GOETHE UNIVERSITÄT FRANKFURT AM MAIN

www.uni-frankfurt.de

SoSe 2014: M-TANI: Big Data Analytics

Lecture 3 – 14/05/2014

Sead Izberovic

Dr. Nikolaos Korfiatis

Agenda

- Recap from the previous session
- The concept of similarity
  - Jaccard similarity
  - k-shingles
  - Min-Hashing
- Clustering (Stanford slides)
  - Distance measures
  - Hierarchical Clustering
  - Partitional Clustering

Why to use similarity

- **Finding documents with similar words**
  - Classification
  - Plagiarism
- **Finding similar users**
  - Recommendation systems
- **Finding similar images**
  - Infringement of a copyright

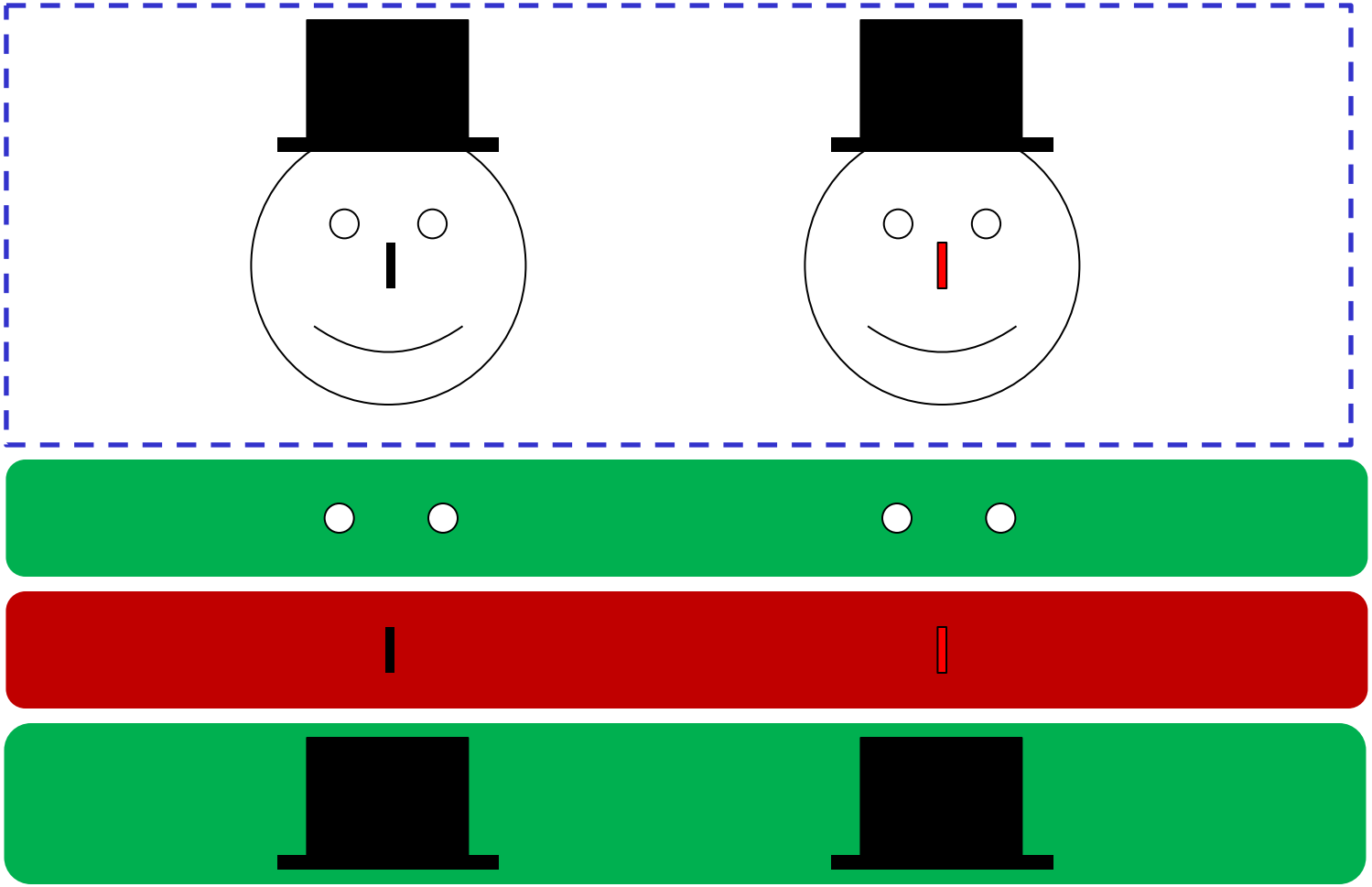
from [1]

