Plan

Logistics

Review

Similarity Estimation

Thanks for coming!

Tea time @ 2 6th floor

JL Lemma

$x_1, \ldots, x_n \in \mathbb{R}^d$ $\quad k = O\left(\frac{\log n}{\epsilon^2}\right)$

$\Pi \in \mathbb{R}^{k \times d}$ $\quad \Pi_{i,j} =$ random variable

$(1-\epsilon) \|x_i - x_j\|_2^2$

$\leq \|\Pi x_i - \Pi x_j\|_2^2$

$\leq (1+\epsilon) \|x_i - x_j\|_2^2$ wp $9/10$

$x = x_i - x_j$ and union bound

Distributional JL

$k = O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$

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

Proved with clever tricks :)

wp $1-\delta$

what about inner product?

$|\langle x_i, x_j \rangle - \langle \Pi x_i, \Pi x_j \rangle| \leq \frac{\epsilon}{2}(\|x_i\|_2^2 + \|x_j\|_2^2)$

Using JL Lemma!

Application: Fast Set Join estimation

X = people in class

Y = people who climb

$$x = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{matrix} \leftarrow \text{Sujay} \\ \\ \leftarrow \text{Iris} \\ \end{matrix} \qquad y = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \begin{matrix} \leftarrow \text{Iris} \\ \\ \leftarrow \text{Aidan} \end{matrix}$$

