Locality Sensitive Hashing

Ravi Kumar
Google

Background

- Webscale data
- Efficient algorithms
- Small memory footprint
- Approximate answers suffice

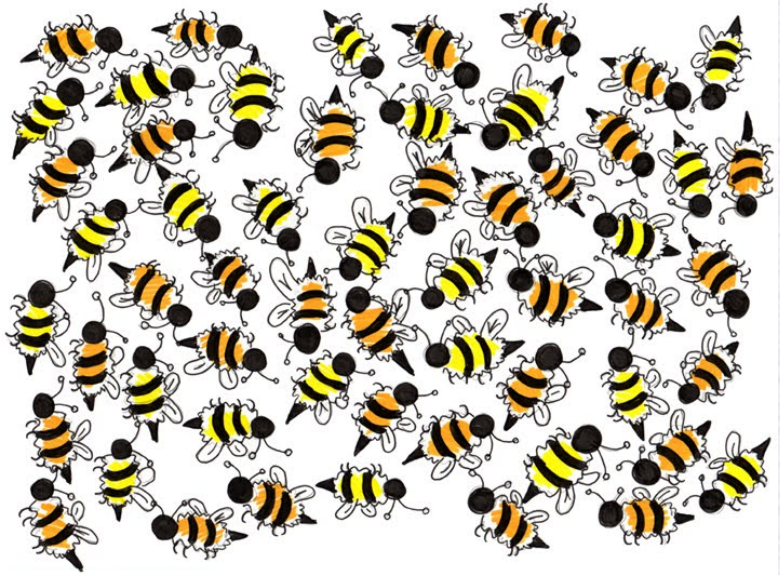
- Three example problems …

Duplicate detection

- Given a large set of documents, find the near-duplicates among them
  - Documents = text, web pages, homework solutions, papers by an author, …
  - Distance notion = Hamming distance, Jaccard coefficient, edit distance, …
- All-pair similarity is infeasible

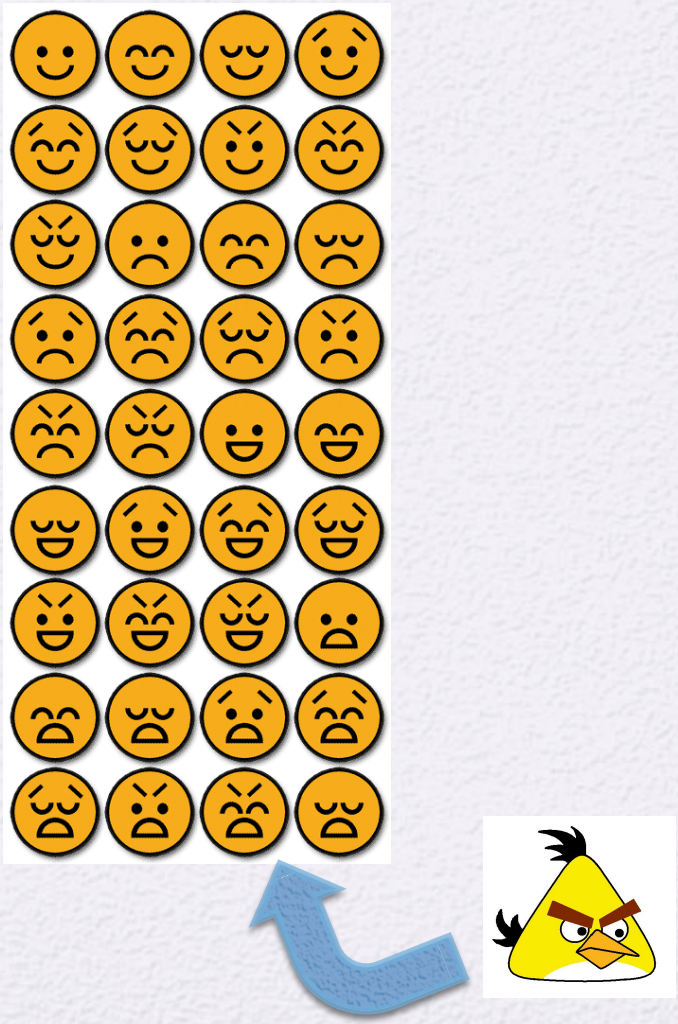
Clustering

- Given a set of objects with pairwise distances, group them into clusters
  - Objects = points in a metric space, documents, …
  - Distances = L_1, L_2, …
- Sub-quadratic solutions are desirable



