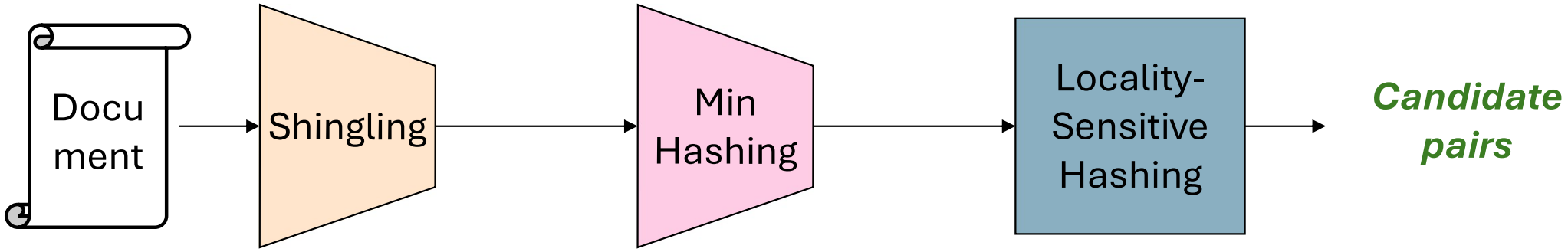


Locality Sensitive Hashing

Focus on pairs of signatures likely to be from similar documents

The key idea

Generate a **sketch** for every object that:

1) is ***much shorter*** than its # features (i.e. **d)**

2) transforms ***similarity*** between two feature vectors

into ***equality*** *of their shorter sketches*.

- ✓ It is ***randomized***, correct ***with high probability***

  (good if this is the only way to approach the problem !!)

- ✓ It guarantees ***local access*** to data, which is good for speed in disk/distributed setting

The hamming case

- Consider vectors p, q of **d** *binary* features

- Hamming distance

  D(p,q)= #bits where *p* and *q* differ

- Define hash function *h* by choosing a set *I* of *r* random coordinates

  h(p) = projection of vector *p* on *I's* coordinates

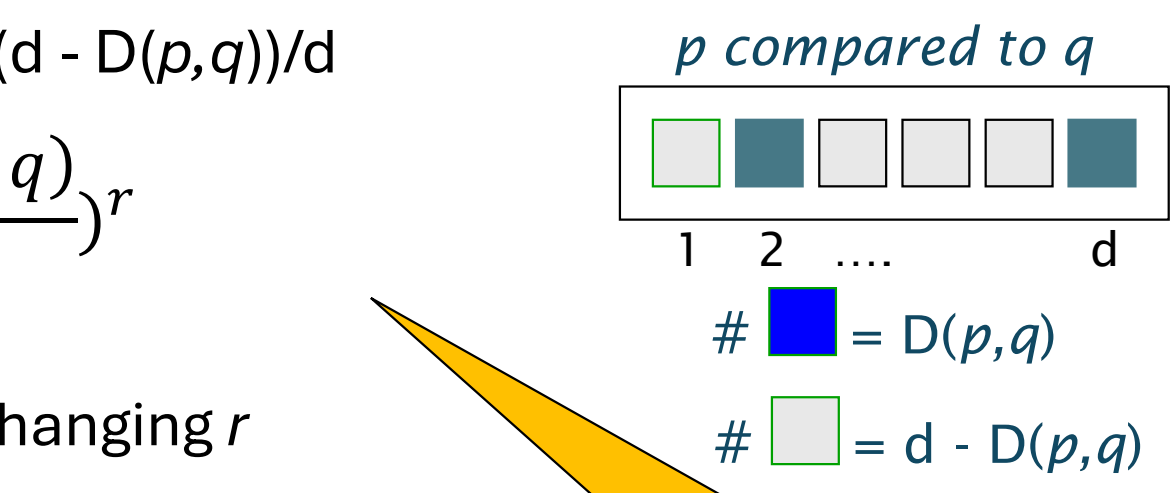
Example: If r=2, *assume I={1,4} then it is h(p=**0**10**1**1) = 01*

A key property

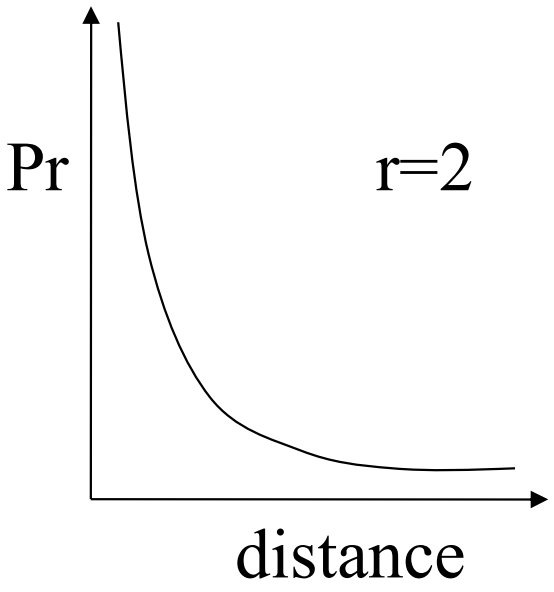
Pr[*picking x:p[x]=q[x]*] = # /d = (d - D(*p*,*q*))/d