$\langle x, y \rangle = |X \cap Y|$

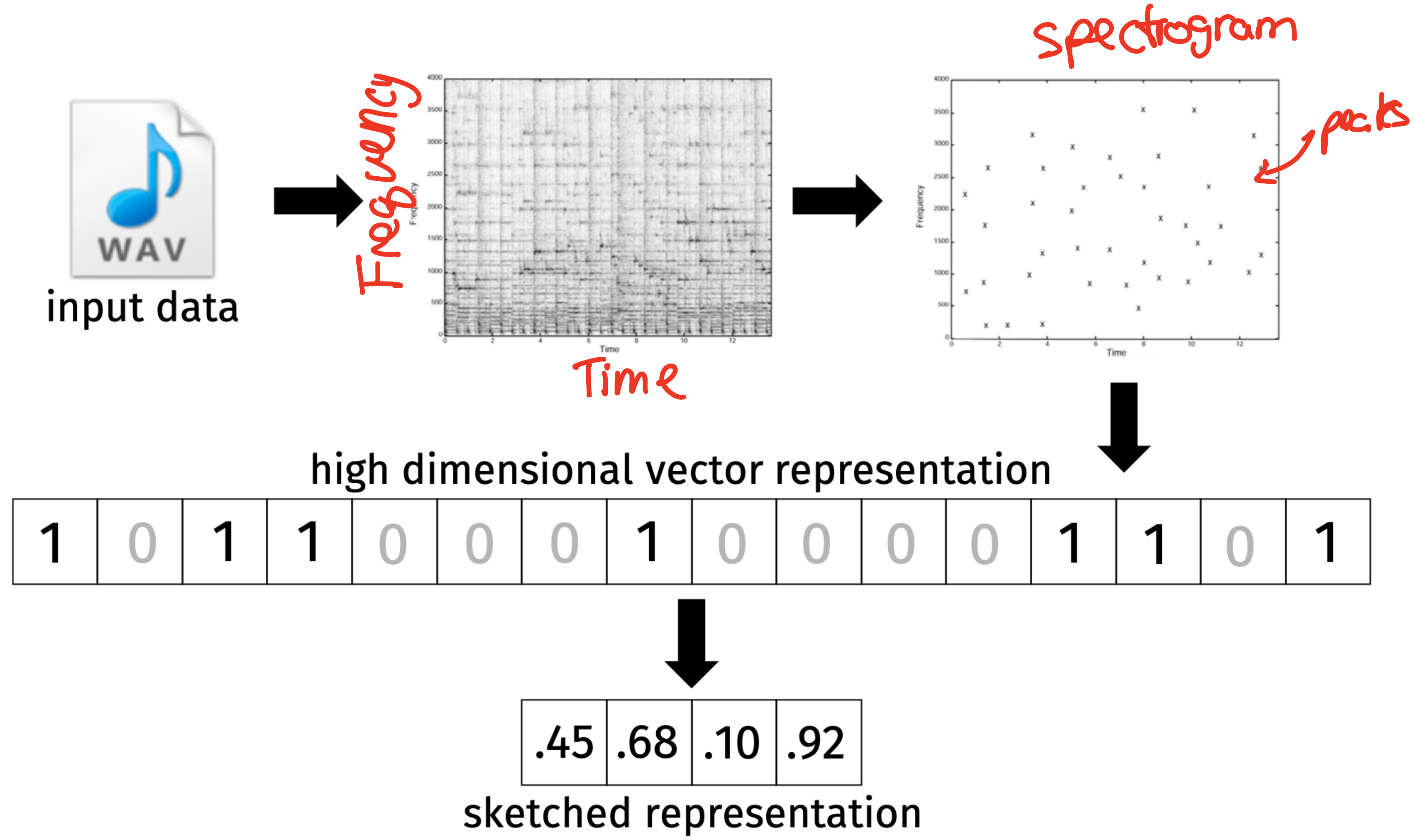
$\|x\|_2^2 = |X|$

estimate with $\Pi x, \Pi y$

Similarity Estimation

JL preserves distance,
how about "similarity"?

Shazam matches short,
noisy (hehe) clips against
huge database



Problem: Given query $q \in \mathbb{R}^d$, find similar song $y \in \mathbb{R}^d$

With n songs, $O(nd)$ space and $O(nd)$ naive search

"Sketch" into Lower Dimension

Want $c(q) \in \mathbb{R}^k$ for $k \ll d$

$C(x) \approx C(y)$ when $x \approx y$

How do we quantify?

Jaccard Similarity

$$J(x,y) = \frac{|X \cap Y|}{|X \cup Y|}$$

$$= \frac{\#\text{non-zero in common}}{\#\text{non-zero total}}$$

e.g. $X = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ $\quad y = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

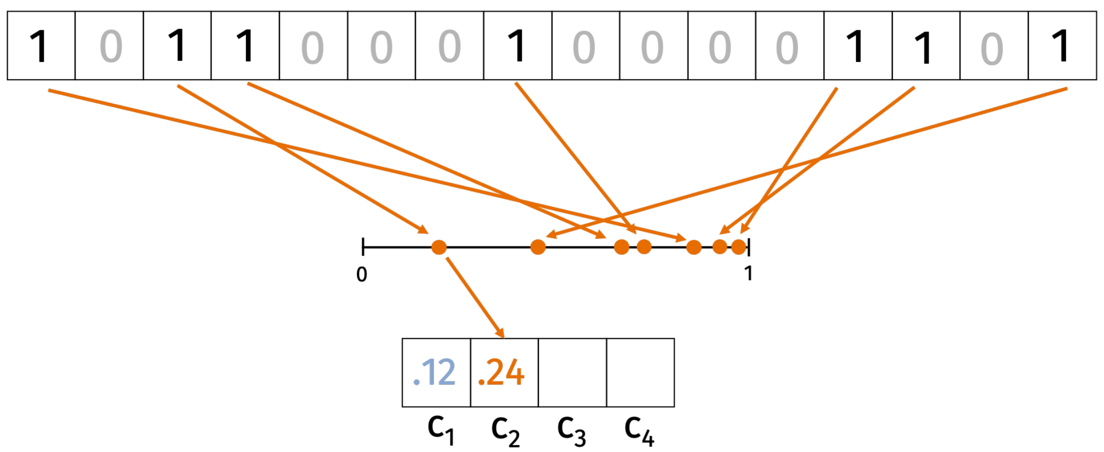
$J(X,y) = ?$

Also useful for

- "bag of word" documents
- cached webpages
- earthquake detection

Min Hash

$$c : \{0,1\}^d \to \mathbb{R}^k$$



$$c_i = \min_{j \in \{1,\dots,d\} : x_j = 1} h_i(j)$$

$$\Pr(c_i(x) = c_i(y)) = \ ?$$

Estimate $J(x,y)$ using c_i

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

$$= \frac{1}{k} \langle c(x), c(y) \rangle$$

$$\mathbb{E}[\hat{J}(x,y)] = \ ?$$

$$\mathrm{Var}(\hat{J}(x,y)] \leq \ ?$$

Chebyshev's

$$\Pr\left(\left| \hat{J}(x,y) - \mathbb{E}[J(x,y)] \right| \geq \alpha \sqrt{\mathrm{Var}(\hat{\tilde{J}}(x,y))} \right) \leq \frac{1}{\alpha^2}$$