Near neighbors

- Given a set of objects, preprocess them such that given a new object, one can find the closest in the set
  - Objects = documents, points, images, …
  - Distances = L_1, L_2, …
- Query time should be small



Common theme

- All these problems involve computing with object similarities
  - Try to avoid the quadratic blowup
- Question: Can we represent similarities between objects in a succinct manner?
  - As a function of a single object
    - By obtaining a "sketch" of the object
  - Sacrifice exactness for efficiency
    - By using randomization

LSH: Locality Sensitive Hashing C'02

- U = Universe of objects
- S: $U \times U \rightarrow [0, 1]$ = Similarity function

An LSH for a similarity S is a probability distribution over a set $\mathcal{H}$ of hash functions such that

$$Pr_{h \in \mathcal{H}}[h(A) = h(B)] = S(A, B)$$

for each $A, B \in U$

LSH (contd)

- LSHs represent similarities between objects using probability distributions over hash functions
  - Hash collision captures object similarity
- LSH has found uses in many applications
  - Near-duplicate detection
  - Near-neighbor search
  - Sketching
  - Clustering

LSH: Gap definition IMRS'97, IM'98, GIM'99

- S: $U \times U \rightarrow [0, 1]$ = Similarity function over a universe U of objects

An (r, R, p, P)-LSH for a similarity S is a probability distribution over a set $\mathcal{H}$ of hash functions such that

- $S(A, B) \geq R \Rightarrow Pr_{h \in \mathcal{H}}[h(A) = h(B)] > P$
- $S(A, B) < r \Rightarrow Pr_{h \in \mathcal{H}}[h(A) = h(B)] < p$

for each $A, B \in U$; here, $r < R$ and $P > p$

Original definition implies an (r, R, r, R) gap version

Eg 1. Hamming similarity

- Given two n-bit vectors x and y

$$HS(x, y) = \#\{ i : x_i = y_i \}/n$$

- Eg, disjoint vectors have similarity 0 and HS(x, x) = 1

x = 01001, y = 10011, HS(x, y) = 2/5

- 1 – HS(x, y) is the Hamming distance metric

Sampling hash IM'98

- $\mathcal{H} = \{h_1, \ldots, h_n\}$, where $h_i(x) = x_i$
  - The i-th hash function outputs the i-th bit of x

Claim. Sampling hash forms an LSH for Hamming similarity

$$\Pr[h(x) = h(y)] = \Pr_i[h_i(x) = h_i(y)] = HS(x, y)$$

Eg 2. Jaccard similarity

- Given two sets A and B

$$J(A, B) = |A \cap B| \ / \ |A \cup B|$$

- Eg, disjoint sets have similarity 0 and $J(A, A) = 1$

  $A = \{1, 2\}$, $B = \{2, 3\}$, $J(A, B) = 1/3$

- $1 - J(A, B)$ is a metric
- Used extensively in many scientific and sociological applications
- Paul Jaccard introduced this similarity in 1901 for comparing and clustering fields of flowers on the Alps

MinHash B'97, BCFM'00

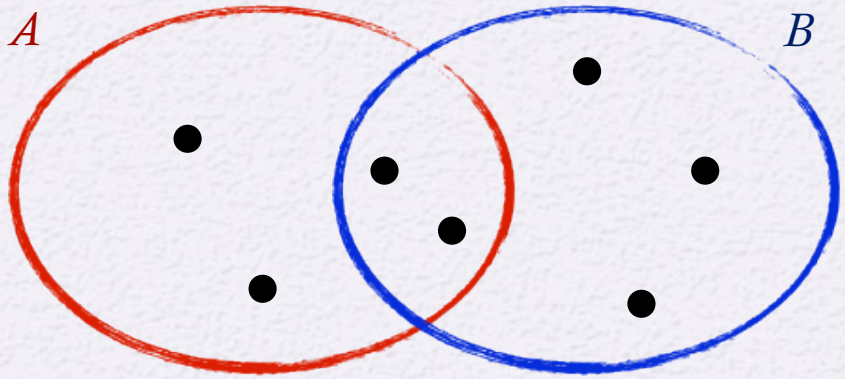
- Given a universe U, pick a permutation π on U uniformly at random
- Hash each subset $S \subseteq U$ to the minimum value it contains according to π
- Eg, A = {1, 2}, B = {2, 3}

