Tuesday, March 10

Midterm 1

Minimum:	52
Median:	82
Mean:	78
Maximum:	96
Standard deviation:	14

Grades in canvas!

Long tail, will curve eg. 60% to C

Plan

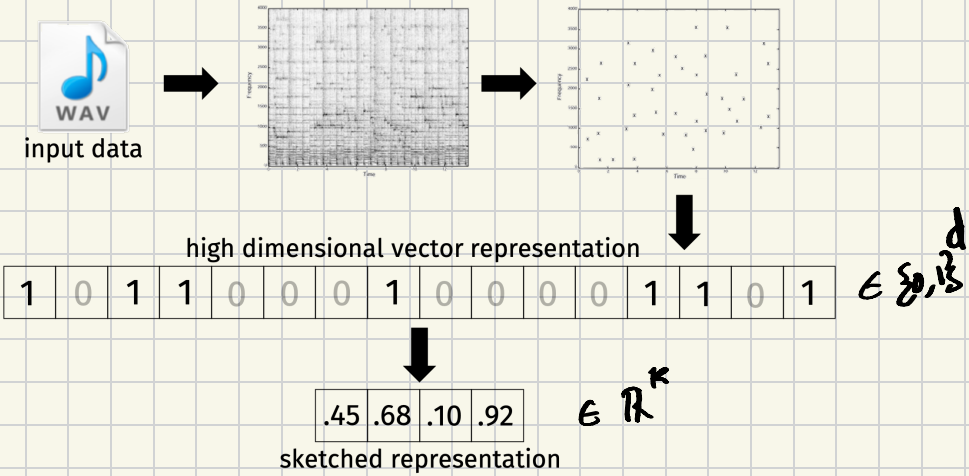
- Similarity estimation
- Locality sensitive Hashing

Similarity Estimation

Q: How does Shazam match song clips to a library of n = tens of millions of songs in a fraction of a second?

↑ background noise

↑ song clips offset



Query song $x \in \{0,1\}^d$

Goal: Find "nearby" $y \in \{0,1\}^d$

Challenge:

- $O(nd$ bits$)$
- $O(d)$ time to compare q, y

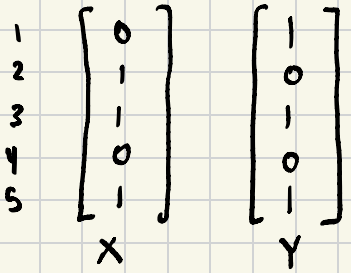
Building Block:

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

to preserve Jaccard similarity

Jaccard Similarity

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



↑ vector representation

← set representation

$\{2,3,5\}$ $\{1,3,5\}$

Applications

- "bag of words"
- earthquakes
- cached webpages

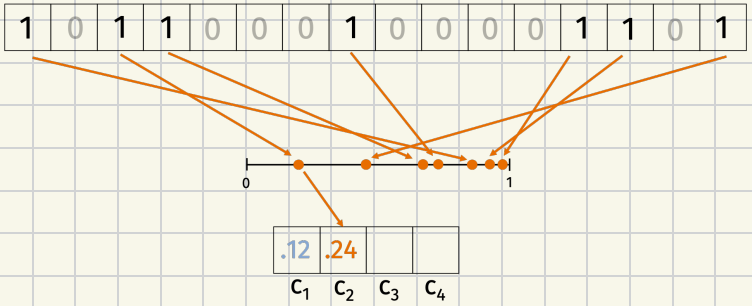
Min Hash Algorithm

$X \in \{0,1\}^d$

Compression function $C: \{0,1\}^d \to \mathbb{R}^k$

k Hash functions

$$C_i(x) = \min_{j \in [d]: x_j = 1} h_i(j)$$



Claim: $\Pr(C_i(x) = C_i(y)) = J(x,y) = \frac{|x \cap y|}{|x \cup y|}$

$$\Rightarrow \hat{J}(x,y) = \frac{1}{k} \sum_{i=1}^{k} \mathbb{1}[C_i(x) = C_i(y)]$$

