustering

Suppose we find the optimal clustering $B_1, \dots, B_k$ in the low-dimensional space, i.e. :

$$\widetilde{Cost}(B_1, \dots, B_k) = \widetilde{Cost}^*$$

Then:

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

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

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.

Jaccard Similarity

Often vector embeddings used in semantic search are binary. For such vectors, Jaccard similarity is often used instead of Euclidean distance or inner product to compute similarity.

Jaccard Similarity

Often vector embeddings used in semantic search are binary. For such vectors, Jaccard similarity is often used instead of Euclidean distance or inner product to compute similarity.

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. $0 \leq J(\mathbf{q}, \mathbf{y}) \leq 1$.

Jaccard Similarity

Often vector embeddings used in semantic search are binary. For such vectors, Jaccard similarity is often used instead of Euclidean distance or inner product to compute similarity.

Definition (Jaccard Similarity)

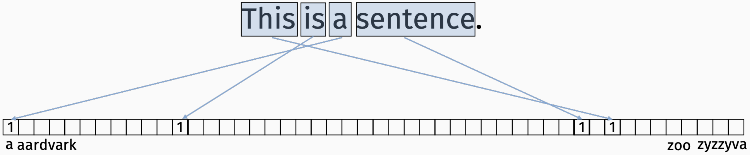
$$J(\mathbf{q}, \mathbf{y}) = \frac{|\mathbf{q} \cap \mathbf{y}|}{|\mathbf{q} \cup \mathbf{y}|} = \frac{\#\text{ of non-zero entries in common}}{\text{total }\#\text{ of non-zero entries}}$$

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

Example: $\mathbf{q} = [1, 0, 1, 0, 1], \mathbf{y} = [1, 1, 0, 0, 1]$. Then $J(\mathbf{q}, \mathbf{y}) = \frac{2}{4}$.

Jaccard Similarity for Document Comparison

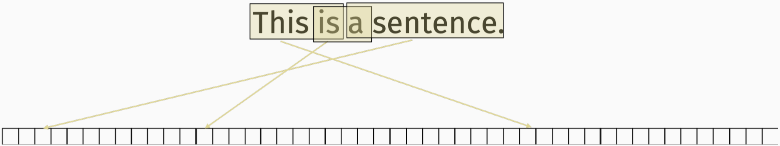
"Bag-of-words" model:



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.

Similarity Estimation

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

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

$\Downarrow$ C

.45	.68	.10	.92

Similarity Estimation

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

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

$$\Big\downarrow C$$

.45	.68	.10	.92

Use $C(\mathbf{q}), C(\mathbf{y})$ to approximately compute the Jaccard similarity $J(\mathbf{q}, \mathbf{y}) = \frac{|\mathbf{q} \cap \mathbf{y}|}{|\mathbf{q} \cup \mathbf{y}|}$.

MinHash

MinHash (Broder, '97):

- Choose k random hash functions $h_1, \ldots, h_k : \{1, \ldots, d\} \to [0, 1]$.

MinHash

MinHash (Broder, '97):

- Choose k random hash functions $h_1, \ldots, h_k : \{1, \ldots, d\} \rightarrow [0,1]$.
- For $i \in 1, \ldots, k$,
  - Let $c_i = \min_{j, \mathbf{q}_j = 1} h_i(j)$.

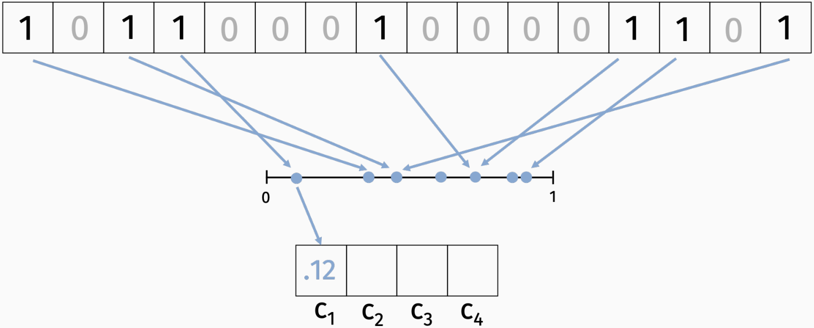
MinHash (Broder, '97):

- Choose k random hash functions $h_1, \ldots, h_k : \{1, \ldots, d\} \rightarrow [0,1]$.
- 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