$\pi = (1 < 2 < 3)$, h(A) = 1, h(B) = 2
$\pi = (1 < 3 < 2)$, h(A) = 1, h(B) = 3
$\pi = (2 < 1 < 3)$, h(A) = 2, h(B) = 2

$\pi = (2 < 3 < 1)$, h(A) = 2, h(B) = 2
$\pi = (3 < 1 < 2)$, h(A) = 1, h(B) = 3
$\pi = (3 < 2 < 1)$, h(A) = 2, h(B) = 3

MinHash (contd)

Claim. MinHash forms an LSH for Jaccard similarity



$$\Pr[h(A) = h(B)] = |A \cap B| \; / \; |A \cup B| = J(A, B)$$

Eg 3. Angle similarity

- Given two unit vectors x and y

  $\theta(x, y)$ = angle between x and y

- Natural measure of similarity for high-dimensional vectors
- Eg, $\theta(x, x) = 0$ and $\theta(x, y)$ maximum at $y = -x$

  $x = (\sqrt{3}/2, 1/2)$, $y = (1/\sqrt{2}, 1/\sqrt{2})$, $\theta(x, y) = \pi/12$

- Used extensively in text processing, machine learning applications

SimHash C'02

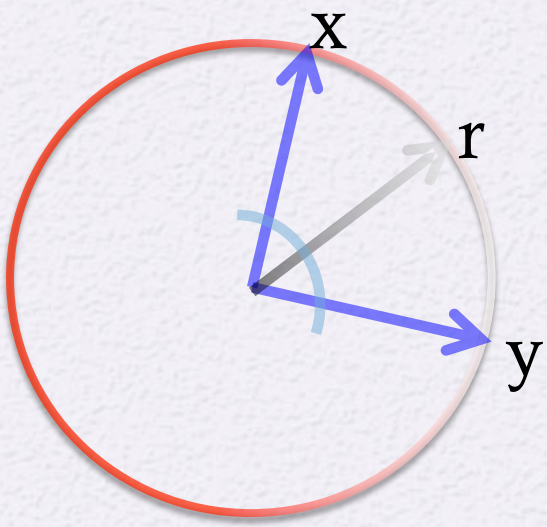
- Pick a random unit vector r
- Hash each vector x by computing $\text{sgn}\langle x, r\rangle$

Eg, $x = (\sqrt{3}/2, 1/2)$, $r = (0.41, -0.91)$, $h(x) = -0.1$

- Can also pick each entry of r from N(0, 1) and normalize

SimHash (contd)

Claim. SimHash forms an LSH for angle similarity



$$\Pr[h(x) = h(y)] = 1 - \theta(x, y)/\pi$$

A different set similarity measure: if x and y are characteristic vectors $\theta = \arccos(|A \cap B| / (\sqrt{A}\sqrt{B}))$

Eg 4, Gap Hamming KOR'98

- Let k, ϵ be given
- Given two n-bit vectors x and y

  $H_{k,\epsilon}(x, y)$ = 0 if $H(x, y) < k$

  = 1 if $H(x, y) > (1+\epsilon)\, k$
- Biased projection: choose independently

  r_i = 0 wp $1 - 1/(4k)$

  = 1 wp $1/(4k)$
- $h(x) = \bigoplus_i r_i\, x_i$

Gap Hamming (contd)

Claim. $\delta_1 \leq \Pr[h(x) \neq h(y)] \leq \delta_2$ and
$\delta_2 - \delta_1 = \Theta(1 - \exp(-\epsilon/2))$

Proof.

$$\Pr[h(x) \neq h(y)] = \frac{1}{2}\left(1 - \left(1 - \frac{1}{2k}\right)^{H(x,y)}\right)$$

Application: Dup detection

- Given a collection of web documents, find document pairs that are near-duplicates
  - Need not be an identical copy of one another
- Each document is represented by a set of k-grams (shingles)
- Eg, document = abacacd, k = 2
  shingles = { ab, ba, ac, ca, cd }
- Two documents are similar if they share many shingles
  - Compute MinHash or SimHash using several hash functions
  - Concatenate the hashes to obtain a "signature"
  - Sort documents by their signatures

Which similarities admit LSH?

