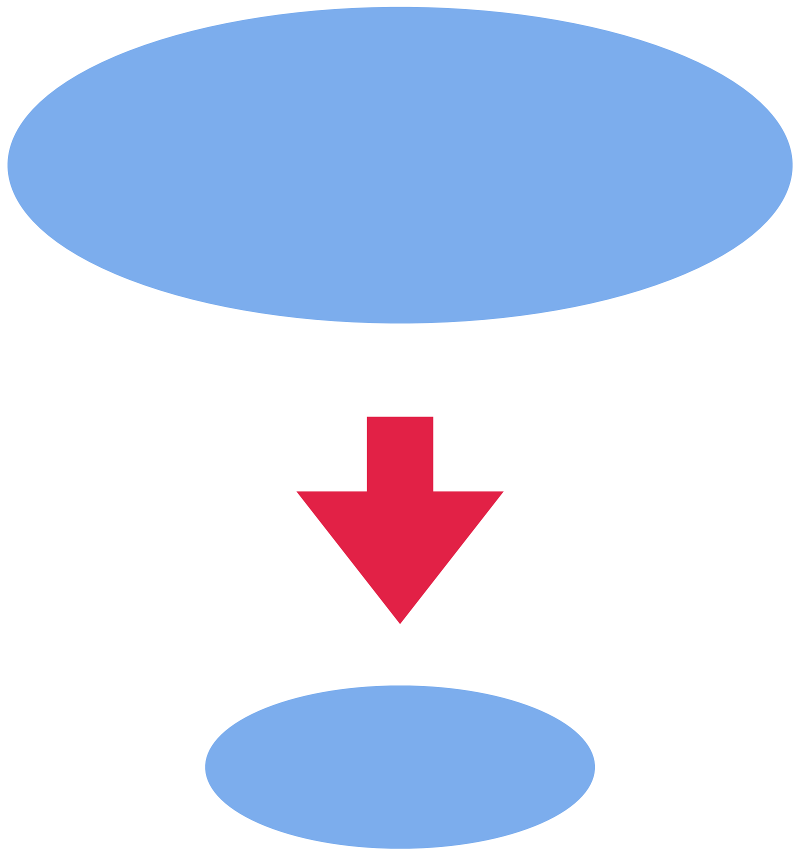
CSE 566 Spring 2023

Locality Sensitive Hashing

Instructor: Mingfu Shao

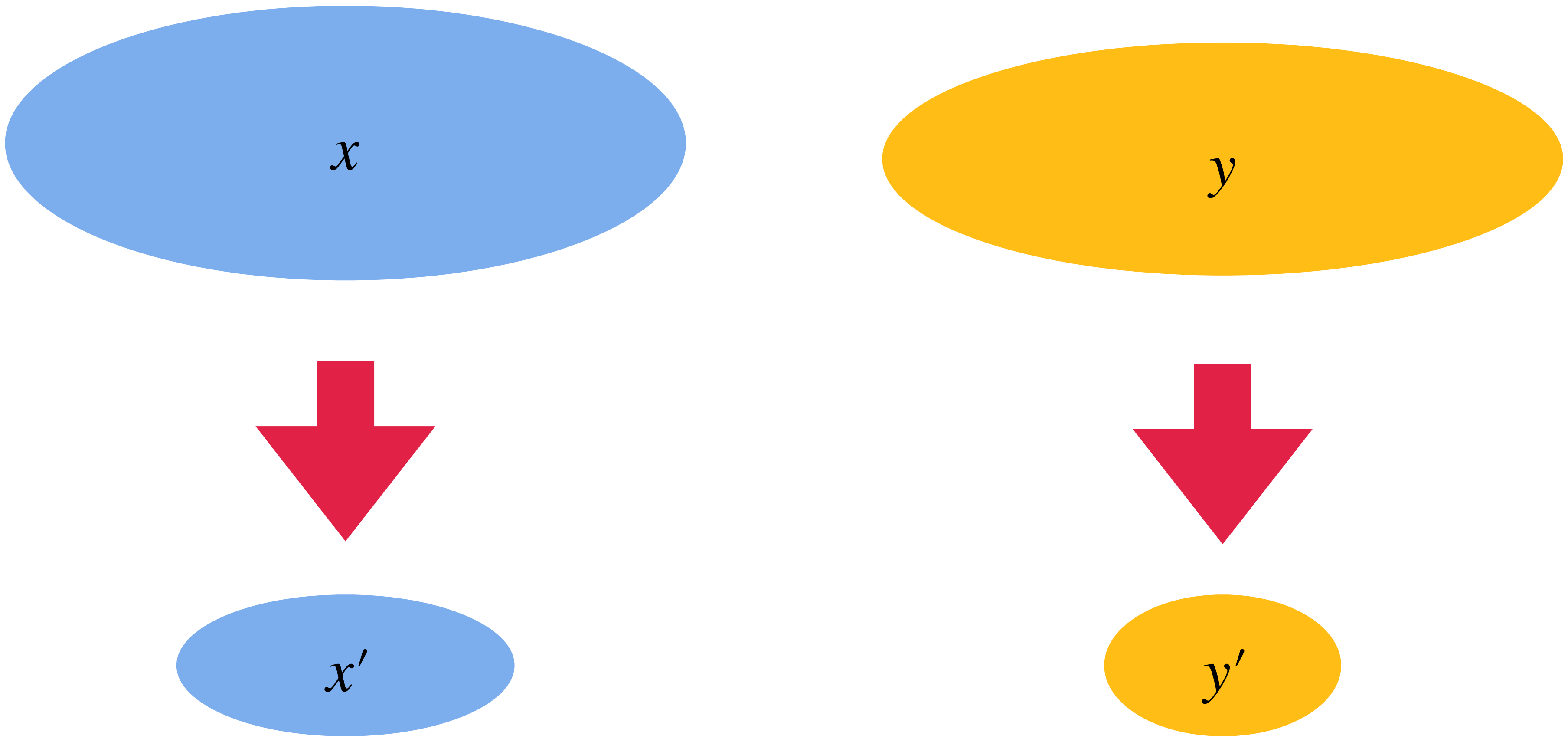
Sketching / Fingerprinting

- Extract a sketch/fingerprint of small size, that is "representative" of the original, large-scale data.



Locality Sensitive Hashing (LSH)

- If x and y are similar, then x' and y' are similar.
- If x and y are dissimilar, then x' and y' are dissimilar.



Formal Definition

- A set of hash functions $\mathcal{F}$ is a *locality sensitive hash (LSH) family* for similarity measure $s(\cdot,\cdot)$ if for any x and y we have $\Pr_{f\in\mathcal{F}}(f(x) = f(y)) = s(x, y)$.
- The randomness comes for picking f from $\mathcal{F}$ uniformly at randomly.

Hamming Similarity

- Hamming distance between $x, y \in \{0,1\}^n$ is defined as the number of locations where $x_i \neq y_i$, $d(x, y) = |\{i \mid x_i \neq y_i\}|$.
- Hamming similarity: $h(x, y) = 1 - d(x, y)/n$.

$x = 0111011$ $\quad f_2(x) = 1.$

$y = 1010101$ $\quad f_2(y) = 0$

$d(x,y) = 5$, $h(x,y) = 1 - 5/7 = \frac{2}{7}.$

LSH Family for Hamming Similarity

$x \in \{0,1\}^n$, $1 \le i \le n$.

- Define hash function $f_i(x) = x_i$; define $\mathcal{F} = \{f_1, f_2, \cdots, f_n\}$.
- **Fact**: $\mathcal{F} = \{f_1, \cdots, f_n\}$ is a LSH family for hamming similarity.
- **Proof**: to prove that $\Pr_{f \in \mathcal{F}}(f(x) = f(y)) = h(x, y)$ for every two binary vectors $x, y \in \{0,1\}^n$.