$$\epsilon \overset{\text{want}}{=} \alpha \sqrt{\mathrm{Var}(\hat{\tilde{J}}(x,y))} \qquad \frac{1}{\alpha^2} \overset{\text{want}}{=} \delta$$

$$k = ?$$

$$J(x,y) - \epsilon \leq \hat{J}(x,y) \leq \hat{J}(x,y) + \epsilon \quad \text{wp } 1/\delta$$

$\underline{\text{using biased coin theorem}}$

- ↳ heads if $c_i(x) = c_i(y)$
- ↳ bias $b = J(x,y)$

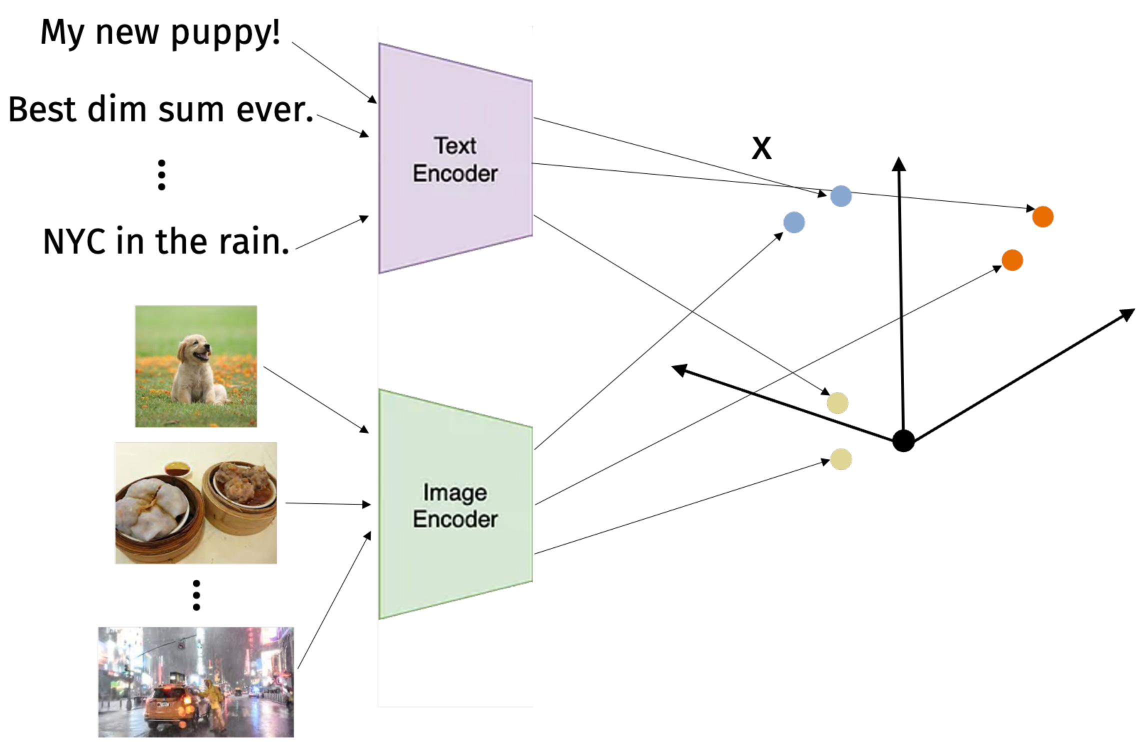
Get $k = O\left(\frac{\log 1/\delta}{\epsilon^2}\right)$

$O(d) \rightarrow O(k)$ compute (approx) similarity

$O(dn) \rightarrow O(kn)$ naive search

How do we find similar points faster?