- There are several similarities used in different scientific fields
  - Encyclopedia of Distances DL'11
- When does a similarity admit an LSH?
  - Similarity is LSHable if there is an LSH for it

A metric condition C'02

Theorem. S is LSHable $\Rightarrow$ 1 – S is a metric

Proof. Fix a hash function h and define

$$\Delta_h(A, B) \equiv [h(A) \neq h(B)]$$

$$1 - S(A, B) = \Pr_{h \in \mathcal{H}} \Delta_h(A, B)$$

$\Delta_h(A, B)$ satisfies the triangle inequality

$$\Delta_h(A, B) + \Delta_h(B, C) \geq \Delta_h(A, C)$$

Non-LSHable similarities

Distance $d(A, B) = 1 - S(A, B)$

Sørensen-Dice: $S(A, B) = 2|A \cap B| / (|A| + |B|)$

$A = \{a\}, B = \{b\}, C = \{a, b\}$

$S(A, B) = 0, S(A, C) = 2/3, S(B, C) = 2/3$

$d(A, B) = 1, d(A, C) = 1/3, d(B, C) = 1/3$

Overlap: $S(A, B) = |A \cap B| / \min(|A|, |B|)$

$S(A, B) = 0, S(A, C) = 1 = S(B, C)$

$d(A, B) = 1, d(A, C) = 0 = d(B, C)$

Sørensen-Dice and Overlap similarities are not LSHable

Binary range

Lemma. If S is LSHable, then (1+S)/2 is LSHable where range of hash functions is {0, 1}

Proof. Let $\mathcal{H}$ be a hash family for S and let $\mathcal{B}$ be a pairwise independent hash family whose domain is the range of $\mathcal{H}$ and whose range is {0, 1}

Consider $\{ b \circ h : b \in \mathcal{B}, h \in \mathcal{H} \}$

$$\Pr_{b\in\mathcal{B},\, h\in\mathcal{H}}[b(h(A))=b(h(B))] = (1+S(A, B))/2$$

wp S(A, B), $h(A) = h(B) \Rightarrow b(h(A)) = b(h(B))$

wp 1–S(A, B), $h(A) \neq h(B) \Rightarrow b(h(A)) = b(h(B))$, wp 1/2

Small support

Lemma. If $S : U \times U \to [0, 1]$ is LSHable, then the support of the LSH is $O(|U|^2)$

Proof. Consider the system of linear equations

p_h = probability assigned to hash function h

$$\forall\, A \neq B,\ S(A, B) = \sum_{h\,:\,h(A) = h(B)} p_h$$

$$\sum_h p_h = 1$$

Number of constraints = $|U|\ (|U|\text{-}1)/2 + 1$

An embeddability condition

Lemma. If S is LSHable, then 1 – S is isometrically embeddable into $L_1^{O(|U|^3)}$

Proof. Embedding $\phi(A)$ is as follows. For hash function h_i, $\phi(A)$ is a zero vector except for i-th co-ordinate, which has value

$$0.5*\Pr[h_i \text{ is chosen by LSH}]$$

$$L_1(\phi(A), \phi(B)) = \sum_{h_i : h_{i(A)} \neq h_{i(B)}} \Pr[h_i \text{ is chosen}]$$

$$= \Pr_h[h(A) \neq h(B)] = 1 - S(A, B)$$

Other popular set similarities

	Function	LSHable
Jaccard	$\frac{\lvert A \cap B \rvert}{\lvert A \cap B \rvert + \lvert A \triangle B \rvert}$	Yes
Hamming	$\frac{\lvert A \cap B \rvert + \lvert \overline{A \cup B} \rvert}{\lvert A \cap B \rvert + \lvert \overline{A \cup B} \rvert + \lvert A \triangle B \rvert}$	Yes
Sørensen-Dice	$\frac{\lvert A \cap B \rvert}{\lvert A \cap B \rvert + \frac{1}{2} \cdot \lvert A \triangle B \rvert}$	No
Anderberg	$\frac{\lvert A \cap B \rvert}{\lvert A \cap B \rvert + 2 \cdot \lvert A \triangle B \rvert}$	?
Rogers-Tanimoto	$\frac{\lvert A \cap B \rvert + \lvert \overline{A \cup B} \rvert}{\lvert A \cap B \rvert + \lvert \overline{A \cup B} \rvert + 2 \cdot \lvert A \triangle B \rvert}$	?