$x = 0\,1\,1\,1\,0\,1\,1$

$y = 1\,0\,1\,0\,1\,0\,1$

$$\Pr(f(x) = f(y)) = \frac{n - d(x,y)}{n}$$

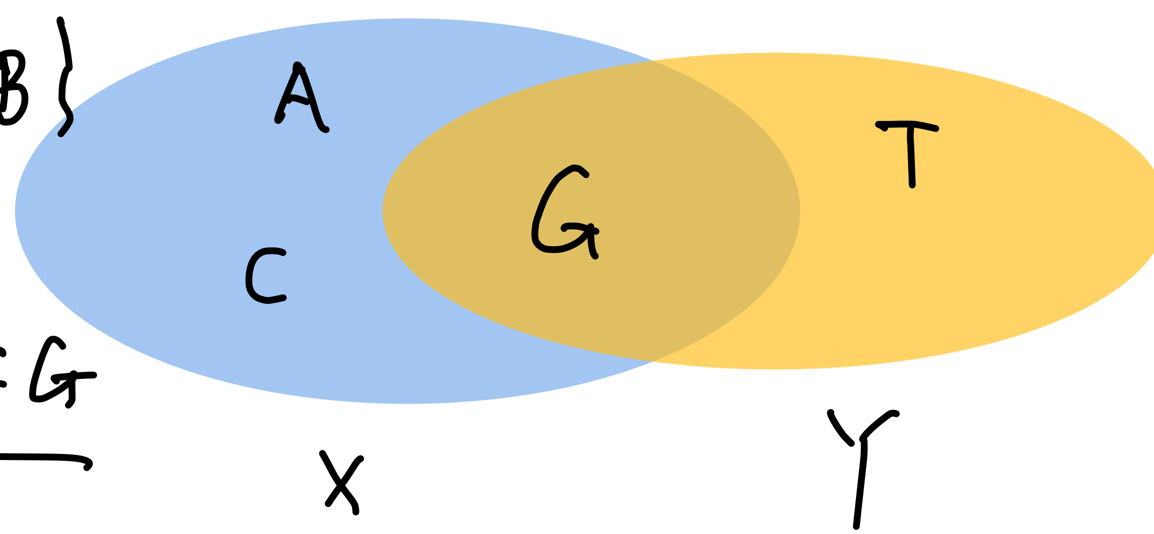
$$= h(x,y)$$

Jaccard Similarity

- The Jaccard similarity between two sets X and Y, where X and Y are subsets of U, is defined as

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

$U = \{A, C, G, T, B\}$



$J(X, Y) = \frac{1}{4}$

$\pi: \underline{C < A < B < T < G}$

$f_\pi(X) = C, \quad f_\pi(Y) = T.$

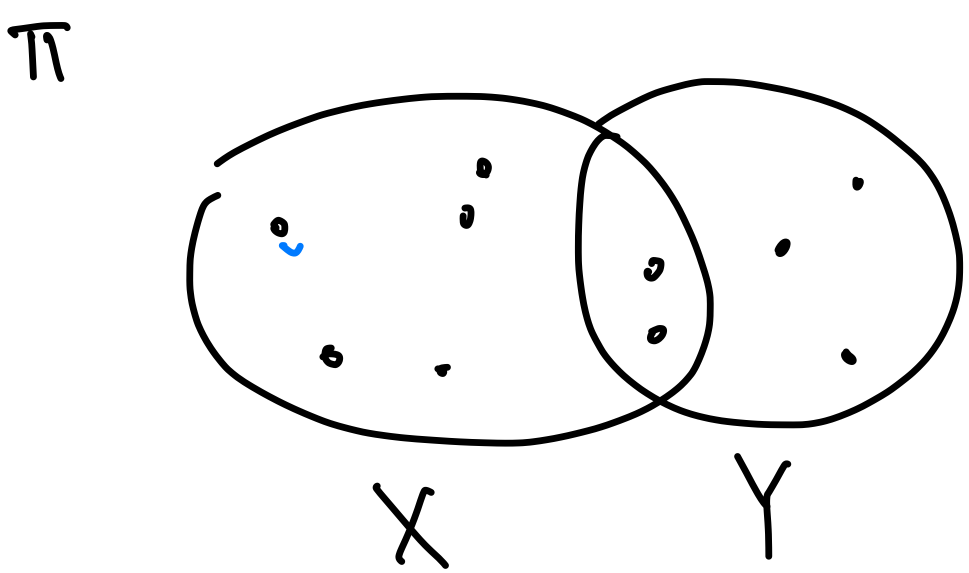
LSH Family for Jaccard Similarity

- Let π be a permutation/order of U. Define function $f_\pi(X)$ maps X to the smallest element in X, i.e., $f_\pi(X) = \arg\min_{x \in X} \pi(x)$.
- Let Π be the set of all possible permutations over U. Define $\mathcal{F} = \{f_\pi \mid \pi \in \Pi\}$.

$|\pi| = |U|!$

LSH Family for Jaccard Similarity

- **Fact**: $\mathscr{F} = \{f_\pi \mid \pi \in \Pi\}$ is a LSH family for Jaccard similarity.
- **Proof**: to prove that $\Pr_{\pi \in \Pi}(f_\pi(X) = f_\pi(Y)) = J(X, Y)$ for every two sets $X, Y \subset U$.
  - For a fixed π, $f_\pi(X) = f_\pi(Y)$ iff $f_\pi(X \cup Y) \in X \cap Y$.
  - For each $a \in X \cup Y$, $\Pr(f_\pi(X \cup Y) = a) = 1/|X \cup Y|$.