Claim: When $k = O(\frac{1}{\epsilon^2 \delta})$, w.p. $1-\delta$,

$$J(x,y) - \epsilon \leq \hat{J}(x,y) \leq J(x,y) + \epsilon$$

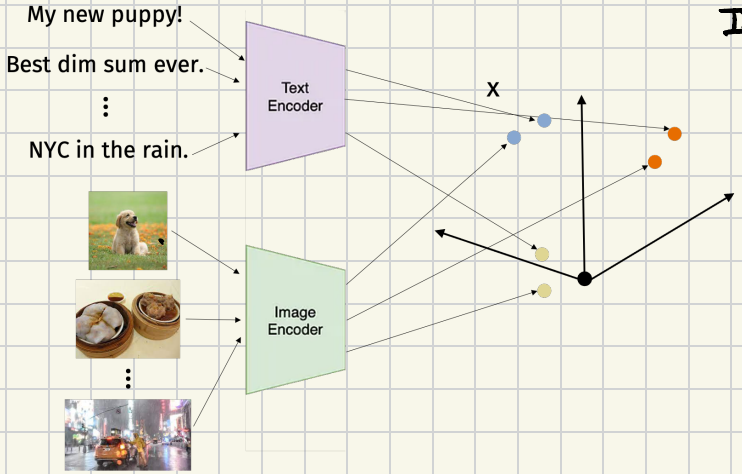
Near Neighbor Search

Goal: Given $x \in \mathbb{R}^d$, find "nearest" vectors among $y_1, \ldots, y_n \in \mathbb{R}^d$

Using MinHash, storage / time go from $O(nd)$ to $O(nk)$.

↑ still linear!

Is faster possible?



Important problem for multimodal embeddings!

Locality Sensitive Hashing

$h : \mathbb{R}^d \to \{1, \ldots, m\}$

h is "locally sensitive" if

- $Pr(h(x)=h(y))$ is high when x,y close
- $Pr(h(x)=h(y))$ is low when x,y far

Attempt #0:

$c : \{0,1\}^d \to [0,1]$ minhash

$g : [0,1] \to \{1, \ldots, m\}$ uniform hash

$h(x) = g(c(x))$

$Pr(h(x)=h(y)) =$

Preprocessing

Select h.

Instantiate table with $m = O(n)$ cells.

Put y_i in cell $h(y_i)$ $\forall i$

Query

Given x, search in cell $h(x)$ for near neighbors.

Two issues:

1. False negative rate

2. False positive rate

Attempt #1: Increase FNR

Idea: More chances to find y

t tables with $h_1, \ldots, h_t$ Preprocess and query in all t

$\Pr(\text{find } y) =$

Attempt #2: Reduce FPR

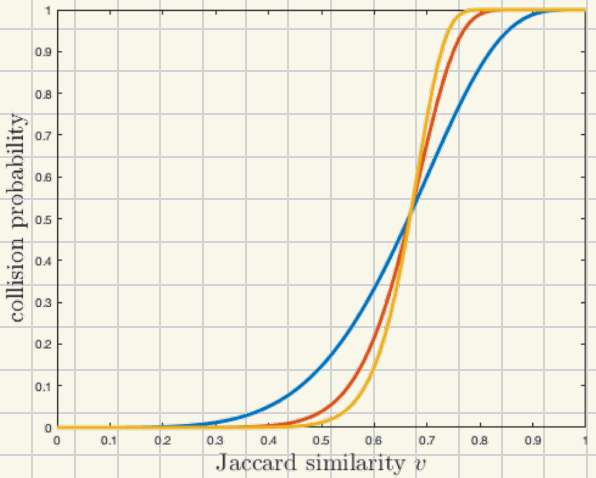
Idea: Harder to appear in the same cell

r copies of min hash. $h_i(x) = g(c_1(x), \ldots, c_r(x))$

$$\Pr(h(x) = h(y)) = J(x,y)^r + (1 - J(x,y)^r)\frac{1}{m}$$

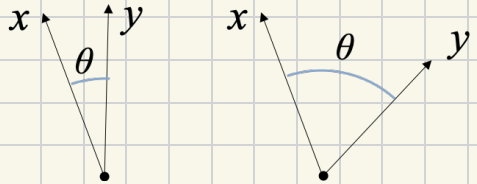
$\Pr(\text{find } y) =$

Tuning



Desmos with r and t!

MinHash works for binary, what about similarity between general vectors $x, y \in \mathbb{R}^d$?



$$\cos\theta = \frac{\langle x, y \rangle}{\|x\|_2 \|y\|_2}$$

$$\|x-y\|_2^2 = \langle x-y, x-y \rangle$$

$$= \|x\|_2^2 - 2\langle x, y \rangle + \|y\|_2^2$$

unit norm

$$= 2 - 2\cos\theta$$

↑ "inverse" of distance

Thursday, March 12

- No OH Monday 3/23
- Written notes are not ready!

