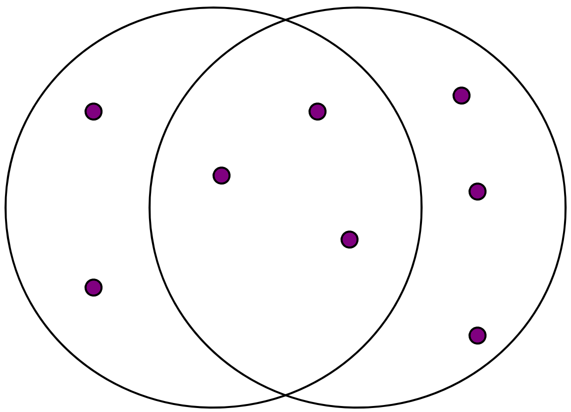
Minhashing

Jaccard Similarity Measure
Constructing Signatures

Jaccard Similarity

- The *Jaccard similarity* of two sets is the size of their intersection divided by the size of their union.
- $Sim(C_1, C_2) = |C_1 \cap C_2| / |C_1 \cup C_2|$.

Example: Jaccard Similarity



3 in intersection.
8 in union.
Jaccard similarity
 = 3/8

From Sets to Boolean Matrices

- Rows = elements of the universal set.
  - Example: the set of all k-shingles.
- Columns = sets.
- 1 in row *e* and column *S* if and only if *e* is a member of *S*.
- Column similarity is the Jaccard similarity of the sets of their rows with 1.
- Typical matrix is sparse.

Example: Column Similarity

$\underline{C_1}$	$\underline{C_2}$		
0	1		*
1	0		*
1	1	*	*
0	0		
1	1	*	*
0	1		*

$\text{Sim}(C_1, C_2) =$

$2/5 = 0.4$

Four Types of Rows

- Given columns C_1 and C_2, rows may be classified as:

	C_1	C_2
a	1	1
b	1	0
c	0	1
d	0	0

- Also, a = # rows of type a, etc.
- Note $Sim(C_1, C_2) = a/(a + b + c)$.

Minhashing

- Imagine the rows permuted randomly.
- Define *minhash function* $h(C)$ = the number of the first (in the permuted order) row in which column C has 1.
- Use several (e.g., 100) independent hash functions to create a signature for each column.
- The signatures can be displayed in another matrix – the *signature matrix* – whose columns represent the sets and the rows represent the minhash values, in order for that column.

Minhashing Example

Input matrix

1	4	3	1	0	1	0
3	2	4	1	0	0	1
7	1	7	0	1	0	1
6	3	6	0	1	0	1
2	6	1	0	1	0	1
5	7	2	1	0	1	0
4	5	5	1	0	1	0

Signature matrix *M*

2	1	2	1
2	1	4	1
1	2	1	2

Surprising Property

- The probability (over all permutations of the rows) that $h(C_1) = h(C_2)$ is the same as $Sim(C_1, C_2)$.
- Both are $a/(a+b+c)$!
- Why?
  - Look down the permuted columns C_1 and C_2 until we see a 1.
  - If it's a type-*a* row, then $h(C_1) = h(C_2)$. If a type-*b* or type-*c* row, then not.

Similarity for Signatures

- The *similarity of signatures* is the fraction of the minhash functions in which they agree.
  - Thinking of signatures as columns of integers, the similarity of signatures is the fraction of rows in which they agree.
- Thus, the expected similarity of two signatures equals the Jaccard similarity of the columns or sets that the signatures represent.
  - And the longer the signatures, the smaller will be the expected error.

Min Hashing – Example

Input matrix

1	4	3		1	0	1	0
3	2	4		1	0	0	1
7	1	7		0	1	0	1
6	3	6		0	1	0	1
2	6	1		0	1	0	1
5	7	2		1	0	1	0
4	5	5		1	0	1	0

Signature matrix *M*

2	1	2	1
2	1	4	1
1	2	1	2

	1-3	2-4	1-2
Col/Col	0.75	0.75	0
Sig/Sig	0.67	1.00	0

Implementation of Minhashing

- Suppose 1 billion rows.
- Hard to pick a random permutation of 1…billion.
- Representing a random permutation requires 1 billion entries.
- Accessing rows in permuted order leads to thrashing.

Implementation – (2)

- A good approximation to permuting rows: pick, say, 100 hash functions.
- For each column *c* and each hash function h_i, keep a "slot" $M(i, c)$.
- Intent: $M(i, c)$ will become the smallest value of $h_i(r)$ for which column *c* has 1 in row *r*.
  - I.e., $h_i(r)$ gives order of rows for i^{th} permutation.

Implementation – (3)

for each row r **do begin**
  for each hash function h_i **do**
    compute $h_i(r)$;
  for each column c
    if c has 1 in row r
      for each hash function h_i **do**
        if $h_i(r)$ is smaller than $M(i, c)$ **then**
          $M(i, c) := h_i(r)$;
end;

Example

Row	C1	C2
1	1	0
2	0	1
3	1	1
4	1	0
5	0	1

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

	Sig1	Sig2
$h(1) = 1$	1	∞
$g(1) = 3$	3	∞
$h(2) = 2$	1	2
$g(2) = 0$	3	0
$h(3) = 3$	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

Implementation – (4)

- Often, data is given by column, not row.
  - Example: columns = documents, rows = shingles.
- If so, sort matrix once so it is by row.