

CS276A
 Text Information Retrieval, Mining, and Exploitation

 Supplemental Min-wise Hashing Slides
 (Broder97, Broder98)
 (Adapted from Rajeev Motwani's CS361A slides)

Set Similarity

- **Set Similarity** (Jaccard measure)

$$\text{sim}_J(C_i, C_j) = \frac{|C_i \cap C_j|}{|C_i \cup C_j|}$$
- **View sets as columns of a matrix; one row for each element in the universe. $a_{ij} = 1$ indicates presence of item i in set j**
- **Example**

	C_1	C_2	
0	1		$\text{sim}_J(C_1, C_2) = 2/5 = 0.4$
1	0		
1	1		
0	0		
1	1		
0	1		

Identifying Similar Sets?

- **Signature Idea**
 - Hash columns C_i to signature $\text{sig}(C_i)$
 - $\text{sim}_J(C_i, C_j)$ approximated by $\text{sim}_H(\text{sig}(C_i), \text{sig}(C_j))$
- **Naïve Approach**
 - Sample P rows uniformly at random
 - Define $\text{sig}(C_i)$ as P bits of C_i in sample
 - **Problem**
 - sparsity \rightarrow could easily miss interesting part of columns
 - sample would get only 0's in columns

Key Observation

- For columns C_i, C_j , four types of rows

	C_i	C_j
A	1	1
B	1	0
C	0	1
D	0	0
- **Overload notation:** $A = \#$ of rows of type A
- **Claim**

$$\text{sim}_J(C_i, C_j) = \frac{A}{A+B+C}$$

Min Hashing

- Randomly **permute** rows
- Hash $h(C_i) =$ index of first row with 1 in column C_i
- **Surprising Property**

$$P[h(C_i) = h(C_j)] = \text{sim}_J(C_i, C_j)$$
- **Why?**
 - Both are $A/(A+B+C)$
 - Look down columns C_i, C_j until first **non-Type-D** row
 - $h(C_i) = h(C_j) \leftrightarrow$ type A row

Min-Hash Signatures

- Pick P random row permutations
- **MinHash Signature**

$$\text{sig}(C) = \text{list of } P \text{ indexes of first rows with 1 in column } C$$
- **Similarity of signatures**
 - Let $\text{sim}_H(\text{sig}(C_i), \text{sig}(C_j)) =$ fraction of permutations where MinHash values agree
 - Observe $E(\text{sim}_H(\text{sig}(C_i), \text{sig}(C_j))) = \text{sim}_J(C_i, C_j)$

Example

				Signatures			
				S ₁	S ₂	S ₃	
				Perm 1 = (12345)	1	2	1
				Perm 2 = (54321)	4	5	4
				Perm 3 = (34512)	3	5	4
				Similarities			
				1-2	1-3	2-3	
Col-Col	0.00	0.50	0.25				
Sig-Sig	0.00	0.67	0.00				

	C ₁	C ₂	C ₃
R ₁	1	0	1
R ₂	0	1	1
R ₃	1	0	0
R ₄	1	0	1
R ₅	0	1	0

Exercise: What similarities do we get if we also picked a min-hash for Perm 4=(21453) ?

7

Implementation Trick

- Permuting rows even once is prohibitive
- **Row Hashing**
 - Pick P hash functions $h_k: \{1, \dots, n\} \rightarrow \{1, \dots, O(n)\}$
 - Ordering under h_k gives random row permutation
- **One-pass Implementation**
 - For each C_i and h_k , keep "slot" for min-hash value
 - Initialize all $\text{slot}(C_i, h_k)$ to infinity
 - Scan rows in arbitrary order looking for 1's
 - Suppose row R_j has 1 in column C_i
 - For each h_k ,
 - if $h_k(i) < \text{slot}(C_i, h_k)$, then $\text{slot}(C_i, h_k) \leftarrow h_k(i)$

8

Example

		C ₁	C ₂		
R ₁	1	0	h(1) = 1	1	-
R ₂	0	1	g(1) = 3	3	-
R ₃	1	1	h(2) = 2	1	2
R ₄	1	0	g(2) = 0	3	0
R ₅	0	1	h(3) = 3	1	2
			g(3) = 2	2	0
			h(4) = 4	1	2
			g(4) = 4	2	0
			h(5) = 0	1	0
			g(5) = 1	2	0

$h(x) = x \bmod 5$
 $g(x) = 2x+1 \bmod 5$

9

Comparing Signatures

- **Signature Matrix S**
 - Rows = Hash Functions
 - Columns = Columns
 - Entries = Signatures
- **Compute** – Pair-wise similarity of signature columns

10