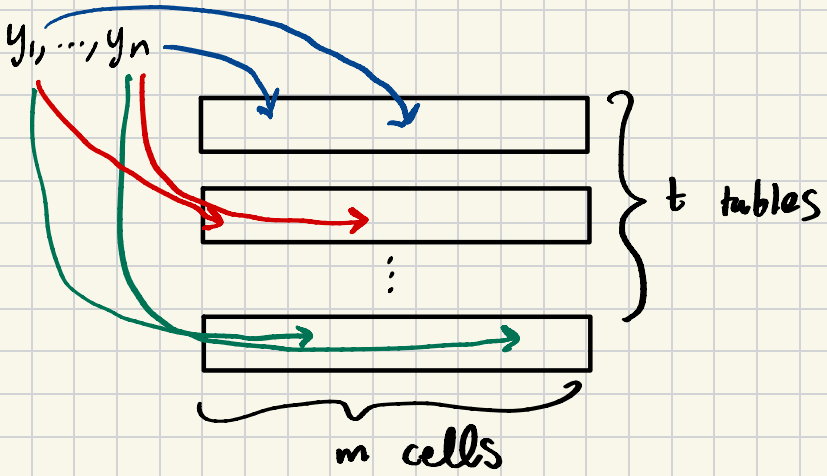
Plan

- SimHash
- Singular Value Decomposition

Locality Sensitive Hashing

$h_1, \dots, h_t : \mathbb{R}^d \to \{1, \dots, m\}$

$$\Pr(h_i(x) = h_i(y)) = \begin{cases} \text{high} & \text{if } x, y \text{ close} \\ \text{low} & \text{otherwise} \end{cases}$$



Given x,

$$\Pr(\text{find } y) = 1 - \Pr(h_i(x) \neq h_i(y))^t$$

Jaccard

$c_j^{(i)} : \{0,1\}^d \to [0,1]$ Min Hash

$f^{(i)} : [0,1]^d \to \{1, \dots, m\}$

$$h_i(x) = f^{(i)}\left(c_1^{(i)}(x), \dots, c_r^{(i)}(x)\right)$$

$$\Pr(h_i(x) = h_i(y)) \approx J(x,y)^r$$

when $m = O(n)$

SimHash

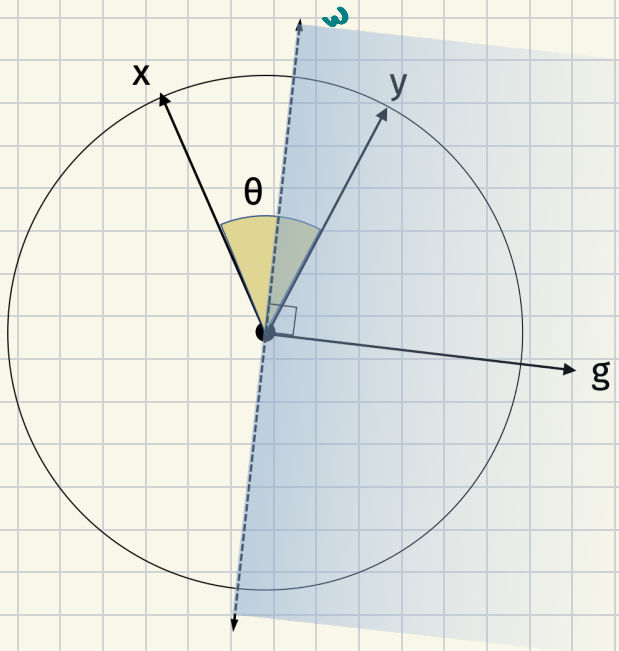
Let $g_1, \ldots, g_r \sim \mathcal{N}(0, I)$

$f : \{-1, 1\}^r \to \{1, \ldots, m\}$ uniform random

$h_i(x) = f([\text{sign}(\langle g_1, x \rangle), \ldots, \text{sign}(\langle g_r, x \rangle)])$

$$\Pr_g(\text{sign}(\langle g, x \rangle) = \text{sign}(\langle g, y \rangle)) = 1 - \frac{\theta}{\pi}$$

↑ Same sign iff x, y on the same side of hyperplane w



$\Pr(h_i(x) = h_i(y)) =$

Linear Algebra Review

Suppose $X \in \mathbb{R}^{d \times d}$ is symmetric

$Xv = \lambda v$ (λ: eigenvalue, v: eigenvector)