The concept of similarity

Similarity problem

- „Many similarity problems can be described as finding a subset of some universal set that have significant intersection.” [3]

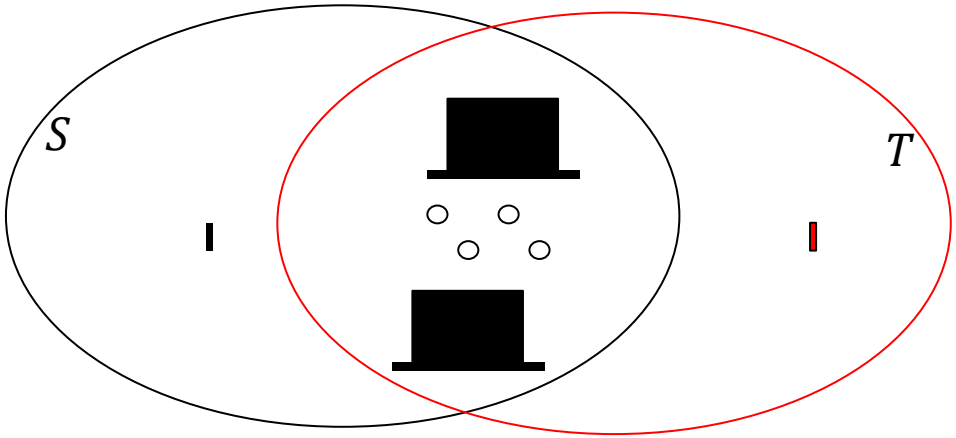
Converting a image into sets



Jaccard similarity

- Jaccard similarity is a similarity matric based on sets

$$Jaccard\ similarity = SIM(S,T) = \frac{|S \cap T|}{|S \cup T|}$$



$$SIM(S,T) = \frac{|S \cap T|}{|S \cup T|} = \frac{2}{3}$$

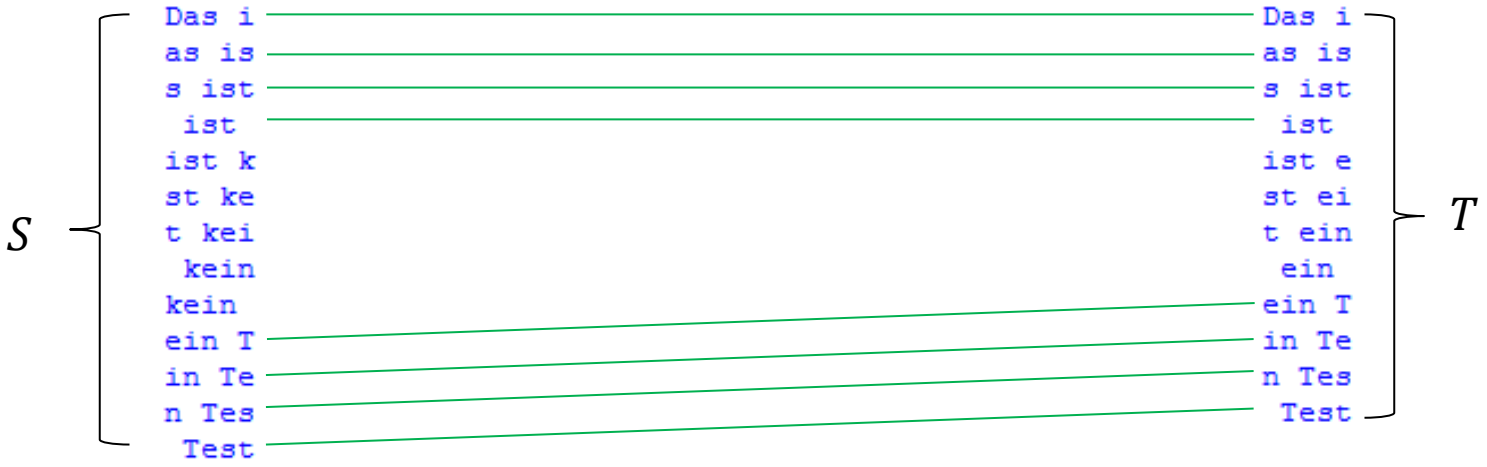
K-Shingles

- Method to represent documents as sets
- D is the document
- k is the length of the substrings

$k = 5$

$D_1 =$ Das ist kein Test

$D_2 =$ Das ist ein Test



$$SIM(S,T) = \frac{|S \cap T|}{|S \cup T|} = \frac{8}{17} = 0.471$$

adapted from [1]

K-Shingles

- **Problems for large files**
  - Is using lot of space
- **Solution**
  - Compression

from [1]

K-Shingles compression

- **Compression**
  - Hashing
- **Document representation**
  - By a set of hash values of its k-shingles
- **Calculation of the similarity**
  - Using the hash values instead of the k-shingles

from [2]