⋮

Compositional tools

Idea: Build a new LSH from existing LSHs

Claim. The similarity S such that S(A, B) = 1 for each A, B, is LSHable

Claim. The similarity S such that S(A, B) = 0 for each A ≠ B, is LSHable

Composition: Products

Claim. If S and T are LSHable then S·T is also LSHable

Proof. Independently sample h_1 from the LSH of S, h_2 from the LSH of T, concatenate $h(x) = (h_1(x), h_2(x))$

By independence

$$\Pr[h(A) = h(B)] = S(A,B) \cdot T(A, B)$$

Corollary. If S is LSHable then S^n is also LSHable for $n = 0, 1, 2, \ldots$

Composition: Convex combinations

Claim. If $S_0, S_1, \ldots$ are LSHable and if $p = (p_0, p_1, \ldots)$ is a distribution then $S = \sum_i p_i S_i$ is LSHable

Proof.

1. Sample i according to p
2. Sample h_i from the LSH of S_i
3. Output $(i, h_i(S))$

Probability generating functions

$p(x) = \sum_{i=0,\infty} p_i\, x^i$ is a probability generating function (PGF) iff $(p_0, p_1, \ldots)$ is a probability distribution

Eg, $\sum_{i=0,\infty} x^i/2^i = x/(2-x)$ has $(0, 1/2, 1/4, 1/8, \ldots)$

- $x/(w-(w-1)x)$ for $w \geq 1$
- $1 - (1 - x)^\alpha$ for $\alpha \in (0, 1]$
- $(\tan x)/(\tan 1)$, $(\arcsin x)/(\arcsin 1)$, e^{x-1}, …

Lemma. If S is LSHable and p is PGF, then p(S) is LSHable

PGFs are LSH-preserving CK'12

LSH-preserving transformations: those that preserve the LSHability of similarities

Theorem. A function is LSH-preserving if it is equal to a PGF multiplied by some constant $\alpha \in [0, 1]$

Proof. Take the $(1-\alpha, \alpha)$-average between and the similarity that comes out of the PGF and the trivial similarity

Eg, Anderberg similarity

- Definition $S'(A,B) = \frac{|A \cap B|}{|A \cap B| + 2 \cdot |A \triangle B|}$
- Consider the PGF

$$f(x) = \sum x^i/2^i = x/(2-x)$$

$$f\left(\frac{|A \cap B|}{|A \cup B|}\right) = \frac{\frac{|A \cap B|}{|A \cup B|}}{2 - \frac{|A \cap B|}{|A \cup B|}} = \frac{|A \cap B|}{2 \cdot |A \cup B| - |A \cap B|} = S'(A,B)$$

- MinHash is an LSH for Jaccard

LSH and PGFs

$p(x) = \sum_{i=0,\infty} p_i x^i$ is a probability generating function (PGF) iff $(p_0, p_1, \ldots,)$ is a probability distribution

- If p(x) is a PGF and $\alpha \in [0, 1]$, then S is LSHable $\Rightarrow$ $\alpha \cdot p(S)$ is LSHable
- If f(x) is such that f(S) is LSHable $\Leftarrow$ S is LSHable, then $\exists$ PGF p(x) and constant $\alpha \in [0,1]$ such that $f(x) = \alpha \cdot p(x)$

Theorem. A function is LSH-preserving iff it is equal to a PGF multiplied by some constant in [0,1]

Only if: Ingredients

1. We want to show that f is LSH-preserving only if it is a scaled-down PGF
2. The main task is to prove that all the forward differences of f have to be non-negative
3. To prove that, we assume that f had some specific negative forward difference, which we use to build an LSHable similarity IntSim
4. Finally, we give a solution to f(IntSim)'s dual system that proves that f(IntSim) is not LSHable

Rational set similarities (RSS)

Given $0 \leq x$, y and $0 \leq z \leq z'$, with $\max\{x,y,z'\} > 0$, the rational set similarity $S_{x,y,z,z'}$ between non-empty subsets $A,B \subseteq U$ is defined as

$$S_{x,y,z,z'}(A,B) = \frac{x\,|A \cap B| + y\,\left|\overline{A \cup B}\right| + z\,|A \Delta B|}{x\,|A \cap B| + y\,\left|\overline{A \cup B}\right| + z'\,|A \Delta B|}$$

LSHability of RSS

Theorem. The following are equivalent

i. $S_{x,y,z,z'}$ is LSHable
ii. $1 - S_{x,y,z,z'}$ satisfies the triangle inequality
iii. $z' \geq \max \{x,y,z\}$

Application: Near neighbors

d = metric of interest

Nearest neighbor problem: Given a set U of vectors and a query vector q, find $p^* \in U$ such that $d(p^*, q) \leq d(p, q)$ for all $p \in U$