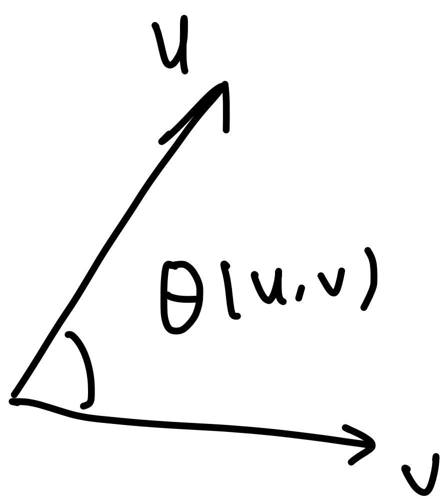
if $m = f_\pi(X \cup Y) \in X \cap Y$

$\Rightarrow f_\pi(X) = m,\ f_\pi(Y) = m$

$$\Pr(f_\pi(x) = f_\pi(y)) = \frac{|X \cap Y|}{|X \cup Y|}$$

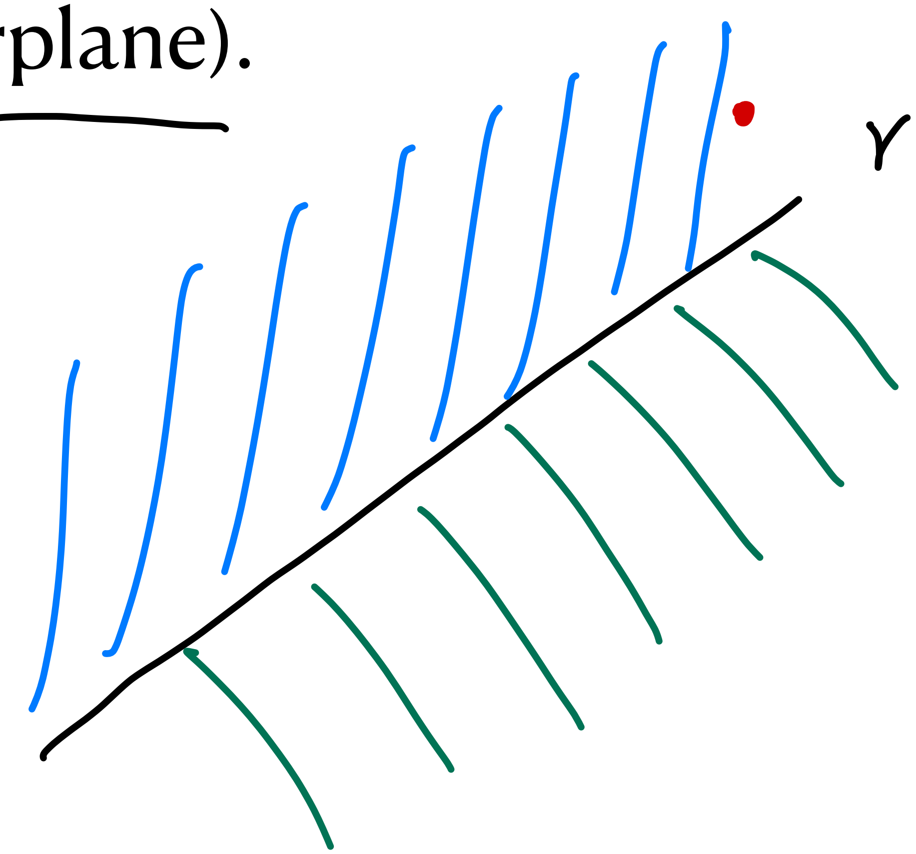
Angular Similarity

- $\theta(u, v)$: the angle between vectors u and v, where $u, v \in \mathbb{R}^d$.
- Angular Similarity: $1 - \theta(u, v)/\pi$.