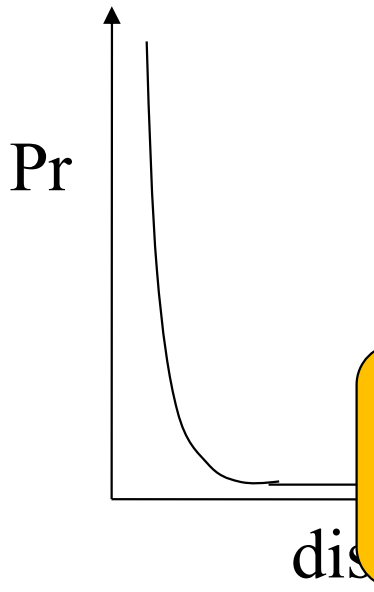
$$\Pr[\, h(p) = h(q)] = (1 - \frac{D(p,q)}{d})^r$$

We can vary the probability by changing *r*



= s^r
where s is the similarity between p and q





Larger r
Fewer False Positive
What about False Negatives?

Panoramica di LSH (Locality-Sensitive Hashing)

Processo

Da un documento si estraggono shingle, si applica **Min-Hashing** per creare firme, quindi **LSH** identifica coppie di firme simili (*candidate pairs*).

Obiettivo

Concentrarsi su coppie di firme che probabilmente derivano da documenti simili.

Idea Chiave

- Generare una firma (*sketch*) per ogni oggetto, più corta del numero di feature (d).
- Trasformare la similarità tra vettori di feature nella probabilità di uguaglianza delle firme.

1. Generare una firma (sketch) più corta del numero di feature (d)

Obiettivo: Ridurre la dimensionalità dei dati mantenendo l'informazione sulla similarità.

Come funziona:

- I dati originali (es. documenti) sono rappresentati come vettori di *features* (es. shingle, parole).
- Un documento può avere migliaia di shingle ($d \approx 10^4$), ma il confronto diretto è costoso.
- **Min-Hashing** genera una firma compatta tramite questi passi:
  1. Applicare una funzione hash casuale h a ogni shingle.
  2. Per ogni funzione hash, selezionare il valore hash *minimo* del documento.
  3. Ripetere con k funzioni hash diverse per ottenere una firma di lunghezza $k \ll d$.

Perché funziona:

- La probabilità che due documenti condividano un hash minimo è pari alla loro **similarità di Jaccard** s.

Dimostrazione intuitiva:

1. **Definizione di similarità di Jaccard**: Dati due insiemi A e B,
$$s = \frac{|A \cap B|}{|A \cup B|}.$$
2. **Permutazioni casuali**: Il Min-Hashing simula una permutazione casuale degli elementi (shingle) dei documenti. Per ogni permutazione, si seleziona il primo elemento (hash minimo) nell'ordine permutato.
3. **Probabilità di collisione**: La probabilità che il primo elemento della permutazione appartenga a $A \cap B$ è:
$$\Pr[\text{Min-Hash}(A) = \text{Min-Hash}(B)] = \frac{|A \cap B|}{|A \cup B|} = s.$$
Questo perché tutti gli elementi di $A \cup B$ hanno la stessa probabilità di essere il primo nella permutazione.

Esempio numerico:

- Documento 1 (A): Shingle = {a, b, c, d}
- Documento 2 (B): Shingle = {a, c, e, f}
- $A \cap B = \{a, c\}$ (2 elementi), $A \cup B = \{a, b, c, d, e, f\}$ (6 elementi)
- $s = \frac{2}{6} = \frac{1}{3}$.
- Generando una permutazione casuale degli elementi di $A \cup B$, la probabilità che il primo elemento sia in $A \cap B$ è $\frac{1}{3}$.

2. Trasformare la similarità tra vettori nella probabilità di uguaglianza delle firme

Obiettivo: Convertire s in una probabilità di collisione tra firme.

Meccanismo:

- Per due documenti con s, la probabilità che una funzione Min-Hash coincida è s.
- Con k funzioni hash, la probabilità di almeno una collisione è $1 - (1 - s)^k$.

Esempio numerico:

- Se $s = 0.8$:
  - $k = 1 \implies \Pr = 0.8$.
  - $k = 2 \implies \Pr = 1 - (0.2)^2 = 0.96$.

LSH: Bandizzazione

La **bandizzazione** in LSH (Locality-Sensitive Hashing) è una tecnica usata per velocizzare la ricerca di elementi simili in grandi dataset. Funziona dividendo le firme hash (sequenze di numeri che rappresentano i dati) in "bande" più piccole, aumentando la probabilità di trovare elementi vicini senza confrontare ogni coppia possibile.

Come funziona in breve:

- **Divisione in bande**: Una firma hash lunga viene spezzata in segmenti (bande) più corti.
- **Confronto rapido**: Solo gli elementi che hanno almeno una banda identica vengono considerati simili e confrontati in dettaglio.
- **Riduzione dei calcoli**: Questo filtra rapidamente i candidati, evitando di analizzare ogni coppia possibile.

Esempio pratico: Supponiamo di avere due firme hash di 12 bit che rappresentano due documenti:

- Doc1: `101101011010`
- Doc2: `101100011011`

Passo 1: Dividiamo ogni firma in 3 bande da 4 bit:

- Doc1: `1011 0101 1010`
- Doc2: `1011 0001 1011`

Passo 2: Confrontiamo le bande:

- Banda 1: `1011` = `1011` (**uguale!**)
- Banda 2: `0101` $\neq$ `0001`
- Banda 3: `1010` $\neq$ `1011`

Risultato: Poiché c'è almeno una banda uguale (la prima), Doc1 e Doc2 sono considerati "**candidati simili**" e vengono analizzati più a fondo. Se nessuna banda fosse uguale, li scarteremmo subito.