If full rank,

$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$

$v_1, v_2, \ldots, v_d \in \mathbb{R}^d$

v_i are orthonormal:

$$\langle v_i, v_j \rangle = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

$$X = \sum_{i=1}^{d} \lambda_i v_i v_i^T$$

Easy to see $X v_j = \lambda_j v_j$

Frobenius Norm

$$\|X\|_F^2 = \sum_{i=1}^{d} \sum_{j=1}^{d} ([X]_{i,j})^2$$

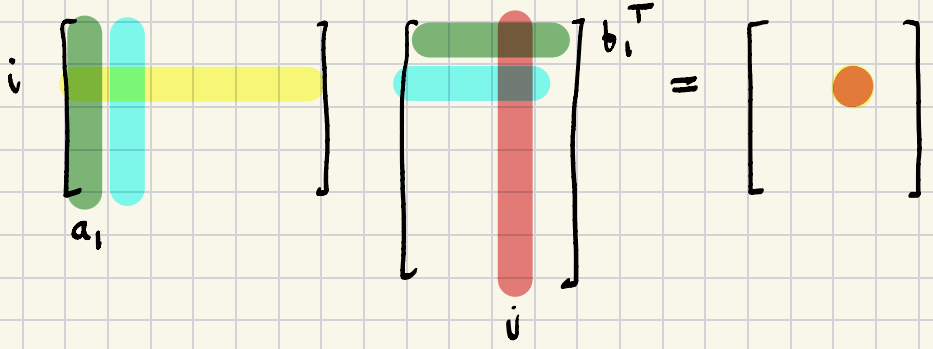
$$= tr(X^T X)$$

$$= tr\left(\sum_{i=1}^{d} \lambda_i v_i v_i^T \sum_{j=1}^{d} \lambda_j v_j v_j^T \right)$$

$$= tr\left(\sum_{i=1}^{d} \lambda_i^2 v_i v_i^T \right)$$

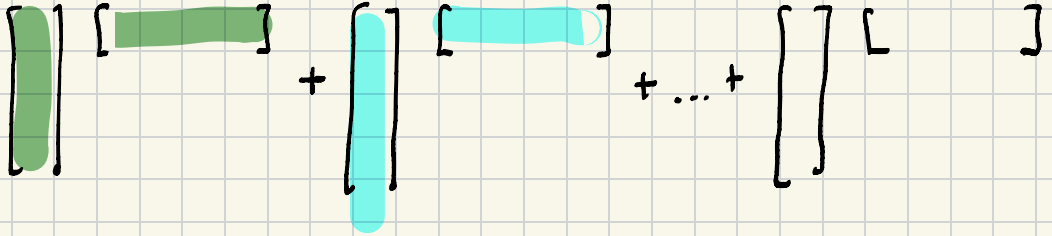
Matrix Multiplication

$A \in \mathbb{R}^{m \times n}$ $B \in \mathbb{R}^{n \times m}$



$$[AB]_{i,j} = \sum_{k=1}^{n} [A]_{i,k} [B]_{k,j}$$

$$= \sum_{k=1}^{n} a_k b_k^T$$

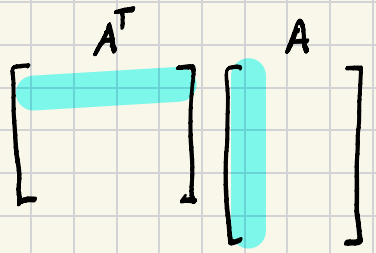


Frobenius Norm

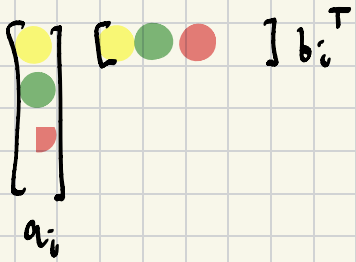
$A \in \mathbb{R}^{m \times n}$

$$\|A\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} [A]_{i,j}^2 = \sum_{j=1}^{n} \sum_{i=1}^{m} [A^T]_{j,i} [A]_{i,j}$$

$$= \sum_{j=1}^{n} [A^T A]_{j,j} = \text{trace}(A^T A)$$



Cyclic Property of the Trace



$A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times m}$

$$AB = \sum_{i=1}^{m} a_i b_i^T$$

$$\text{tr}(AB) = \sum_{i=1}^{m} \text{tr}(a_i b_i^T) = \sum_{i=1}^{m} b_i^T a_i = \sum_{i=1}^{n} [BA]_{i,i} = \text{tr}(BA)$$

Eigendecomposition

Square, symmetric $X \in \mathbb{R}^{d \times d}$

$Xv = \lambda v$ (λ: eigenvalue, v: eigenvector)

