Locality Sensitive Hashing an algorithm for "Sublinear Time Similarity Search"

Rameshwar Pratap
rameshwar.pratap@gmail.com

March 21, 2018

Outline

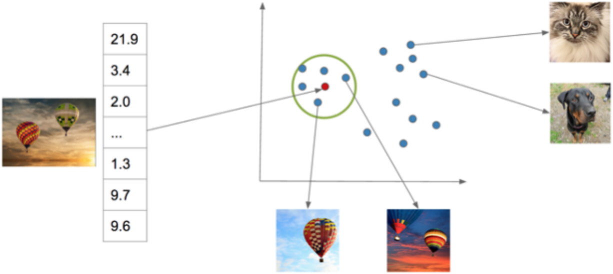
1. Similarity search problem
2. LSH – an algorithm for sublinear time similarity search
3. Variants of LSH
    - Sets and Jaccard similarity
    - Binary data and Hamming distance
    - Real valued vectors and Cosine similarity
    - Real valued vectors and Euclidean distance

Similarity Search in High Dimension

Given a query object and a database of objects the problem is to find a similar object *w.r.t.* the query object.

Similarity Search in High Dimension

Given a query object and a database of objects the problem is to find a similar object *w.r.t.* the query object.



Similarity Search in High Dimension

Exact nearest neighbor

1. given a set $\mathcal{X} = \{x_i\}_{i=1}^n$ of objects (off-line)
2. given a query object q (query time)
3. find the object in $\mathcal{X}$ that is the most similar to q

Similarity Search in High Dimension

Exact nearest neighbor

1. given a set $\mathcal{X} = \{x_i\}_{i=1}^{n}$ of objects (off-line)
2. given a query object q (query time)
3. find the object in $\mathcal{X}$ that is the most similar to q

Exact nearest neighbor for Euclidean distance

Algorithm	Query Time	Space
Full indexing	O(d log n)	$n^{O(d)}$
Linear scan	O(dn)	O(dn)

Feasible only when d is very small – "Curse of dimensionality".

Approximate Similarity Search

Idea: rather than retrieving for the exact nearest neighbor, make a "good guess" of the nearest neighbor.

Approximate Nearest Neighbor: Problem Statement

- given a set $\mathcal{X}$ of objects (off-line)
- given accuracy parameter ε (off-line)
- given a query object q (query time)
- find an object $z \in \mathcal{X}$ that is most similar to q such that

$$\mathrm{dis}(q, z) \leq (1+\varepsilon)\mathrm{dis}(q, x), \forall x \in \mathcal{X}$$

Approximate Similarity Search can be done in sublinear time with provable guarantee.

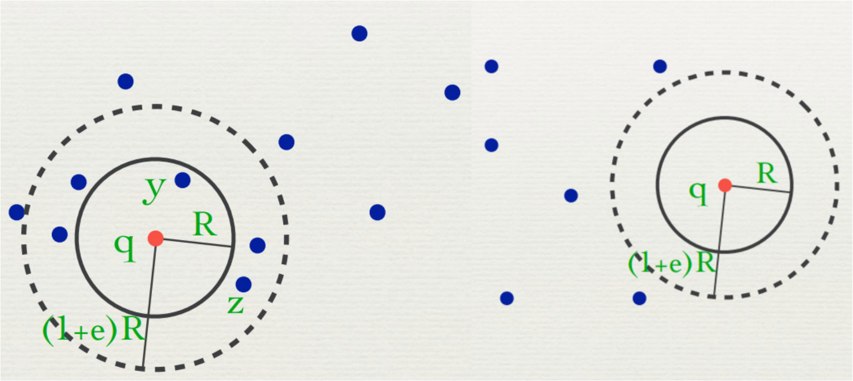
Approximate Similarity Search

Let's first focus on an easier version of the problem –approximate near neighbor

Approximate Near Neighbor

- given a set $\mathcal{X}$ of objects (off-line)
- given accuracy parameter ε and distance threshold R (off-line)
- given a query object q (query time)
- find an object $z \in \mathcal{X}$ that is most similar to q such that
  - if there is object y in $\mathcal{X}$ s.t. $\mathrm{dis}(q, y) \leq R$, then return object z in $\mathcal{X}$ s.t. $\mathrm{dis}(q, z) \leq (1 + \varepsilon)R$.
  - if there is no object y in $\mathcal{X}$ s.t. $\mathrm{dis}(q, y) \leq (1 + \varepsilon)R$, then return NO!.