In pratica, questo metodo è utile per trovare somiglianze (es. documenti simili) senza dover confrontare ogni bit di ogni elemento!

Probabilità di collisione:

$$P(\text{candidati}) = 1 - (1 - s^r)^b$$

dove:

- s è la similarità tra i documenti,

- r è il numero di righe per banda,
- b è il numero di bande ($k = b \times r$).

Effetto di r e b:

- Aumentando r, si riduce la probabilità di falsi positivi, rendendo il filtro più selettivo.
- Aumentando b, si aumenta la probabilità di trovare coppie simili, ma aumenta anche il rischio di falsi positivi.

Vantaggi della bandizzazione:

- Riduce drasticamente il numero di confronti necessari.
- Permette di concentrarsi solo su coppie promettenti, migliorando l'efficienza.

Vantaggi

- È randomizzato, ma corretto con alta probabilità.
- Garantisce accesso locale ai dati, utile per velocità in contesti disk-based.

Hamming Distance

Si considerano vettori p e q di d feature binarie.

Distanza di Hamming

La distanza di Hamming $D(p, q)$ è il numero di bit in cui p e q differiscono.

Funzione Hash

Si scelgono l coordinate casuali e si proietta p su di esse ($h(p)$).

Esempio: Se $l = 2$, coordinate scelte $\{1, 4\}$:

$$h(p) = p[1]p[4].$$

Se $p = 01011$, allora:

$$h(p) = 01.$$

Esempio Pratico Integrato

Concetto base:
La distanza di Hamming misura quante posizioni differiscono tra due vettori binari p e q di lunghezza d. In LSH, per velocizzare il confronto, si usano funzioni hash che proiettano i vettori su un sottoinsieme di coordinate casuali.

Come funziona in breve:

- **Distanza di Hamming**: Conta i bit diversi tra p e q.
- **Funzione Hash**: Scegli l coordinate casuali e considera solo quei bit per creare una firma più corta $h(p)$.
- **Confronto**: Se $h(p) = h(q)$, i vettori sono probabilmente simili.

Esempio pratico:

- Abbiamo due vettori binari:

$$p = \texttt{01011}, \quad q = \texttt{11001}$$

- **Passo 1**: Calcoliamo la distanza di Hamming:

p :	0	1	0	1	1
q :	1	1	0	0	1
Differenze:	$\neq$	$=$	$=$	$\neq$	$=$

  Differenze: bit 1 ($0 \neq 1$), bit 4 ($1 \neq 0$) $\rightarrow$ Distanza = 2.

- **Passo 2**: Funzione hash con $l = 2$, coordinate casuali $\{1, 4\}$:

$$h(p) = p[1]p[4] = 01 \quad (\text{da } \texttt{01011})$$

$$h(q) = q[1]q[4] = 10 \quad (\text{da } \texttt{11001})$$

- **Risultato**: $h(p) = 01 \neq 10 = h(q)$, quindi non collidono nella stessa hash. Se fossero stati uguali (es. entrambi 01), li avremmo considerati candidati simili e controllati meglio.

In LSH, questo riduce i confronti, focalizzandosi solo sui vettori con hash simili!

Proprietà Probabilistica

Probabilità di Collisione

La probabilità che $h(p) = h(q)$ è data da:

$$\Pr[h(p) = h(q)] = \left(\frac{d - D(p,q)}{d}\right)^r = s^r,$$

dove:

- r è il numero di coordinate scelte (bande),
- d è la lunghezza totale dei vettori,
- $D(p, q)$ è la distanza di Hamming (numero di bit diversi),
- $s = \frac{d - D(p,q)}{d}$ è la similarità tra i vettori.

Spiegazione Intuitiva con Esempio

Concetto base:
In LSH con distanza di Hamming, la probabilità che due vettori p e q collidano ($h(p) = h(q)$) dipende da:

- La loro distanza di Hamming $D(p, q)$,
- Il numero di bit r considerati per l'hash.

Formule chiave:

1. Probabilità di scegliere un bit uguale:

$$\Pr[\text{bit uguale}] = \frac{\text{bit uguali}}{d} = \frac{d - D(p, q)}{d}$$

2. Probabilità di collisione con r bit:

$$\Pr[h(p) = h(q)] = \left(1 - \frac{D(p, q)}{d}\right)^r$$

Esempio pratico:

- Vettori binari di lunghezza $d = 5$:

$$p = \texttt{10110}, \quad q = \texttt{10011}$$

- **Passo 1**: Calcolo della distanza di Hamming:

p :	1	0	1	1	0
q :	1	0	0	1	1
Differenze:	$=$	$=$	$\neq$	$=$	$\neq$

  $D(p, q) = 2$ (bit 3 e 5 differiscono).

- **Passo 2**: Probabilità di un singolo bit uguale:

$$\frac{d - D(p, q)}{d} = \frac{5 - 2}{5} = 0.6$$

- **Passo 3**: Probabilità di collisione con $r = 2$:

$$(0.6)^2 = 0.36 \quad (36\%)$$

- Con $r = 4$:
$$(0.6)^4 = 0.1296 \quad (12.96\%)$$

Interpretazione:

- Con $r = 2$, c'è il 36% di probabilità che p e q collidano (più falsi positivi).
- Con $r = 4$, la probabilità scende al 12.96% (meno falsi positivi, ma più falsi negativi).