$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$

$v_1, v_2, \ldots, v_d \in \mathbb{R}^d$

$$\langle v_i, v_j \rangle = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

$$X = \sum_{i=1}^{d} \lambda_i v_i v_i^T$$

$XX =$

$X^T X =$

Singular Vector Decomposition

Any matrix $X \in \mathbb{R}^{n \times d}$

WLOG, $n \geq d$

Singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_d$

Left singular vectors $u_1, u_2, \ldots, u_d \in \mathbb{R}^n$

Right singular vectors $v_1, v_2, \ldots, v_d \in \mathbb{R}^d$

$$\langle u_i, u_j \rangle = \langle v_i, v_j \rangle = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

$$X = \sum_{i=1}^{d} \sigma_i u_i v_i^T$$

$XX =$

$X^T X =$

$$Xa = \sum_{i=1}^{d} u_i \sigma_i v_i^T a$$

1. Project vector onto $v_1, \ldots, v_d$
2. Scale coordinates
3. Linear combination of $u_1, \ldots, u_d$

SVD useful for...

- Pseudo inverse $X^+ = \sum_{i=1}^{d} \frac{1}{\sigma_i} v_i u_i^T$
- Condition number σ_1 / σ_d
- Matrix norms i.e., $\|X\|_2 = \sigma_1$, $\|X\|_F^2 = \sum_{i=1}^{d} \sigma_i^2$
- Principal Component Analysis
- Powers $X^p = \sum_{i=1}^{d} \sigma_i^p u_i v_i^T$

Low Rank Approximation

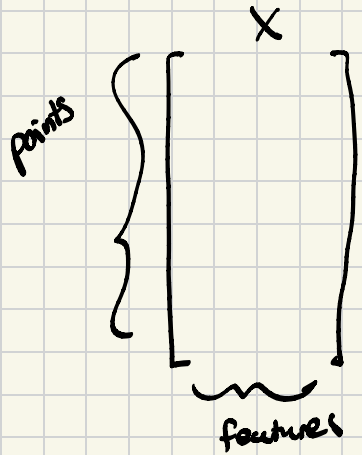
Rank characterizes structure:

↳ rank k if k unique points

↳ $\approx$ rank k if k clusters of points

↳ rank k if k indep. features

$\Rightarrow$ reduce dimension using structure!



Goal: Given X, best rank-k approximation

$$X_k = \text{argmin} \; \| X - XWW^T \|_F^2$$

XWW^T where $W \in \mathbb{R}^{d \times k}$ with orthonormal columns i.e. $W^TW = I$

Eckart-Young-Mirsky: $X_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$

$$(*) = \|X - XWW^T\|_F^2 = \|X(I - WW^T)\|_F^2$$

$$= tr\left((I-WW^T)X^TX(I-WW^T)\right)$$

$$= tr\left(X^TX(I-WW^T)^2\right)$$

$$= tr(X^TX) - tr(X^TXWW^T)$$

$(I - WW^T) = (I-WW^T)^T = (I-WW^T)^2$

cyclic property of the trace

linearity of trace

$$\arg\min_W (*) = \arg\max_W tr(X^TXWW^T)$$

$$tr(X^TXWW^T) = tr(W^TX^TXW)$$

$$= tr\left(W^T \sum_{i=1}^{d} v_i \sigma_i^2 v_i^T W\right)$$

$$= \sum_{i=1}^{d} \sigma_i^2 \, tr(W^T v_i v_i^T W)$$

$$= \sum_{i=1}^{d} \sigma_i^2 \|v_i^T W\|_2^2$$

$$X^TX = \sum_{i=1}^{d} \sigma_i v_i u_i^T \sum_{j=1}^{d} \sigma_j u_j v_j^T$$

$z_i = \|v_i^T W\|_2^2$ $\quad 0 \le z_i \le 1$ because columns of W are orthonormal

$$\sum_{i=1}^{d} z_i = \sum_{i=1}^{d} tr(W^T v_i v_i^T W) = tr(W^T \sum_{i=1}^{d} v_i v_i^T W)$$

$$= tr(W^T I_{d\times d} W) = tr(I_{k\times k}) = k$$

Choose z_i with $0 \le z_i \le 1$ and $\sum_{i=1}^{d} z_i = k$ to maximize $\sum_{i=1}^{d} \sigma_i^2 z_i$

$\therefore$ Choose $z_i = \begin{cases} 1 & \text{if } i \le k \\ 0 & \text{else} \end{cases}$. $W = \sum_{i=1}^{k} v_i e_i^T$

$$X W W^T = \sum_{i=1}^{d} \sigma_i u_i v_i^T \sum_{j=1}^{k} v_j e_j^T \sum_{l=1}^{k} e_l v_l^T$$

$$= \sum_{i=1}^{d} \sigma_i u_i v_i^T \sum_{j=1}^{k} v_j v_j^T = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$