Approximate Near Neighbor



Approximate Near(est) Neighbor

- Using "approximate near neighbor", we can solve "approximate nearest neighbor".

Approximate Near(est) Neighbor

- Using "approximate near neighbor", we can solve "approximate nearest neighbor".
- Let d and D be the smallest and largest pair-wise distance, then set $R = d, (1+\varepsilon)d, (1+\varepsilon^2)d, \ldots, D$

Approximate Near(est) Neighbor

- Using "approximate near neighbor", we can solve "approximate nearest neighbor".
- Let d and D be the smallest and largest pair-wise distance, then set $R = d, (1+\varepsilon)d, (1+\varepsilon^2)d, \ldots, D$
- Number of iterations: $O\left(\log_{1+\varepsilon} D/d\right)$.

Approximate Near(est) Neighbor

- Using "approximate near neighbor", we can solve "approximate nearest neighbor".
- Let d and D be the smallest and largest pair-wise distance, then set $R = d, (1+\varepsilon)d, (1+\varepsilon^2)d, \ldots, D$
- Number of iterations: $O\left(\log_{1+\varepsilon} D/d\right)$.
- return a point found in the non-empty ball with the smallest radius, and answer it as the approximate nearest neighbor for q

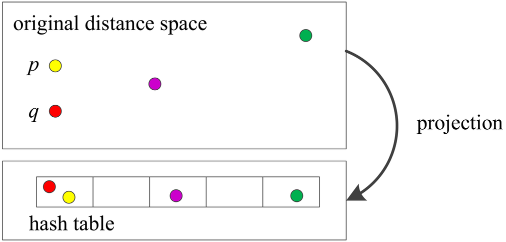
Approximate Near(est) Neighbor

- Using "approximate near neighbor", we can solve "approximate nearest neighbor".
- Let d and D be the smallest and largest pair-wise distance, then set $R = d, (1+\varepsilon)d, (1+\varepsilon^2)d, \ldots, D$
- Number of iterations: $O\left(\log_{1+\varepsilon} D/d\right)$.
- return a point found in the non-empty ball with the smallest radius, and answer it as the approximate nearest neighbor for q

Tool for approximate nearest neighbor – Locality Sensitive Hashing!

Locality sensitive hashing (LSH): Intuition

LSH algorithm hashes "similar" items to same value (bucket) and "not-so similar" items to different values (buckets).



- Idea: Only examine those items where the buckets are shared
- (Pro) Designed correctly, only a small fraction of items are examined
- (Con) There maybe false negatives

Locality sensitive hashing (LSH) : formal definition [Indyk and Motwani, 98]

LSH

A family $\mathcal{F}$ of hash functions is called (s, cs, p_1, p_2)-sensitive if for any two objects x and y

- if $\mathrm{Sim}(x, y) \geq s$, then $\Pr[h(x) = h(y)] \geq p_1$
- if $\mathrm{Sim}(x, y) \leq c.s$, then $\Pr[h(x) = h(y)] \leq p_2$
- probability over selecting h from $\mathcal{F}$
- $c < 1$, and $p_1 >> p_2$