- Several applications in several domains

Near neighbors: Given a set U of vectors, a query vector q, and an $\epsilon > 0$, find $p' \in U$ such that $d(p', q) \leq (1+\epsilon)\, d(p^*, q)$

Given R (threshold) and δ (recall), preprocess U to create a data structure to return $\geq 1 - \delta$ fraction of the neighbors at distance $\leq$ R of the query

NN via LSH

- LSH gives a bucket for each vector
  - Concatenate several independent hashes to obtain a bucket
- If two vectors are similar, then they hash to the same bucket with high probability
- If two vectors are dissimilar, then they hash to the same bucket with much lower probability

NN via SimHash

Given: d-dimensional input vectors

- A = k×d N(0,1) matrix; A_i = i-th row
  b = k-vector, each entry uniform in [4]
  $h_i(x) = \lfloor (A_i\, x + b_i) / 4 \rfloor$
- Buckets: let $g(x) = (h_1(x), \ldots, h_k(x))$
  - g(x) is a bucket of x
- Create independent buckets $g_1(x), \ldots, g_L(x)$
  - Each input is stored in L buckets

Querying: Search all the L buckets to get a candidate set

- Projection + candidate search

NN via SimHash (contd)

Hash to the same bucket: $\Pr[g(x) = g(y)] = S^k(x, y)$

Share some bucket:

$$\Pr[\exists\, j,\ g_j(x) = g_j(y)] = 1 - (1 - S^k(x, y))^L$$

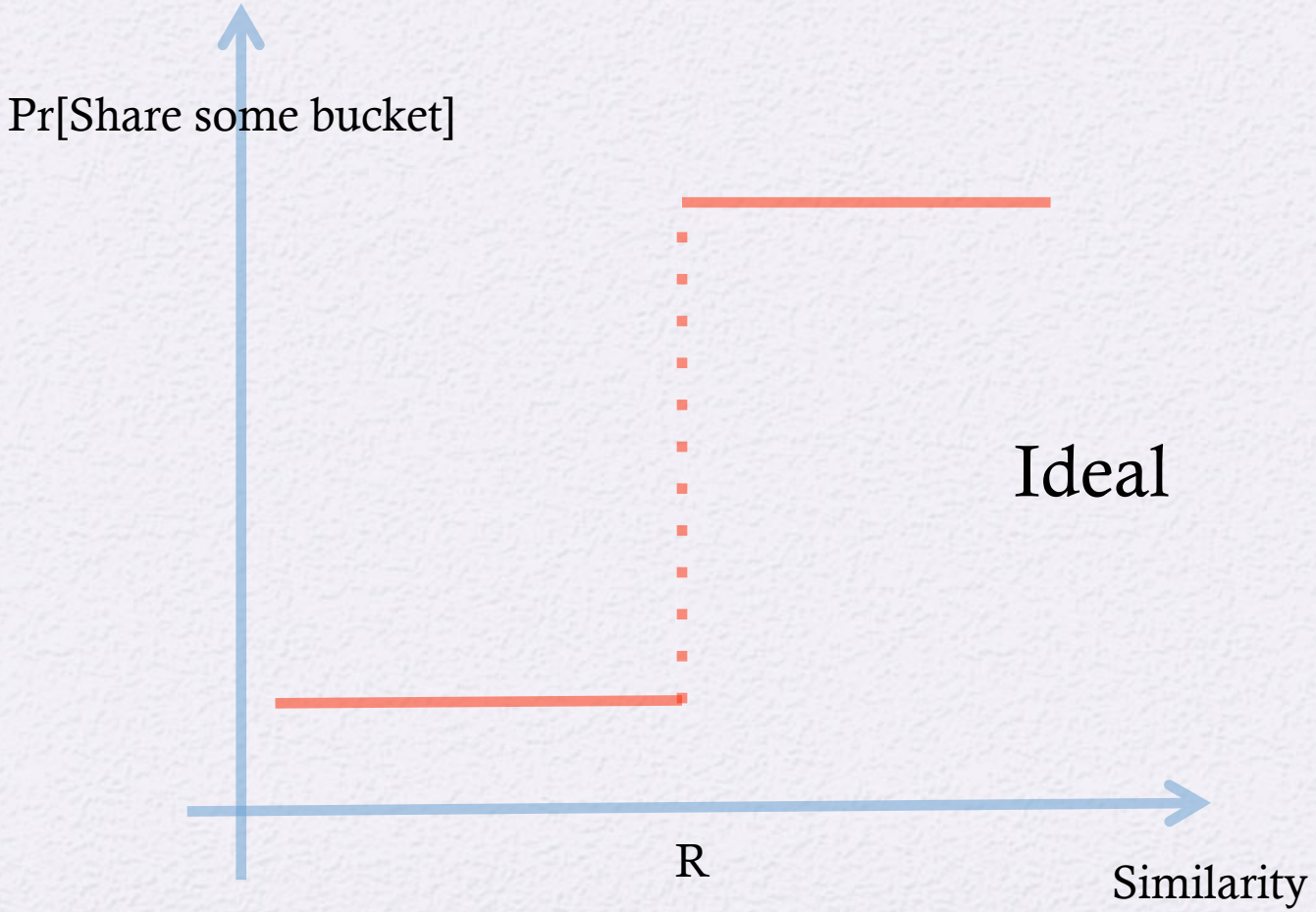
p(u) = collision probability for two distance-u vectors

