

Web Mining
Exercises

Mauro Brunato, Elisa Cilia

May 18, 2011

Exercise 1

A corpus contains the following five documents:

d_1	To be or not to be, this is the question!
d_2	I have a pair of problems for you to solve today.
d_3	It's a long way to Tipperary, it's a long way to go. . .
d_4	I've been walking a long way to be here with you today.
d_5	I am not able to question these orders.

The indexing system only considers nouns, adjectives, pronouns, adverbs and verbs. All forms are converted to singular, verbs are converted to the infinitive tense, removes all punctuation marks and translates all letters to uppercase. Conjunctions, prepositions, articles and exclamations are discarded as well. Multiple occurrences of the same term within a document are not counted.

For instance, the phrase

Hey, it's not too late to solve these exercises!

becomes

IT BE NOT TOO LATE SOLVE THIS EXERCISE

- 1.1) What is the minimum dimension (number of coordinates) of the TFIDF vector space for this collection of documents?
- 1.2) Fill the 5×5 matrix of Jaccard coefficients between all pairs of documents.
- 1.3) Apply an agglomerative clustering procedure to the collection. as a measure of similarity between two clusters D_1 and D_2 , consider the highest similarity between d_1 and d_2 , with $d_1 \in D_1$ and $d_2 \in D_2$.
- 1.4) Draw the resulting dendrogram.

Solution — The stripped-down documents are the following (the third columns count the number of different terms in each document, just to ease up the calculation of the Jaccard coefficient):

d_1	BE NOT THIS QUESTION	4
d_2	I HAVE PAIR PROBLEM YOU SOLVE TODAY	7
d_3	IT BE LONG WAY TIPPERARY GO	6
d_4	I HAVE BE WALK LONG WAY HERE YOU TODAY	9
d_5	I BE NOT ABLE QUESTION THIS ORDER	7

1.1) The collection includes 20 different terms: ABLE, BE, GO, HAVE, I, IT, HERE, LONG, NOT, ORDER, PAIR, PROBLEM, QUESTION, SOLVE, THIS, TIPPERARY, TODAY, WALK, WAY, and YOU. Therefore, the vector representation requires at least 20 dimensions.

1.2) The table of Jaccard coefficients is the following. Only the upper triangular part is shown, since the Jaccard coefficient is symmetrical.

	d_1	d_2	d_3	d_4	d_5
d_1	1	0	1/9	1/12	4/7
d_2		1	0	1/3	1/13
d_3			1	1/4	1/12
d_4				1	1/7
d_5					1

1.3) The two most similar documents are d_1 and d_5 , so they can be joined in the same partition. The similarity matrix becomes:

	$\{d_1, d_5\}$	$\{d_2\}$	$\{d_3\}$	$\{d_4\}$
$\{d_1, d_5\}$	1	1/13	1/9	1/7
$\{d_2\}$		1	0	1/3
$\{d_3\}$			1	1/4
$\{d_4\}$				1

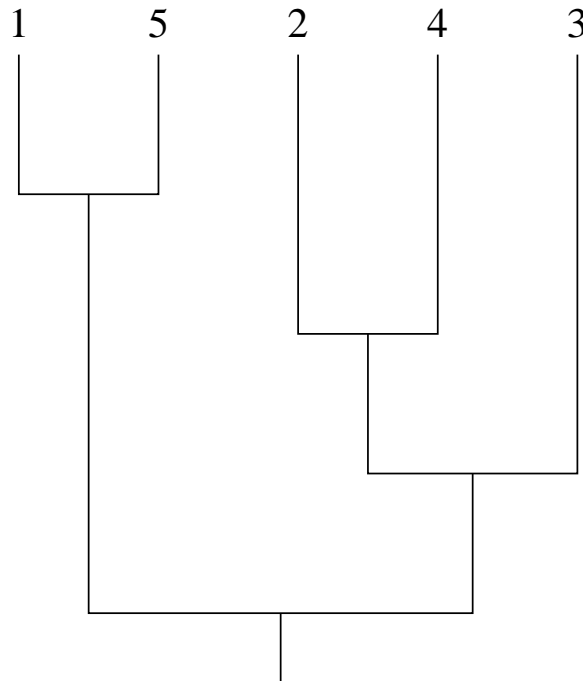
After this step, singletons $\{d_2\}$ and $\{d_4\}$ are most similar, and shall be joined:

	$\{d_1, d_5\}$	$\{d_2, d_4\}$	$\{d_3\}$
$\{d_1, d_5\}$	1	1/7	1/9
$\{d_2, d_4\}$		1	1/4
$\{d_3\}$			1

Next, singleton d_3 joins cluster $\{d_2, d_4\}$:

	$\{d_1, d_5\}$	$\{d_2, d_3, d_4\}$
$\{d_1, d_5\}$	1	1/7
$\{d_2, d_3, d_4\}$		1

Finally, the two remaining clusters can be merged together. The corresponding dendrogram is the following:



Exercise 2

In the same setting as in the previous exercise, estimate the Jaccard coefficient for all document pairs based on the application of five random permutations.