LSH for Sets over Jaccard Similarity: An example

- **Data representation:** Sets.
- **Similarity measure:** Jaccard similarity.
  For two sets x and y, their Jaccard similarity is defined as

$$\mathrm{JS}(x, y) = \frac{|x \cap y|}{|x \cup y|}$$

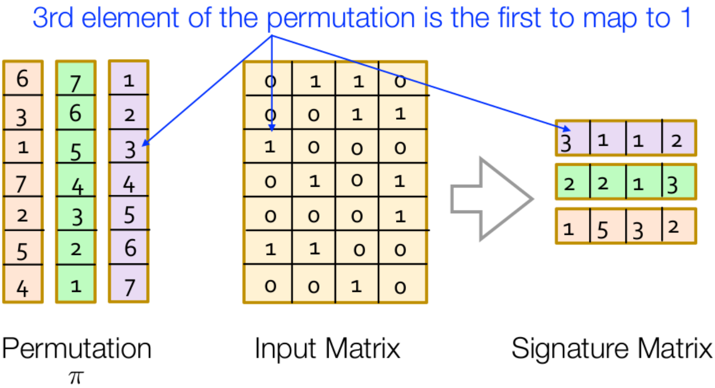
- For example: if $x = \{0, 3, 4\}$ and $y = \{1, 3, 5\}$, then $\mathrm{JS}(x, y) = 1/5$.

Min-wise permutations – LSH for Sets over Jaccard Similarity [Broder *et. al.*, 1998]

Sets can also be represented as binary vectors.

Min-wise permutations – LSH for Sets over Jaccard Similarity [Broder *et. al.*, 1998]

Sets can also be represented as binary vectors.



Minhash – LSH for Sets over Jaccard Similarity

Sets can be represented as binary vectors.

Permutation

6	7	1
3	6	2
1	5	3
7	4	4
2	3	5
5	2	6
4	1	7

Input Matrix

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

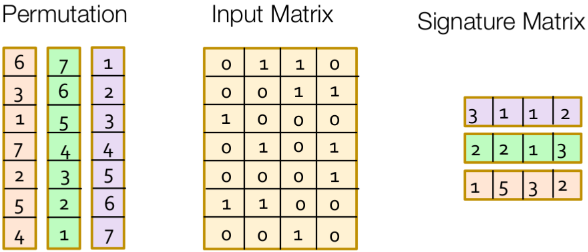
Signature Matrix

3	1	1	2
2	2	1	3
1	5	3	2

	1-2	2-3	3-4	1-3	1-4	2-4
Jaccard	1/4	1/5	1/5	0	0	1/5
Signature	1/3	1/3	0	0	0	0

Minhash – LSH for Sets over Jaccard Similarity

Sets can be represented as binary vectors.



	1-2	2-3	3-4	1-3	1-4	2-4
Jaccard	1/4	1/5	1/5	0	0	1/5
Signature	1/3	1/3	0	0	0	0

Theorem

$$\Pr[h_\pi(x) = h_\pi(y)] = \frac{|x \cap y|}{|x \cup y|} = \mathrm{JS}(x, y).$$

Minhash – LSH for Sets over Jaccard Similarity

Theorem

$\Pr[h_\pi(x) = h_\pi(y)] = \frac{|x \cap y|}{|x \cup y|} = \mathrm{JS}(x, y).$

Probability of collision is equal to their Jaccard similarity.

Minhash – LSH for Sets over Jaccard Similarity

Theorem

$\Pr[h_\pi(x) = h_\pi(y)] = \frac{|x \cap y|}{|x \cup y|} = \mathrm{JS}(x, y).$

Probability of collision is equal to their Jaccard similarity.
Question: Is it enough?