K-Shingles compression

Example

$k = 2$

$D_1 =$ abcab

$Shi(D_1) = \{ab, bc, ca\}$

$\mathrm{h}(D_1) = \{1, 5, 7\}$

from [1]

Min-Hashing

- **Converting large sets to short signatures**
- **Data represented as sparse matrices**
  - Converting sets to bit vectors

Min-Hashing

Converting sets to bit vectors

Example

$k = 2$

$D_1 =$ abcab

$Shi(D_1) = \{ab, bc, ca\}$

$Con\big(Shi(D_1)\big) = [1,1,1,0,0] = C_1$

$D_2 =$ dcabx

$Shi(D_2) = \{dc, ca, ab, bx\}$

$Con\big(Shi(D_2)\big) = [1,0,1,1,1] = C_2$

Shingels	C_1	C_2
ab	1	1
bc	1	0
ca	1	1
dc	0	1
bx	0	1

adapted from [1]

Min-Hashing

- **Jaccard similarity on vectors**

$$SIM(C_1, C_2) = \frac{x}{x + y}$$

- **Rows divided into three classes**
  - X: rows have 1 in both columns
  - Y: rows have 1 just in one of the columns
  - Z: rows have 0 in both columns

x	1	1	0
y	0	1	0
z	0	0	1

adopted from [1]

Min-Hashing

Example

x	1	1	0
y	0	1	0
y	1	0	1
y	1	0	0
	0	0	1
x	1	1	0
x	1	1	1

$$SIM(C_1, C_2) = \frac{x}{x+y} = \frac{3}{3+3} = \frac{3}{6}$$

adapted from [1]

Min-Hashing

- **Converting large sets to short signatures**
- **Data represented as sparse matrices**
  - Converting sets to bit vectors
- **Create a small hash value $\boldsymbol{h(C_i)}$ of each column $\boldsymbol{C_i}$, such that:**
  - $\boldsymbol{h(C_i)}$ is small enough to fit in RAM
  - $\boldsymbol{SIM(C_i, C_j)}$ is the nearly the same as the similarity $\boldsymbol{h(C_i)}$ and $\boldsymbol{h(C_j)}$

from [1]

Min-Hashing

- **Find a hash function $h(C_i)$ such that:**
  - if $SIM(C_i, C_j)$ is high, then with high probability $h(C_i) = h(C_j)$
  - if $SIM(C_i, C_j)$ is low, then with high probability $h(C_i) \neq h(C_j)$
- **The hash function depends on the similarity matric**
- **Hash function for the Jaccard similarity is called Min-Hashing**

from [1]

Min-Hashing

Idea

1. **Permute the rows of a boolean matrix under the random permutation π**
2. **Define a hash function $h_\pi(C_i)$ such that:**
   - $h_\pi(C_i)$ is the index of the first row where C_i has the value 1 (in the permuted order π)

from [2]

Min-Hashing

Calculating Min-Hash signatures

- $\boldsymbol{SIG(i,c)}$ is the element of the signature matrix for the $\boldsymbol{i}$th hash function and column $\boldsymbol{c}$
- Initially, set $\boldsymbol{SIG(i,c)}$ to ∞ for all $\boldsymbol{i}$ and $\boldsymbol{c}$
- We handle row r by doing the following:

1. Compute $h_1(r), h_2(r), \ldots, h_n(r)$.
2. For each column c do the following:
   (a) If c has 0 in row r, do nothing.
   (b) However, if c has 1 in row r, then for each $i = 1, 2, \ldots, n$ set $\text{SIG}(i, c)$ to the smaller of the current value of $\text{SIG}(i, c)$ and $h_i(r)$.

from [1]

Min-Hashing

Example

Row	S_1	S_2	S_3	S_4	$x+1 \mod 5$	$3x+1 \mod 5$
0	1	0	0	1	1	1
1	0	0	1	0	2	4
2	0	1	0	1	3	2
3	1	0	1	1	4	0
4	0	0	1	0	0	3

Initialization

	S_1	S_2	S_3	S_4
h_1	∞	∞	∞	∞
h_2	∞	∞	∞	∞

from [1]

Min-Hashing

Example

Row	S_1	S_2	S_3	S_4	$x+1 \mod 5$	$3x+1 \mod 5$
0	1	0	0	1	1	1
1	0	0	1	0	2	4
2	0	1	0	1	3	2
3	1	0	1	1	4	0
4	0	0	1	0	0	3

$h_1(0) = 1$ and $h_2(0) = 1$

	S_1	S_2	S_3	S_4
h_1	1	∞	∞	1
h_2	1	∞	∞	1

from [1]