Compromesso di r:

- r grande: Curva più ripida → meno falsi positivi, più falsi negativi.
- r piccolo: Curva più piatta → più falsi positivi, meno falsi negativi.

Effetto di r

Aumentando r, la probabilità di collisione diminuisce per coppie meno simili, riducendo i falsi positivi. La slide mostra graficamente come $r = 4$ abbia una curva più ripida di $r = 2$, indicando meno falsi positivi.

Confronto tra Probabilità di Collisione: Bandizzazione vs. Distanza di Hamming

Il rapporto tra le probabilità di collisione nell'LSH con bandizzazione e nella distanza di Hamming è legato al modo in cui entrambe sfruttano la similarità dei dati, ma con meccanismi diversi. Ecco una sintesi strutturata:

1. Probabilità di Collisione nella Bandizzazione (LSH)

- **Formula**:
$$P(\text{candidati}) = 1 - (1 - s^r)^b$$
dove:
  - s: Similarità tra documenti (es. Jaccard)
  - r: Numero di righe per banda
  - b: Numero di bande
- **Significato**:
  - Probabilità che due firme condividano almeno una banda
  - $r \uparrow$: Curva più ripida → meno falsi positivi
  - $b \uparrow$: Maggiore probabilità di trovare coppie simili (più falsi positivi)

2. Probabilità di Collisione con Distanza di Hamming

- **Formula**:
$$\Pr[h(p) = h(q)] = \left(1 - \frac{D(p,q)}{d}\right)^r = s^r$$
dove:
  - $D(p,q)$: Distanza di Hamming
  - d: Lunghezza dei vettori
  - r: Bit selezionati
- **Significato**:
  - Probabilità di collisione su r coordinate casuali
  - $r \uparrow$: Probabilità $\downarrow$ drasticamente per coppie lontane

Relazione tra i Concetti

Analogie:

- Entrambe usano una misura di similarità s come base
- r controlla la selettività in entrambi i casi
- Trade-off falsi positivi/negativi regolato da r

Differenze:

- **LSH con bandizzazione**:
  - Applicabile a firme generiche (es. Min-Hash)
  - Combina b bande e r righe
  - Formula di amplificazione: $1 - (1 - s^r)^b$
- **Distanza di Hamming**:
  - Specifica per vettori binari
  - Proiezione diretta su coordinate
  - Formula base: s^r

Esempio Numerico Comparato

Per due documenti con similarità $s = 0.8$:

- **Bandizzazione** ($r = 2$, $b = 3$):
$$P = 1 - (1 - 0.8^2)^3 = 1 - (0.36)^3 \approx 0.95 \quad (95\%)$$
- **Distanza Hamming** ($r = 2$):
$$\Pr = 0.8^2 = 0.64 \quad (64\%)$$

Conclusione

- La bandizzazione usa firme generiche e multiple bande (b) per bilanciare precisione/recall
- La distanza di Hamming lavora su vettori binari con proiezione diretta
- Entrambe usano r come leva principale per controllare il trade-off falsi positivi/negativi

Reiterate b times (calle band)

Larger b
Fewer False Negatives

1) Repeat **b times** the **r-projections** $h_i(p)$
2) We set $g(p) = < h_1(p), h_2(p), \ldots, h_b(p)>$
3) Declare «p matches q» if **at least** one $h_i(p)=h_i(q)$

Sketch(p)

Example:

*Let us set **r=2, b=3**, assume p = 01**0**01 and q = 01**1**01*

- *I1 = {3,4}, we have $h_1(p) = 00$* and *$h_1(q)=10$*
- *I2 = {1,3}, we have $h_2(p) = 00$* and *$h_2(q)=01$*
- *I3 = {1,5}, we have $h_3(p) = 01$* and *$h_3(q)=01$*

p and q declared to match !!