Expected candidate set size = $1 - (1 - p^k(R))^L$

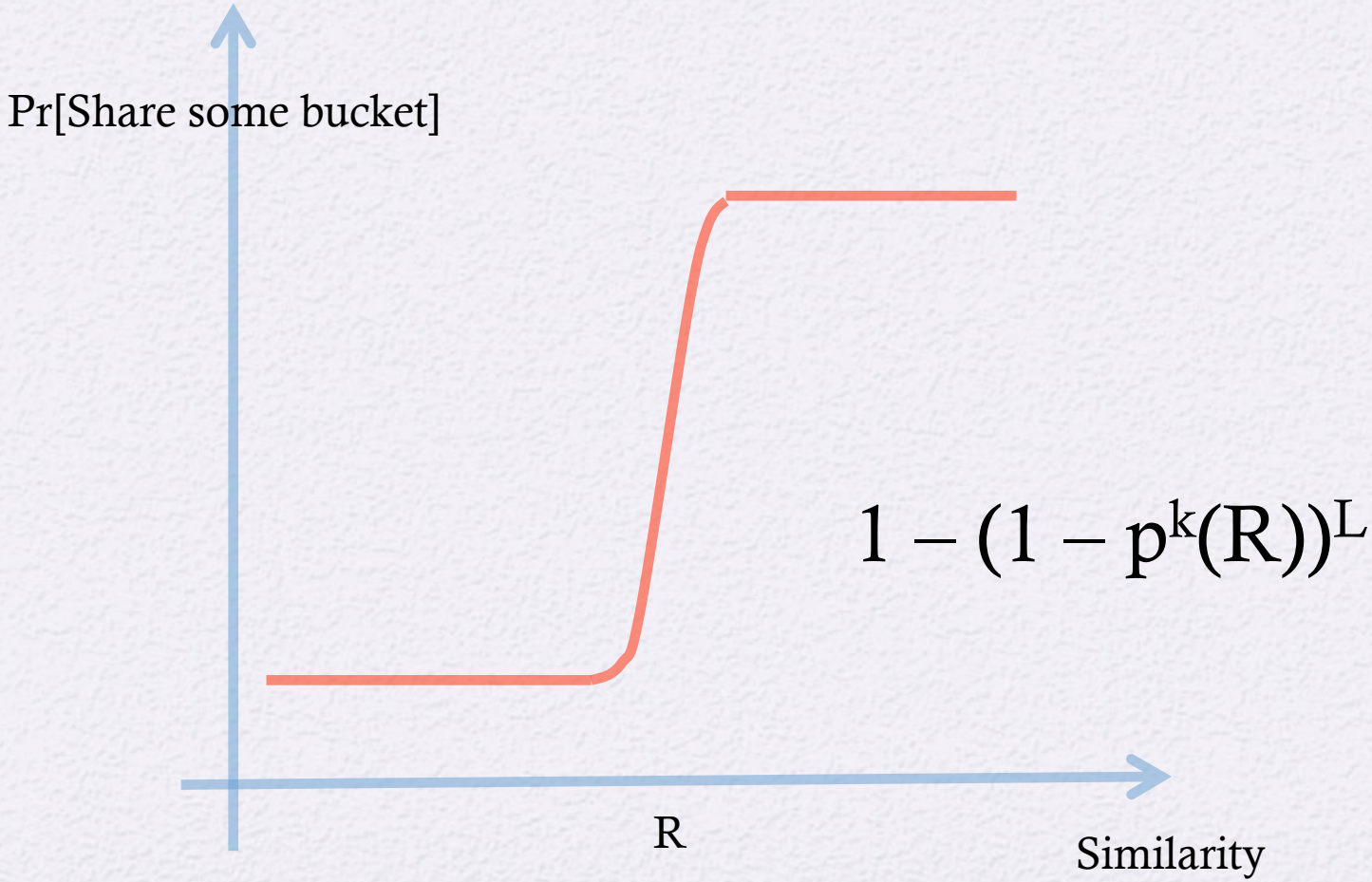
- k, L are determined if this is to be $\geq 1 - \delta$