Min-Hashing

Example

Row	S_1	S_2	S_3	S_4	$x+1 \mod 5$	$3x+1 \mod 5$
0	1	0	0	1	1	1
1	0	0	1	0	2	4
2	0	1	0	1	3	2
3	1	0	1	1	4	0
4	0	0	1	0	0	3

$h_1(1) = 2$ and $h_2(1) = 4$

	S_1	S_2	S_3	S_4
h_1	1	∞	2	1
h_2	1	∞	4	1

from [1]

Min-Hashing

Example

Row	S_1	S_2	S_3	S_4	$x+1 \mod 5$	$3x+1 \mod 5$
0	1	0	0	1	1	1
1	0	0	1	0	2	4
2	0	1	0	1	3	2
3	1	0	1	1	4	0
4	0	0	1	0	0	3

$h_1(2) = 3$ and $h_2(2) = 2$

Not changed!
Because 1<3 and 1<2

Changed!
Because 3<∞ and 2<∞

	S_1	S_2	S_3	S_4
h_1	1	3	2	1
h_2	1	2	4	1

from [1]

Min-Hashing

Example

Row	S_1	S_2	S_3	S_4	$x+1 \mod 5$	$3x+1 \mod 5$
0	1	0	0	1	1	1
1	0	0	1	0	2	4
2	0	1	0	1	3	2
3	1	0	1	1	4	0
4	0	0	1	0	0	3

$h_1(3) = 4$ and $h_2(3) = 0$

Not changed!
Because 1<4, 2<4 and 1<4

Changed!
Because 0<1, 0<4 and 0<1

	S_1	S_2	S_3	S_4
h_1	1	3	2	1
h_2	0	2	0	0

from [1]

Min-Hashing

Example

Row	S_1	S_2	S_3	S_4	$x+1 \mod 5$	$3x+1 \mod 5$
0	1	0	0	1	1	1
1	0	0	1	0	2	4
2	0	1	0	1	3	2
3	1	0	1	1	4	0
4	0	0	1	0	0	3

$h_1(4) = 0$ and $h_2(4) = 3$

Changed!
Because 0<2

Not changed!
Because 0<3

	S_1	S_2	S_3	S_4
h_1	1	3	0	1
h_2	0	2	0	0

from [1]

Min-Hashing

Example

Row	S_1	S_2	S_3	S_4	$x+1 \mod 5$	$3x+1 \mod 5$
0	1	0	0	1	1	1
1	0	0	1	0	2	4
2	0	1	0	1	3	2
3	1	0	1	1	4	0
4	0	0	1	0	0	3

Final signature matrix

	S_1	S_2	S_3	S_4
h_1	1	3	0	1
h_2	0	2	0	0

from [1]

Min-Hashing properties

- **Claim:** $\mathbf{P}\left[\boldsymbol{h_\pi(C_i) = h_\pi(C_j)}\right] = \boldsymbol{SIM(C_i, C_j)}$

- Let X be a set of shingles, $y \in X$ is a shingle
- Then $\mathrm{P}\left[\pi(y) = min(\pi(y))\right] = \frac{1}{|X|}$
  - It is equally likely that any $y \in X$ is mapped to the min element
- Let y be $\pi(y) = min(\pi(C_i \cup C_j))$
- Then either: $\pi(y) = min(\pi(C_i))$ if $y \in C_i$, or
  $\pi(y) = min(\pi(C_j))$ if $y \in C_j$
- So the prob. that **both** are true is the prob. $y \in C_i \cap C_j$

- $\boldsymbol{P\left[min(\pi(C_i)) = min(\pi(C_j))\right] = \frac{|C_i \cap C_j|}{|C_i \cup C_j|} = SIM(C_i, C_j)}$

from [2]

Min-Hashing

Example

Row	S_1	S_2	S_3	S_4	$x+1 \mod 5$	$3x+1 \mod 5$
0	1	0	0	1	1	1
1	0	0	1	0	2	4
2	0	1	0	1	3	2
3	1	0	1	1	4	0
4	0	0	1	0	0	3

Signature matrix:

	S_1	S_2	S_3	S_4
h_1	1	3	0	1
h_2	0	2	0	0

Similarities:

Documents:	1-2	1-3	1-4
Col/Col:	0	0.25	0.67
Sig/Sig:	0	0	1

from [1]

Literatur

1. Anand Rajaraman, Jeffrey D. Ullman, Jure Leskovec. 2014
   Mining of Massive Datasets
   Cambridge University Press
2. Jure Leskovec. 2009
   Slides: **Mining Massive Data Sets**
   URL: *http://www.stanford.edu/class/cs246/slides/03-lsh.pdf*
3. Anand Rajaraman. 2009
   Slides: **Data Mining**
   URL: *http://infolab.stanford.edu/~ullman/mining/2009/similarity1.ppt*