Measuring the error probability

$$\Pr[\, h_i(p) = h_i(q)] = (1 - \frac{D(p,q)}{d})^r = s^r$$

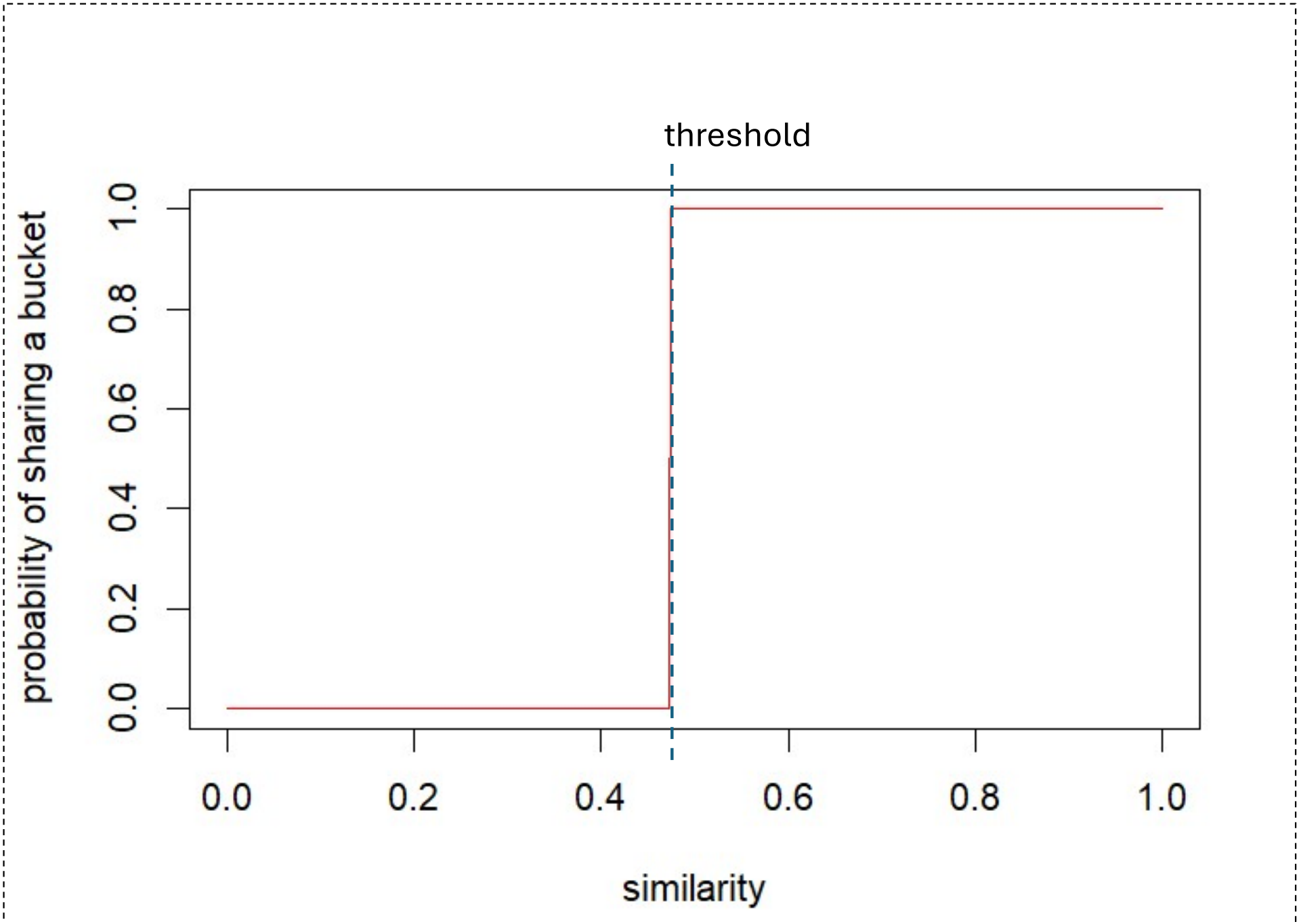
The g() consists of b independent hashes h_i

Pr[p not-similar q] = Pr[$h_i(p) \neq h_i(q),\ \forall i=1, \ldots, b$]

= (Pr[$h_i(p) \neq h_i(q)$])^b

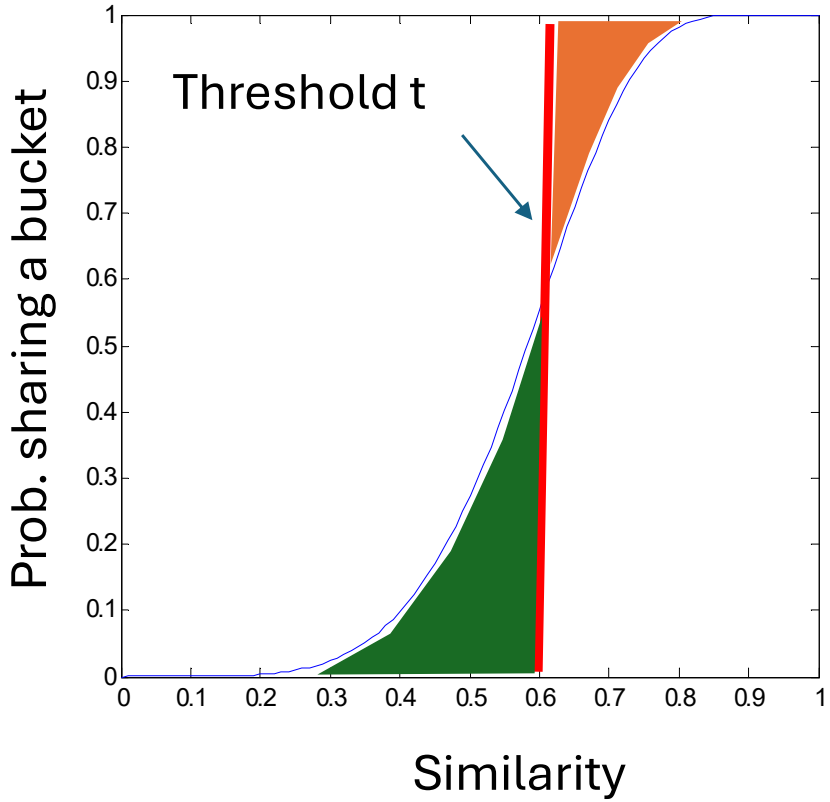
= (1- Pr[$h_i(p) = h_i(q)$])^b

The S-curve $1 - (1 - s^r)^b$



Picking r and *B*: The S-curve

- **Picking *r* and *b* to get the best S-curve**
  - 50 hash-functions (r=5, b=10)