Minhash – LSH for Sets over Jaccard Similarity

Theorem

$\Pr[h_\pi(x) = h_\pi(y)] = \frac{|x \cap y|}{|x \cup y|} = \mathrm{JS}(x, y).$

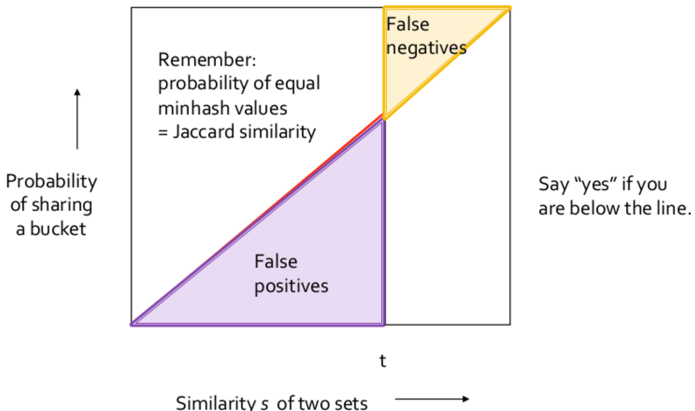
Probability of collision is equal to their Jaccard similarity.
Question: Is it enough? NO!

Minhash – LSH for Sets over Jaccard Similarity

Theorem

$$\Pr[h_\pi(x) = h_\pi(y)] = \frac{|x \cap y|}{|x \cup y|} = \mathrm{JS}(x, y).$$

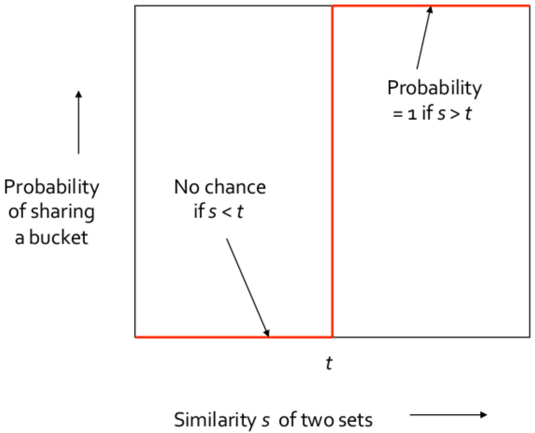
Probability of collision is equal to their Jaccard similarity.
Question: Is it enough? NO!



What do we want?

Minhash

What do we want?



Minhash

Idea: "Amplify the Gap"

- stack together many hash functions (say r)
  - probability of collision for similar objects decreases
  - probability of collision for dissimilar objects decreases much more
- repeat many times (say b)
  - probability of collision for similar objects increases

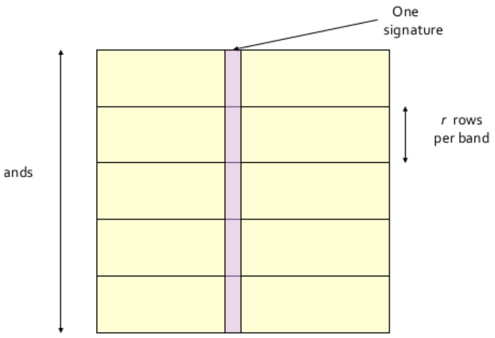


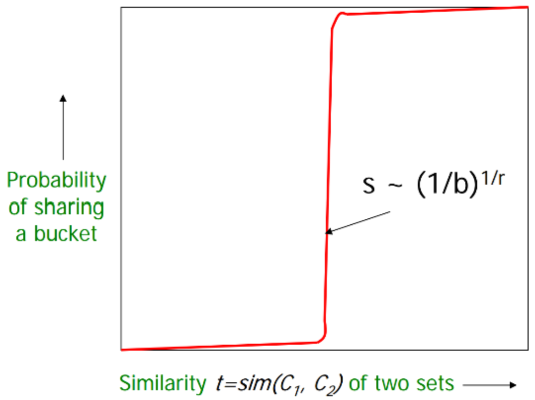
Figure: Minhash Sketch

Minhash – What b bands of r rows yields ?