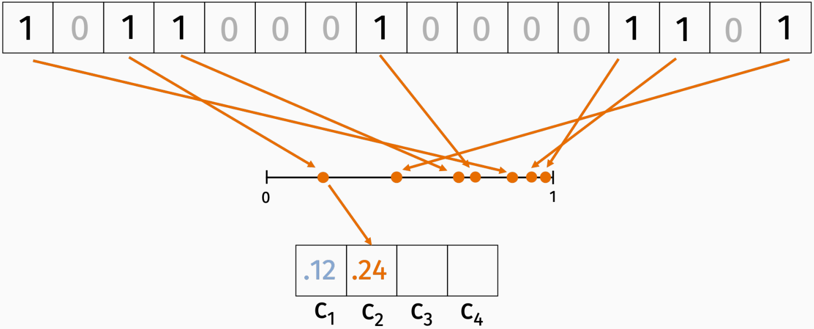
MinHash (Broder, '97):

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



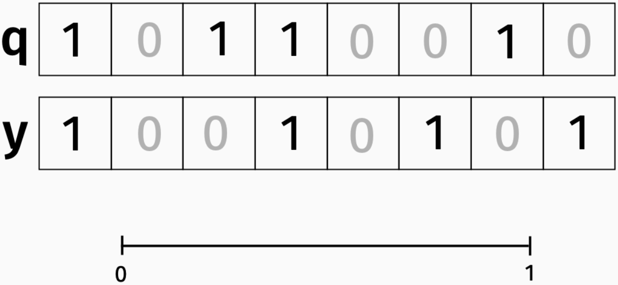
MinHash Analysis

Claim: For all i, $\Pr[c_i(\mathbf{q}) = c_i(\mathbf{y})] = J(\mathbf{q}, \mathbf{y}) = \frac{|\mathbf{q} \cap \mathbf{y}|}{|\mathbf{q} \cup \mathbf{y}|}$.

q	1	0	1	1	0	0	1	0
y	1	0	0	1	0	1	0	1

MinHash Analysis

Claim: For all i, $\Pr[c_i(\mathbf{q}) = c_i(\mathbf{y})] = J(\mathbf{q}, \mathbf{y}) = \frac{|\mathbf{q} \cap \mathbf{y}|}{|\mathbf{q} \cup \mathbf{y}|}$.



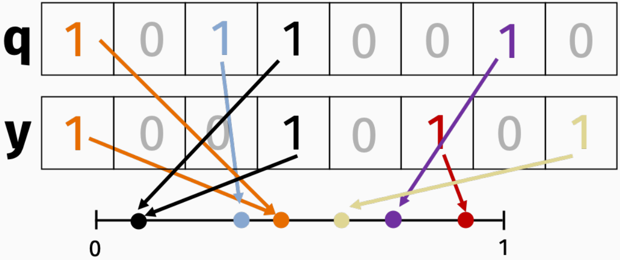
Proof:

1. For $c_i(\mathbf{q}) = c_i(\mathbf{y})$, we need that

$$\arg\min_{i:\mathbf{q}_i=1} h(i) = \arg\min_{i:\mathbf{y}_i=1} h(i)$$

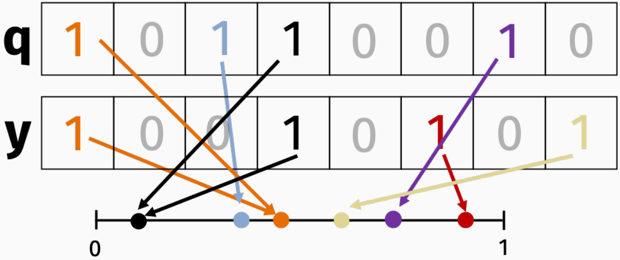
MinHash Analysis

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



MinHash Analysis

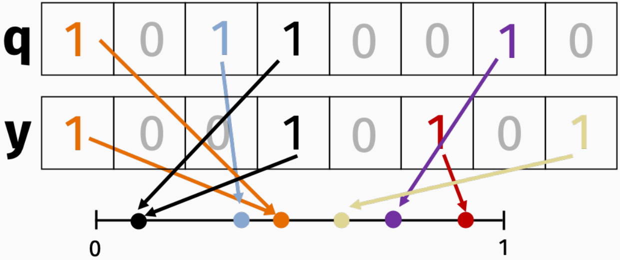
Claim: $\Pr[c_i(\mathbf{q}) = c_i(\mathbf{y})] = J(\mathbf{q}, \mathbf{y})$.



- Number of unique locations on the line: $|\mathbf{q} \cup \mathbf{y}|$.