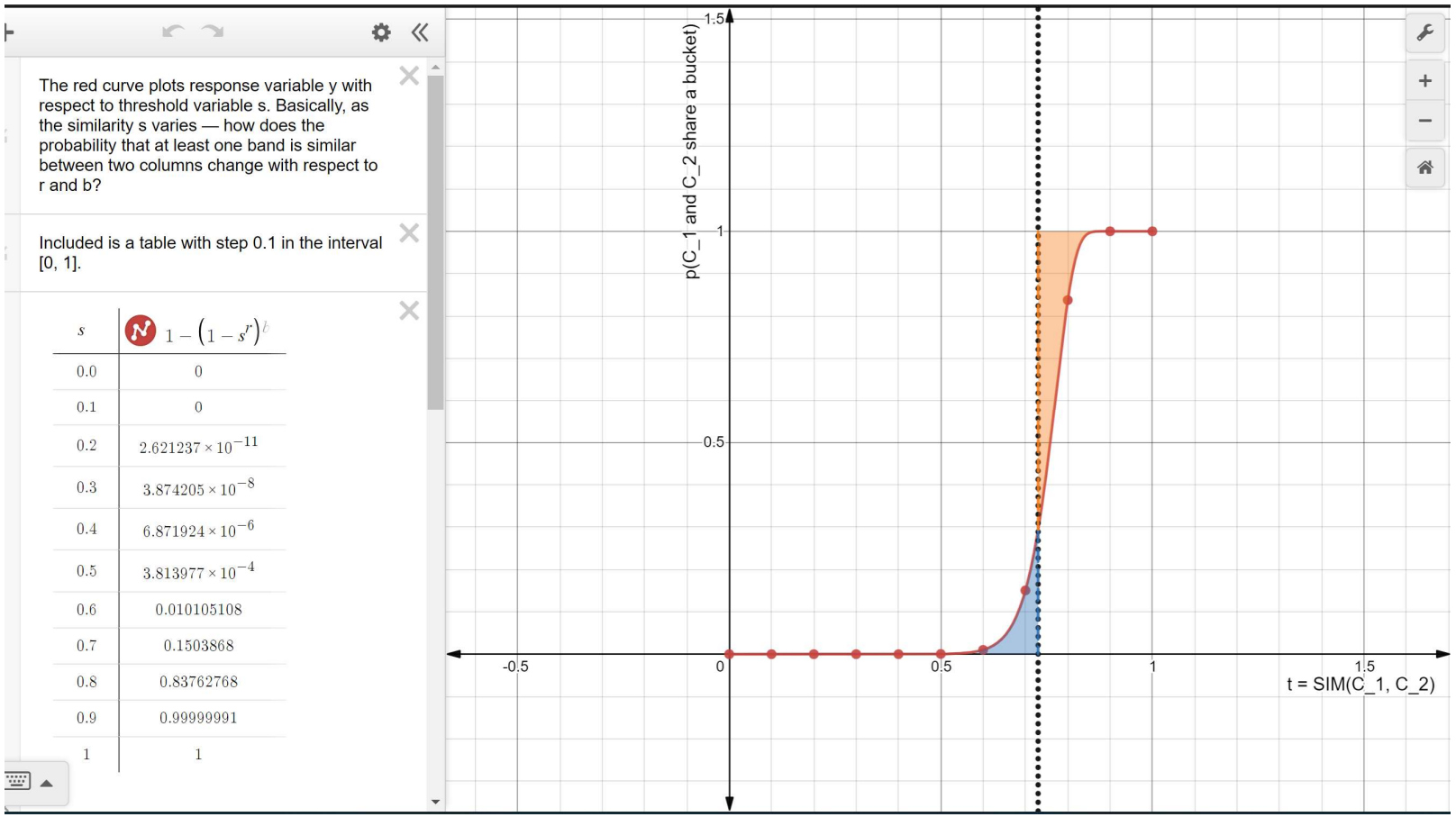


Red area: False Negative rate
Pairs with sim>t that will be never considered

Green area: False Positive rate
Pairs with sim<t, we can Discard these pairs once we Compute the exact distance

Demo

- https://www.desmos.com/calculator/lzzvfjiujn



Banding nel Contesto della Riduzione dei Falsi Negativi e Locality-Sensitive Hashing (LSH)

1 Introduzione

Il *banding* è una tecnica utilizzata nell'ambito del Locality-Sensitive Hashing (LSH) per migliorare l'identificazione di coppie di documenti simili, riducendo i falsi negativi (coppie simili che non vengono rilevate). In questo documento, esploriamo il concetto di banding e il suo ruolo nella riduzione dei falsi negativi, con un focus sulle sue implicazioni pratiche e teoriche.

2 Il Banding e la Riduzione dei Falsi Negativi

2.1 Proiezioni Ripetute

Si generano b proiezioni $h_i(p)$, ognuna di dimensione r, per ogni vettore p. La firma complessiva di p è definita come:

$$g(p) = \langle h_1(p), h_2(p), \ldots, h_b(p) \rangle$$

Ogni $h_i(p)$ proietta p su r coordinate casuali, creando una "banda". Queste bande sono utilizzate per confrontare la similarità tra vettori.

2.2 Criterio di Similarità

Due vettori p e q sono dichiarati simili se almeno una delle b proiezioni coincide, ovvero se esiste un indice i tale che:

$$h_i(p) = h_i(q)$$

Questo approccio aumenta la probabilità di rilevare coppie simili, poiché è sufficiente che coincidano in una sola banda.

2.3 Esempio Pratico

Consideriamo un esempio con $r = 2$, $b = 3$, $p = 01001$, e $q = 01101$:

- $h_1(p) = 00$, $h_1(q) = 10$ (diversi).
- $h_2(p) = 00$, $h_2(q) = 01$ (diversi).
- $h_3(p) = 01$, $h_3(q) = 01$ (uguali).