Projection time = O(kdL), Space = O(nL)

Gap amplification



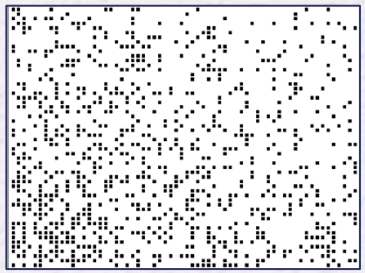
Gap amplification (contd)



$$1 - (1 - p^k(R))^L$$

Sparse Gaussians

- When can we use a sparse Gaussian P instead of the dense Gaussian A?
  - When the input vectors are dense



- If sparsity is p, then projection time = O(pdkL)

$$P_{i,j} = N(0,1)/\sqrt{p} \text{ wp } p$$
$$= 0 \text{ otherwise}$$

Random Hadamard Transform

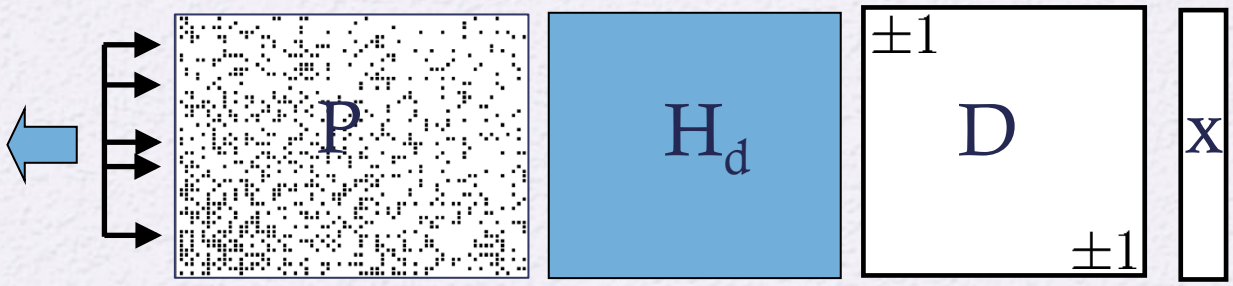
- How to densify input vectors? AC'09
  - Apply a random Hadamard transform
- $x \mapsto H_d D x$

$$H_d = \begin{pmatrix} H_{d/2} & H_{d/2} \\ H_{d/2} & -H_{d/2} \end{pmatrix} \text{ and } H_1 = (1)$$

D is a diagonal of iid ± 1 random variables

Theorem. The above takes O(d lg d) time and achieves $|x|_\infty = O((\log d)/\sqrt{d})$ whp

ACHash DKS'11



Sparsity $p = (\log d)/d$

Projection time = $O(d \log d + k L \log^2 d)$

Theorem. Collision probability is approximated by ACHash

$$-(k+1)\delta + p^k((1+\epsilon)u) \leq p_{AC}(u) \leq p^k((1-\epsilon)u) + (k+1)\delta$$

Argue that the sparsity of P is ok as long as $H_d Dx$ is dense

LSH feasibility problem

Given a similarity function S, is there a distribution over hash functions such that it forms an LSH for S?

Applications to similarities arising in specialized settings

- Entity matching
- Clustering

Infeasibility of feasibility CKM'12

Theorem. The LSH-feasibility problem is NP-hard even if S assumes only values in {0, 1/6, 1/3, 2/3, 1}

Theorem. It is NP-hard to distinguish

- S is LSHable
- S is $n^{2-\epsilon}$-far in L_1 norm from any LSHable similarity

Other interesting topics

- Improving efficiency
  - E2LSH, Multiprobe LSH, Entropy LSH, data-dependent LSH…
- Other LSH constructions
  - Stable distributions
  - LP/SDP rounding
- Relationship to computational models
  - Sketching, streaming
- Lower bounds

Thank you!

Questions/comments

ravi.k53@gmail.com