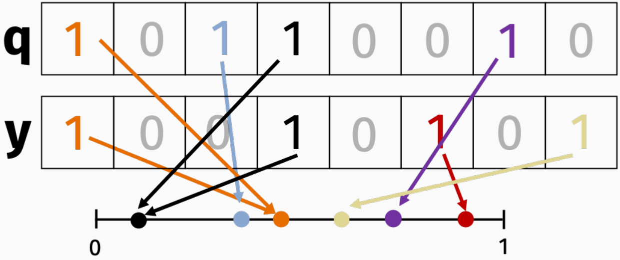
MinHash Analysis

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



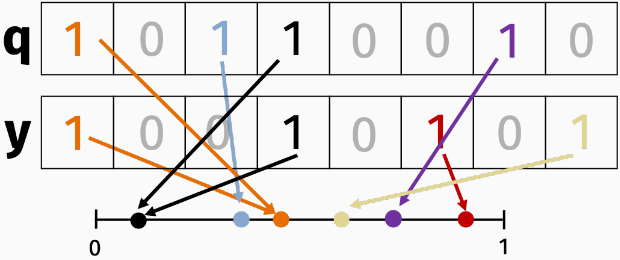
- Number of unique locations on the line: $|\mathbf{q} \cup \mathbf{y}|$.
- Number of colliding balls: $|\mathbf{q} \cap \mathbf{y}|$.

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



MinHash Analysis

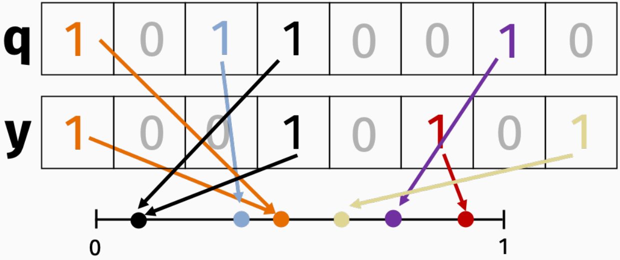
Claim: $\Pr[c_i(\mathbf{q}) = c_i(\mathbf{y})] = J(\mathbf{q}, \mathbf{y})$.



2. Every colliding ball is equally likely to produce the lowest hash value. So:

MinHash Analysis

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



2. Every colliding ball is equally likely to produce the lowest hash value. So:

$$\Pr[c_i(\mathbf{q}) = c_i(\mathbf{y})] = \frac{\#\text{ of colliding balls}}{\#\text{ of distinct balls}} = \frac{|\mathbf{q} \cap \mathbf{y}|}{|\mathbf{q} \cup \mathbf{y}|} = J(\mathbf{q}, \mathbf{y})$$

MinHash Analysis

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

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

Example:

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

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

$$\tilde{J} =$$

MinHash Analysis

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

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

Example:

C(**q**)

.12	.24	.76	.35

C(**y**)

.12	.98	.76	.11

$$\tilde{J} = 0.5$$

Unbiased estimate for Jaccard similarity:

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

MinHash Analysis

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

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

Example:

C(**q**)

.12	.24	.76	.35

C(**y**)

.12	.98	.76	.11

$$\tilde{J} = 0.5$$

Unbiased estimate for Jaccard similarity:

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

The more repetitions, the lower the variance.

MinHash Analysis

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

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

Observe,

$$\mathbb{1}[c_i(\mathbf{q}) = c_i(\mathbf{y})] = \begin{cases} 1 & \text{with probability } J \\ 0 & \text{with probability } 1 - J \end{cases}$$

MinHash Analysis

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

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

Observe,

$$\mathbb{1}[c_i(\mathbf{q}) = c_i(\mathbf{y})] = \begin{cases} 1 & \text{with probability } J \\ 0 & \text{with probability } 1 - J \end{cases}$$

$$\mathrm{Var}[\tilde{J}] = \frac{1}{k^2}\sum_{i=1}^{k} \mathrm{Var}[\mathbb{1}[c_i(\mathbf{q}) = c_i(\mathbf{y})]]$$

MinHash Analysis

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

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

Observe,

$$\mathbb{1}[c_i(\mathbf{q}) = c_i(\mathbf{y})] = \begin{cases} 1 & \text{with probability } J \\ 0 & \text{with probability } 1 - J \end{cases}$$

$$\begin{aligned} \mathrm{Var}[\tilde{J}] &= \frac{1}{k^2}\sum_{i=1}^{k} \mathrm{Var}[\mathbb{1}[c_i(\mathbf{q}) = c_i(\mathbf{y})]] \\ &= \frac{1}{k^2}\sum_{i=1}^{k} J - J^2 \leq \frac{1}{k} \end{aligned}$$

MinHash Analysis

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

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

$$\mathbb{E}[\tilde{J}] = J, \qquad \mathrm{Var}[\tilde{J}] \leq \frac{1}{k}$$

How large does k need to be so that with probability $> 1 - \delta$, $|J - \tilde{J}| \leq \epsilon$?