Poiché $h_3(p) = h_3(q)$, i vettori p e q sono dichiarati simili, anche se differiscono nelle prime due bande.

2.4 Probabilità e Falsi Negativi

La probabilità che $h_i(p) = h_i(q)$ in una singola banda è data da:

$$s^r$$

dove $s = 1 - \frac{D(p,q)}{d}$ rappresenta la similarità tra p e q. La probabilità che p e q non coincidano in nessuna delle b bande (falso negativo) è:

$$\Pr[h_i(p) \neq h_i(q), \forall i] = (1 - s^r)^b$$

La probabilità che siano dichiarati simili (almeno una banda coincide) è:

$$1 - (1 - s^r)^b$$

Aumentando b, $(1-s^r)^b$ diminuisce, riducendo la probabilità di falsi negativi. Ad esempio, se $s = 0.8$, $r = 5$, $s^r = 0.328$, con $b = 20$, la probabilità di un falso negativo è:

$$(1 - 0.328)^{20} \approx 0.00035$$

quindi si rileva il 99.965% delle coppie simili.

2.5 S-Curve e Bilanciamento

La funzione:

$$1 - (1 - s^r)^b$$

descrive la probabilità di dichiarare p e q simili in base alla loro similarità s. Una soglia t separa:

- Coppie con $s > t$: alta probabilità di match, ma se non rilevate sono falsi negativi.
- Coppie con $s < t$: bassa probabilità di match, ma se rilevate sono falsi positivi.

Con $r = 5$ e $b = 10$, la S-curve mostra che per $s > t$, la probabilità di match è alta, riducendo i falsi negativi, mentre per $s < t$, i falsi positivi sono limitati e possono essere filtrati calcolando la distanza esatta.

3 Collegamento con LSH

3.1 Ruolo di LSH

LSH utilizza il Min-Hashing per creare firme compatte (*sketch*) dei documenti. Le firme sono poi proiettate in bucket usando funzioni hash sensibili alla località: coppie simili hanno alta probabilità di cadere nello stesso bucket.

3.2 Integrazione del Banding

In LSH, una singola proiezione $h(p)$ potrebbe non rilevare coppie simili (falsi negativi), soprattutto se r è grande (riduce la probabilità s^r). Il banding introduce b proiezioni indipendenti, aumentando le possibilità di trovare almeno un bucket comune per coppie simili. Questo è cruciale in LSH, poiché l'obiettivo è identificare *candidate pair* (coppie potenzialmente simili) senza calcolare la similarità esatta per tutte le coppie, che sarebbe computazionalmente costoso.

3.3 Bilanciamento con r e b

- r: Controlla la dimensione di ogni banda. Un r grande riduce s^r, diminuendo i falsi positivi (coppie non simili che coincidono), ma aumenta i falsi negativi.
- b: Aumentando il numero di bande, si riducono i falsi negativi, poiché basta una coincidenza per dichiarare una coppia simile.

LSH usa la S-curve per scegliere r e b in modo da massimizzare il rilevamento di coppie simili ($s > t$) e minimizzare i falsi positivi ($s < t$).

4 In Sintesi

Il banding in LSH riduce i falsi negativi ripetendo b volte le proiezioni r-dimensionali, aumentando la probabilità di rilevare coppie simili (da s^r a $1-(1-s^r)^b$). Questo si integra con LSH, che mira a identificare *candidate pair* in modo efficiente, usando la S-curve per bilanciare falsi positivi e negativi attraverso la scelta di r e b, garantendo che coppie con alta similarità siano quasi sempre rilevate.

Example

- Assume the following case:
  - Suppose 100,000 columns of M (100k docs)
  - Signatures of length 100, stored as integers (rows)
  - Therefore, signatures take 40MB
  - Goal: Find pairs of documents that are at least s = 0.8 similar
  - Choose b = 20 bands of r = 5 integers/band

Two columns highly similar

- Find pairs of $\geq$ s=0.8 similarity, let's set b=20, r=5
- Assume: sim(C1, C2) = 0.8
- Since sim(C1, C2) $\geq$ s, we want C1, C2 to be a candidate pair: We want them to hash to at least 1 common bucket (at least one band is identical):
- Prob. C1, C2 identical in one particular band: (0.8) $(0.8)^5$ = 0.328
- So, prob. C1, C2 are not similar in all 20 bands: $(1 - 0.328)^{20}$ = 0.00035
- That is, about 1/3000th of the 80%-similar column pairs are false negatives (we miss them)
- We would find 99.965% pairs of truly similar documents

Two column far away

- Find pairs of $\geq$ s=0.8 similarity, let's set b=20, r=5
- Assume: sim(C1, C2) = 0.3
- Since sim(C1, C2) $<$ s, we want C1, C2 to be a candidate pair: We want them to be not a candidate pair (all bands are differt):
- Prob. C1, C2 identical in one particular band: $(0.3)^5$ = 0.00243
- So, prob. C1, C2 are identical in at least one band is: 1- $(1 - 0.00243)^{20}$ = 0.0474
- We have that 4.75% pair of documents with similarity 0.3 will appear as candidate pairs.
- These are false positive.