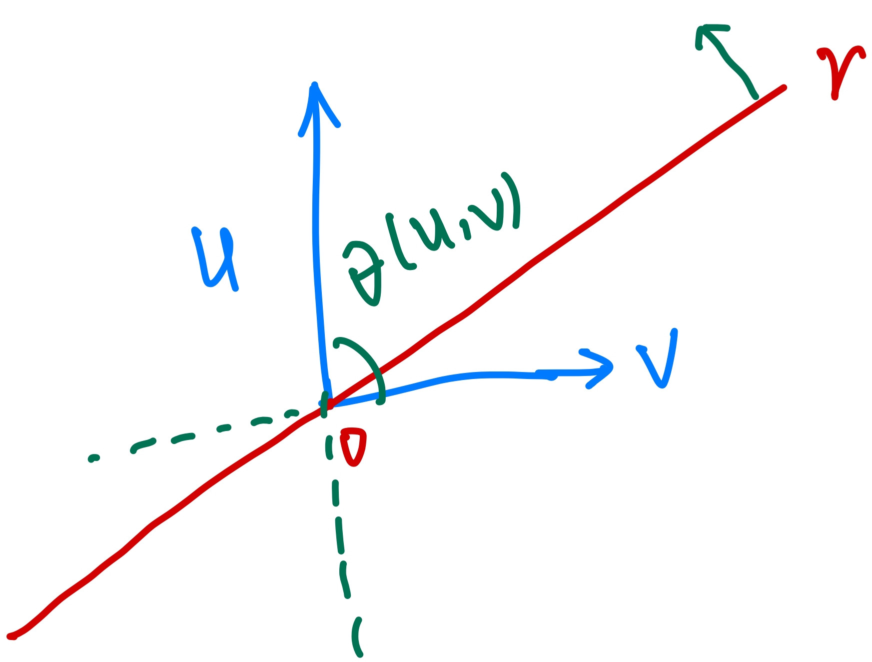
LSH Family for Angular Similarity

- Let $r \in \mathbb{R}^d$ be a vector (aka hyperplane).
- Define function $f_r(u)$: $u \in \mathbb{R}^d$
  - $f_r(u) = 1$, if $u \cdot r \geq 0$
  - $f_r(u) = 0$, if $u \cdot r < 0$
- Define $\mathscr{F} = \{f_r \mid r \in \mathbb{R}^d\}$.



LSH Family for Angular Similarity

- **Fact**: $\mathscr{F} = \{f_r \mid r \in \mathbb{R}^d\}$ is a LSH family for angular similarity.
- **Proof**: to prove that $\Pr_{r \in \mathbb{R}^d}(f_r(u) = f_r(v)) = 1 - \theta(u, v)/\pi$ for every two vectors $u, v \in \mathbb{R}^d$.
    - The probability that a random hyperplane splits vectors u and v is $\theta(u, v)/\pi$, i.e., $\Pr_{r \in \mathbb{R}^d}(f_r(u) \neq f_r(v)) = \theta(u, v)/\pi$.



Sketching using LSH

- **Approach**: randomly pick k hash functions $f_1, \cdots, f_k$ from $\mathcal{F}$, which transform x into $sketch(x) := (f_1(x), f_2(x), \cdots, f_k(x))$.
    - Sketching (large) binary data x: pick k random positions of x, and transform x into a list of k numbers. (random sampling)
    - Sketching (large) set X: pick k random orderings, and transform X into a list of k elements. (min-hash)
    - Sketching (long) vector u: pick k random hyperplanes, and transform u into a (binary) vector of size k. (random projection)

Estimating Similarity

- Consider sketching x and y, with the same random functions:
  - $sketch(x) := (f_1(x), f_2(x), \cdots, f_k(x))$
  - $sketch(y) := (f_1(y), f_2(y), \cdots, f_k(y))$
- Let Z_i be the random variable indicating if $f_i(x) = f_i(y)$. $1 \le i \le k$
- Let $Z := (\sum_{i=1}^{k} Z_i)/k$ be the percentage of "hash-collisions".
- $\mathbb{E}(Z_i) = \Pr(Z_i = 1) = \Pr(f_i(x) = f_i(y)) = s(x, y)$.
- $\mathbb{E}(Z) = \mathbb{E}(\sum_{i=1}^{k} Z_i)/k = s(x, y)$.

$E(Z_i) = Pr(Z_i = 1) \cdot 1 + Pr(Z_i = 0) \cdot 0$

$Var(Z)$. chernoff bound $\rightarrow$ $Pr(Z > (1+\delta) \cdot s(x,y))$

Nearest Neighbor Search

- **Problem**: find the element in $X = \{x_1, x_2, \cdots, x_n\}$ that is nearest to the query q, i.e., $\arg\max_{x_i \in X} s(x_i, q)$.
- Search using LSH; assume $\mathcal{F}$ is the LSH family for $s(\cdot, \cdot)$
  - Draw k functions from $\mathcal{F}$;
  - Sketch each x_i; $sketch(x_i) := (f_1(x_i), f_2(x_i), \cdots, f_k(x_i)) \in \mathbb{Z}^k$
  - Put x_i into the **bucket** labeled by $sketch(x_i)$
  - Sketch q with the same functions: $sketch(q) := (f_1(q), \cdots, f_k(q))$
  - Only compare q with those in bucket $sketch(q)$
  - Repeat above procedure t times.