Useful for CLIP



Locality Sensitive Hashing

- Hash function $h: \{0,1\}^d \to \{1,\dots,m\}$
- Similitarity function S e.g. Jaccard
- h is *locally sensitive* if

$$Pr(h(x)=h(y)) = \begin{cases} \text{large} & \text{when } x \simeq y \\ \text{small} & \text{when } x \not\simeq y \end{cases}$$

Our approach:

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

$g: \mathbb{R} \to \{1,\dots,m\}$ uniform hash function

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

$h(x) = h(y)$ when

(1) $c(x) = c(y)$ or

(2) $c(x)$, $c(y)$ happen to hash to same cell

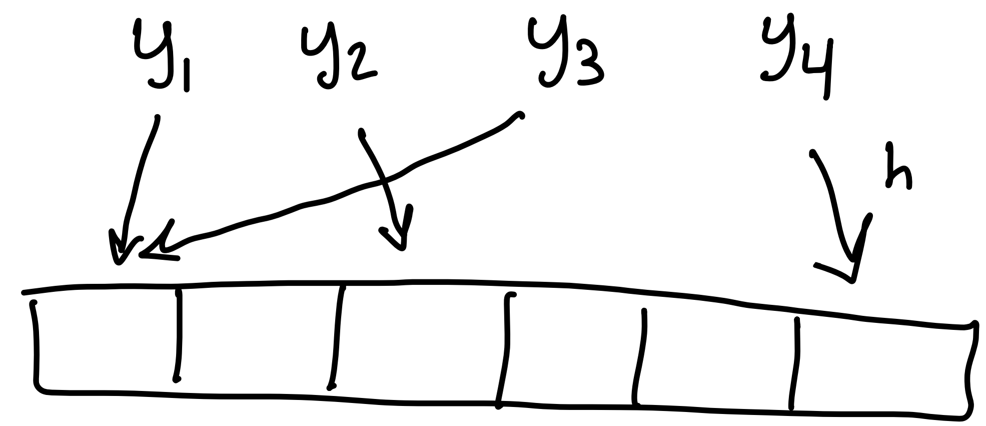
$$\begin{aligned} &Pr(h(x) = h(y)) \\ &= Pr(c(x)=c(y)) \cdot 1 \\ &+ Pr(c(x) \neq c(y)) \cdot \frac{1}{m} \\ &\leq J(x,y) + \frac{1}{m} \end{aligned}$$

Preprocessing

Choose h in terms of g, c

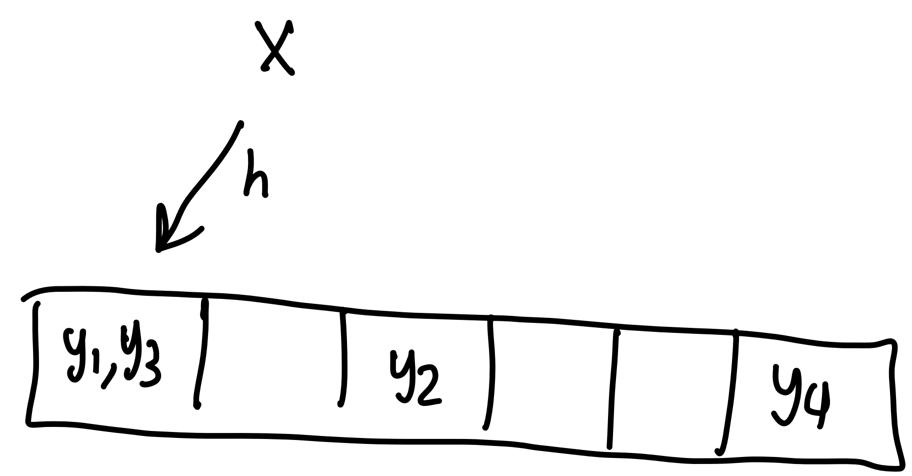
Create a table with m slots

For each vector (song), we compute $h(y)$ and store in corresponding slot



Query

Compute $h(x)$ and look in corresponding cell for similar songs



Repeat with t tables

Two questions:

↳ False negative: What's the probability we don't find a close vector?