MinHash Analysis

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

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

$$\mathbb{E}[\tilde{J}] = J, \qquad \text{Var}[\tilde{J}] \leq \frac{1}{k}$$

How large does k need to be so that with probability $> 1 - \delta$, $|J - \tilde{J}| \leq \epsilon$?

$$\Pr[|J - \tilde{J}| \geq \alpha\sigma] \leq \frac{1}{\alpha^2}$$

MinHash Analysis

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

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

$$\mathbb{E}[\tilde{J}] = J, \qquad \mathrm{Var}[\tilde{J}] \leq \frac{1}{k}$$

How large does k need to be so that with probability $> 1 - \delta$, $|J - \tilde{J}| \leq \epsilon$?

$$\begin{aligned} &\Pr[|J - \tilde{J}| \geq \alpha\sigma] \leq \frac{1}{\alpha^2} \\ \implies &\Pr[|J - \tilde{J}| \geq \alpha/\sqrt{k}] \leq \frac{1}{\alpha^2} \qquad \text{(set } 1/\alpha^2 = \delta\text{)} \end{aligned}$$

MinHash Analysis

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

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

$$\mathbb{E}[\tilde{J}] = J, \qquad \text{Var}[\tilde{J}] \leq \frac{1}{k}$$

How large does k need to be so that with probability $> 1 - \delta$, $|J - \tilde{J}| \leq \epsilon$?

$$\Pr[|J - \tilde{J}| \geq \alpha\sigma] \leq \frac{1}{\alpha^2}$$

$$\implies \Pr[|J - \tilde{J}| \geq \alpha/\sqrt{k}] \leq \frac{1}{\alpha^2} \qquad \text{(set } 1/\alpha^2 = \delta)$$

$$\implies \Pr[|J - \tilde{J}| \geq 1/\sqrt{\delta k}] \leq \delta \qquad \text{(set } \sqrt{\frac{1}{\delta k}} = \epsilon)$$

MinHash Analysis

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

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

$$\mathbb{E}[\tilde{J}] = J, \qquad \mathrm{Var}[\tilde{J}] \leq \frac{1}{k}$$

How large does k need to be so that with probability $> 1 - \delta$, $|J - \tilde{J}| \leq \epsilon$?

$$\Pr[|J - \tilde{J}| \geq \alpha\sigma] \leq \frac{1}{\alpha^2}$$

$$\implies \Pr[|J - \tilde{J}| \geq \alpha/\sqrt{k}] \leq \frac{1}{\alpha^2} \qquad \text{(set } 1/\alpha^2 = \delta\text{)}$$

$$\implies \Pr[|J - \tilde{J}| \geq 1/\sqrt{\delta k}] \leq \delta \qquad \text{(set } \sqrt{\frac{1}{\delta k}} = \epsilon\text{)}$$

MinHash Analysis

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

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

$$\mathbb{E}[\tilde{J}] = J, \qquad \text{Var}[\tilde{J}] \le \frac{1}{k}$$

How large does k need to be so that with probability $> 1 - \delta$, $|J - \tilde{J}| \le \epsilon$?

$$\Pr[|J - \tilde{J}| \ge \alpha\sigma] \le \frac{1}{\alpha^2}$$

$$\implies \Pr[|J - \tilde{J}| \ge \alpha/\sqrt{k}] \le \frac{1}{\alpha^2} \qquad \text{(set } 1/\alpha^2 = \delta)$$

$$\implies \Pr[|J - \tilde{J}| \ge 1/\sqrt{\delta k}] \le \delta \qquad \text{(set } \sqrt{\frac{1}{\delta k}} = \epsilon)$$

Suffices to set $k = O\left(\frac{1}{\epsilon^2 \delta}\right)$.

MinHash Analysis

Conclusion: If $k = \Theta\left(\frac{1}{\epsilon^2 \delta}\right)$, then with prob. $1 - \delta$,

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

MinHash Analysis

Conclusion: If $k = \Theta\left(\frac{1}{\epsilon^2\delta}\right)$, then with prob. $1 - \delta$,

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

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

MinHash Analysis

Conclusion: If $k = \Theta\left(\frac{1}{\epsilon^2 \delta}\right)$, then with prob. $1 - \delta$,

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

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

Can be improved to $\log(1/\delta)$ dependence?