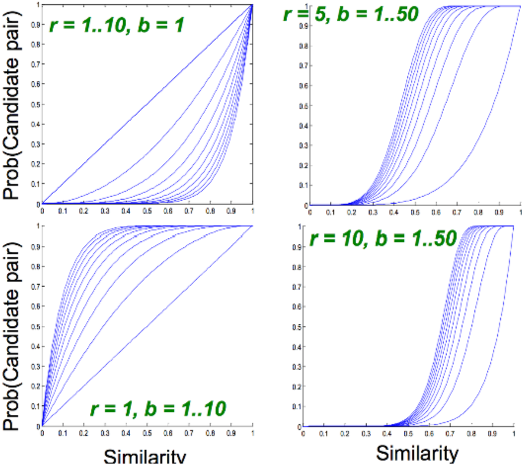
- Suppose sets x and y share similarity s
- Pick any band (r rows)
- Pr that all rows in band equal: s^r
- Pr unequal: $1 - s^r$
- Pr that no band is identical: $(1 - s^r)^b$
- Pr that at least one band is identical: $1 - (1 - s^r)^b$

Minhash – What b bands of r rows yields ?

- Suppose sets x and y share similarity s
- Pick any band (r rows)
- Pr that all rows in band equal: s^r
- Pr unequal: $1 - s^r$
- Pr that no band is identical: $(1 - s^r)^b$
- Pr that at least one band is identical: $1 - (1 - s^r)^b$



Minhash – Picking appropriate values of b and r.



Minhash – Picking appropriate values of b and r.

Theorem (Indyk and Motwani, 98)

Let $\mathcal{X} = \{x_i\}_{i=1}^n$*, q be a given query, and* $x^* \in \mathcal{X}$ *s.t.* $\mathrm{JS}(x^*, q) \geq s$*. If we set our hashing parameters*

$$r = \log_{1/p_2} n; \qquad b = n^\rho \log(1/\delta)$$

where, $p_1 = s, p_2 = c.s, \rho = \frac{\log 1/p_1}{\log 1/p_2} \leq \frac{1}{1+c}$.
Then following two cases are true with probability $> 1 - \delta :$

1. *for some* $i \in \{1, ..., b\}$*, hash value of* x^* *and* q *collides*
2. *no. of collisions with* x' *s.t.* $\mathrm{JS}(x', q) < c.s$ *is at most* $\frac{b}{\delta}$.

Minhash – Picking appropriate values of b and r.

Theorem (Indyk and Motwani, 98)

Let $\mathcal{X} = \{x_i\}_{i=1}^{n}$, q be a given query, and $x^* \in \mathcal{X}$ s.t. $\mathrm{JS}(x^*, q) \geq s$. *If we set our hashing parameters*

$$r = \log_{1/p_2} n; \qquad b = n^{\rho} \log(1/\delta)$$

where, $p_1 = s, p_2 = c.s, \rho = \frac{\log 1/p_1}{\log 1/p_2} \leq \frac{1}{1+c}$.
Then following two cases are true with probability $> 1 - \delta$:

1. *for some* $i \in \{1, ..., b\}$, *hash value of* x^* *and* q *collides*
2. *no. of collisions with* x' *s.t.* $\mathrm{JS}(x', q) < c.s$ *is at most* $\frac{b}{\delta}$.

- Size of Hash table $= b.r = O(n^{\rho} \log n)$
- Worst case query time $= b = o(n)$

1. Similarity Search Problem
2. LSH – a tool for sublinear time similarity search
3. Variants of LSH
4. LSH for other well-known similarity measure

LSH for other similarity measure

Two-step approach for LSH

- Given a data set (representation) and the similarity measure, the aim find a hash function s.t. collision probability is monotonic to their similarity.
- Banding and repetition

LSH for other similarity measure

Two-step approach for LSH

- Given a data set (representation) and the similarity measure, the aim find a hash function s.t. collision probability is monotonic to their similarity.
- Banding and repetition

LSH for other similarity measures

Data representation	Similarity	Reference
Sets	Jaccard	[Broder *et. al.*, 1998]
Binary vectors	Hamming	[Gionis *et. al.*, 1999]
Real-valued vectors	Cosine	[Charikar, 2002]
Real-valued vectors	Euclidean	[Indyk and Motwani, 98]

Thank You

Questions?