↳ False positive: What's the probability we find a far vector?

$$\Pr(\text{find } y) = 1 - \Pr(y \text{ not in table})^t = 1 - \Pr(h_i(x) \neq h_i(y))^t$$

$$= 1 - (1 - J(x,y))^t$$

when $J(x,y) = .4$ and $t = 10$, $\Pr(\text{find } y) = 1 - (1 - .4)^{10} \approx .99$ ☺

when $J(x,y) = .2$ and $t = 10$, $\Pr(\text{find } y) = 1 - (1 - .2)^{10} \approx .89$ ☹

Our approach:

$c_1, \dots, c_r \; : \{0,1\}^d \to [0,1]$ single MinHash

$g : [0,1]^r \to \{1, \dots, m\}$ uniform hash function

$h(x) = g(c_1(x), c_2(x), \dots, c_r(x))$

$$\Pr(h(x)=h(y)) \le \Pr(c_1(x)=c_1(y), \dots, c_r(x)=c_r(y)) + 1 \cdot \frac{1}{m}$$

$$\overset{\text{indep}}{=} \Pr(c_i(x)=c_i(y))^r + \frac{1}{m}$$

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

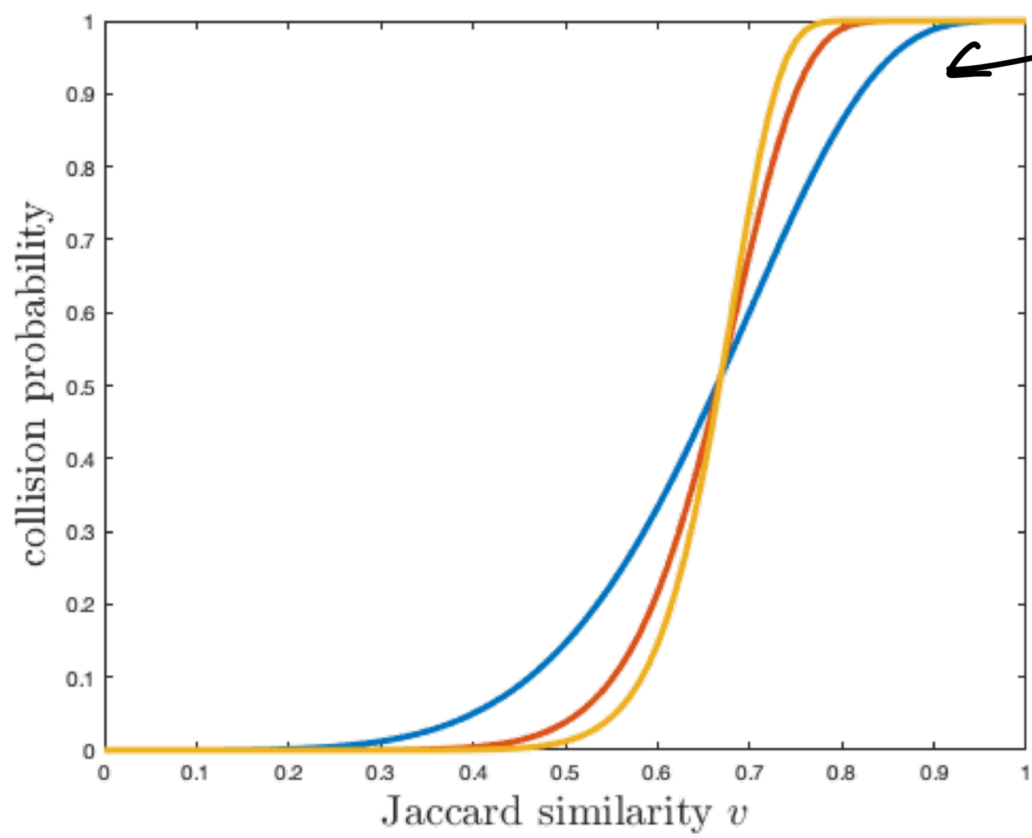
$$\Pr(\text{find } y) = 1 - \Pr(y \text{ not in table})^t = 1 - \Pr(h_i(x) \neq h_i(y))^t$$

$$= 1 - (1 - J(x,y)^r)^t$$

$$\Pr(\text{find } y) = 1 - (1 - J(x,y)^r)^t$$

when $J(x,y) = .4$ and $t = 10$, $r = 2$ $\quad \Pr(\text{find } y) = 1 - (1 - .4^2)^{10} \approx .83$:)

when $J(x,y) = .2$ and $t = 10$, $r = 2$ $\quad \Pr(\text{find } y) = 1 - (1 - .2^2)^{10} \approx .33$:(



different r and t