Example: $b = 20$; $r = 5$

- Similarity threshold **s**
- Prob. that at least 1 of r proj of the sketch is identical:

s	**$1-(1-s^r)^b$**
.2	.006
.3	.047
.4	.186
.5	.470
.6	.802
.7	.975
.8	.9996

Scenario Generale

Caso

Abbiamo 100,000 colonne (documenti) con 100 righe, rappresentate come interi, per un totale di 40MB di firme.

Obiettivo: Trovare coppie di documenti con similarità $s \geq 0.8$.

Parametri:

- $b = 20$ bande,
- $r = 5$ interi/banda.

Due Colonne Altamente Simili

- **Similarità**: $\text{sim}(C_1, C_2) = 0.8$.
- **Condizione**: Poiché $\text{sim}(C_1, C_2) \geq s$, C_1 e C_2 devono essere *candidate pair*, condividendo almeno un bucket comune.
- **Probabilità per banda**:

$$(0.8)^5 = 0.328.$$

- **Probabilità su tutte le bande**:

$$1 - (1 - 0.328)^{20} = 1 - 0.00035 \approx 0.99965.$$

  Quindi, trovano il 99.965% delle coppie simili vere.

- **Falsi negativi**: Solo 1/3000 delle coppie simili sono perse.

Due Colonne Lontane

- **Similarità**: $\text{sim}(C_1, C_2) = 0.3$.
- **Condizione**: Poiché $\text{sim}(C_1, C_2) < s$, C_1 e C_2 non dovrebbero essere *candidate pair* (tutte le bande diverse).
- **Probabilità per banda**:

$$(0.3)^5 = 0.00243.$$

- **Probabilità di match in almeno una banda**:

$$1 - (1 - 0.00243)^{20} \approx 0.0474.$$

  Quindi, il 4.74% delle coppie con similarità 0.3 appaiono come *candidate pair*.

- **Falsi positivi**: Queste sono tutte false positive.

Tabella di Probabilità

La probabilità che almeno una delle r proiezioni sia identica è data da:

$$1 - (1 - s^r)^b.$$

Valori per $b = 20$, $r = 5$:

- $s = 0.2$: 0.006,
- $s = 0.3$: 0.047,
- $s = 0.4$: 0.186,
- $s = 0.5$: 0.470,
- $s = 0.6$: 0.802,
- $s = 0.7$: 0.975,
- $s = 0.8$: 0.9996.

Con $s = 0.8$, la probabilità è quasi 1, confermando l'efficacia per coppie simili.

In Sintesi

Con $b = 20$ e $r = 5$, LSH identifica efficacemente coppie con $s \geq 0.8$ (99.965% di successo), mentre coppie con $s = 0.3$ hanno solo un 4.74% di falsi positivi, bilanciando la precisione per somiglianze elevate e riducendo gli errori per somiglianze basse.

The (off-line) algorithm

- For every feature vector p, compute

$$g(p) = < h_1(p), h_2(p), \ldots, h_b(p)>$$

Sketch(p)

- For every i=1, 2, .., b, create the clustering C_i by putting in the same group vectors *p* and *q* iff $h_i(p) = h_i(q)$

Sort

- Create an undirected graph such that nodes *p* and *q* are **linked** iff their sketches are in the same cluster of C_i for some iteration *i*
- Compute the **connected components** because they provide groups of similar vectors

Notebook

Tutorial su MinHashing e LSH

https://colab.research.google.com/drive/1hjklFJM-1PMLNSl2MOqe7ZnLxkOLLc-A?usp=sharing

Next step: LSH Tuning

- Tune M, b, r to get almost all pairs with similar signatures, but eliminate most pairs that do not have similar signatures

Algoritmo Offline per Identificare Coppie di Documenti Simili con LSH

La slide descrive un algoritmo offline per identificare coppie di documenti simili usando **Locality-Sensitive Hashing (LSH)** con banding, seguito da un'ottimizzazione dei parametri.

Calcolo delle Firme

Per ogni vettore di feature p, si calcola una firma composta da b proiezioni:

$$g(p) = \langle h_1(p), h_2(p), \ldots, h_b(p) \rangle,$$

dove ogni $h_i(p)$ è una proiezione (*sketch*) di p.

Raggruppamento per Bande

Per ogni banda i (da 1 a b), si crea un cluster C_i. Due vettori p e q sono nello stesso cluster C_i se:

$$h_i(p) = h_i(q),$$

cioè se condividono lo stesso bucket nella banda i.

Questo passaggio richiede un ordinamento (*sort*) per identificare i bucket comuni.

Creazione del Grafo

Si costruisce un grafo non orientato in cui:

- I nodi sono i vettori p.
- Due nodi p e q sono collegati da un arco se appartengono allo stesso cluster C_i per almeno una banda i, indicando una potenziale similarità.

Componenti Connesse

Si calcolano le componenti connesse del grafo. Ogni componente connessa rappresenta un gruppo di vettori (documenti) simili, poiché sono collegati attraverso almeno una banda comune.

Prossimo Passo: Ottimizzazione

Si devono regolare i parametri M (numero di documenti), b (numero di bande), e r (dimensione di ogni banda) per:

- Identificare quasi tutte le coppie con firme simili (*minimizzare i falsi negativi*).
- Eliminare la maggior parte delle coppie non simili (*minimizzare i falsi positivi*).

In Sintesi

L'algoritmo offline usa LSH con b bande per raggruppare vettori in cluster basati su firme, costruisce un grafo collegando vettori che condividono almeno un bucket, e identifica gruppi simili tramite componenti connesse. Questo prepara il terreno per ottimizzare M, b, e r.