Exercise 3

Let D be a set of documents over the set T of terms, n_{td} counts the number of occurrences of term t in document d .

3.1) Consider the following term frequency measures:

$$A_1(t, d) = n_{td}, \quad A_2(t, d) = \begin{cases} 1 & \text{if } n_{td} \neq 0 \\ 0 & \text{otherwise,} \end{cases} \quad A_3(t, d) = \frac{n_{td}}{|d|}, \quad A_4(t, d) = \log(1 + n_{td}).$$

Consider each measure according to each of the following criteria *separately*:

1. The size of a document should not matter (e.g., concatenating two copies of the same document should not change the measure).
2. The number of occurrences of the term should not matter, only its presence is important.
3. Increasing the number of occurrences of a term should have a lesser impact on the measure if the term is already frequent.

3.2) Which of the following are suitable IDF functions, and why?

$$B_1(t) = -\log \left(1 - \left(\sum_{d \in D} A_1(t, d) \right)^{-1} \right), \quad B_2(d) = \left(1 + \sum_{t \in T} A_2(t, d) \right)^{-1},$$

$$B_3(t) = \sum_{d \in D} \frac{1}{1 + A_1(t, d)}, \quad B_4(d) = \left(\sum_{t \in T} A_4(t, d) \right)^{-1}$$

Exercise 4

A document retrieval system must be implemented in a structured programming language (Java, C, C++). Documents and terms are represented with their numeric IDs.

4.1) Define the appropriate array and record structures to efficiently store the matrix n_{td} counting the number of occurrences of each term t in each document d , considering that it is very sparse. Define the structure to store inverse document frequency values.

4.2) Write a function `retrieve(q)` which, given the array q of term indices, returns an array with the IDs of the five nearest documents according to the cosine measure in the TFIDF space.

Exercise 5

An information retrieval system manages a corpus of six documents. Given the query q , the system computes the following probabilities for the documents to be relevant:

i	1	2	3	4	5	6
p_i	100%	80%	20%	80%	0	100%

5.1) What strategy can the system adopt in order to maximize its recall score? What strategy can maximize its precision score?

5.2) Suppose that the only documents that are relevant with respect to query q are 1, 2, 4 and 6 (of course, the system does not know this). The system implements two alternative algorithms:

1. let document i appear in the returned list iff $p_i = 100\%$, or
2. let document i appear in the list with probability p_i .

Compute the expected values of precision and recall assigned by the user (who knows the actual document relevance) to the list of documents returned by each algorithm.

Hint — Note that algorithm (1) is deterministic, only algorithm (2) is stochastic.

Solution —

5.1) Let $r = (r_i)$, where r_i is the “true” relevance of document i (remember that the query is fixed). Let $x = (x_i)$,

where $x_i = 1$ iff the IR system returns document i in response to the query. Then,

$$\text{Precision}_r(\mathbf{x}) = \frac{\mathbf{x} \cdot \mathbf{r}}{\sum_{i=1}^6 x_i}, \quad \text{Recall}_r(\mathbf{x}) = \frac{\mathbf{x} \cdot \mathbf{r}}{\sum_{i=1}^6 r_i}.$$

In other words, the “precision” of the answer is the amount of relevant documents within the list provided by the IR system. Its maximum value is attained when all returned documents are relevant, so we need to return only the two documents, 1 and 6, which are certainly relevant to the user. The “recall” of the answer is its property of containing as many relevant documents as possible, and it is maximized by returning all documents (with the possible exception of 5, which is irrelevant for sure).

5.2) In the first case, the IR system provides a deterministic answer, having precision 100% and recall 50%. In the second case, we need to compute precision and recall scores for all possible return strings, and compute their probability-weighted average:

$$E(\text{Precision}) = \sum_{\mathbf{x}} \Pr(\mathbf{x}) \text{Precision}_r(\mathbf{x}), \quad E(\text{Recall}) = \sum_{\mathbf{x}} \Pr(\mathbf{x}) \text{Recall}_r(\mathbf{x}).$$

Note that documents 1 and 6 are always returned, while document 5 is never returned; moreover, documents 2 and 4 are indistinguishable, so we can determine the following table, where precision (left) and recall (right) scores are provided together with their probabilities (in parentheses).

		$x_2 + x_4$					
		0 (.04)		1 (.32)		2 (.64)	
x_3	0 (.8)	$\frac{2}{2} \frac{2}{4}$ (.032)	$\frac{3}{3} \frac{3}{4}$ (.256)	$\frac{4}{4} \frac{4}{4}$ (.512)			
	1 (.2)	$\frac{2}{3} \frac{2}{4}$ (.008)	$\frac{3}{4} \frac{3}{4}$ (.064)	$\frac{4}{5} \frac{4}{4}$ (.128)			

Finally,

$$E(\text{Precision}) = .8 + \frac{2}{3} \cdot .008 + \frac{3}{4} \cdot .064 + \frac{4}{5} \cdot .128 \approx .8 + .005 + .048 + .102 \approx 96\%,$$

$$E(\text{Recall}) = \frac{2}{4} \cdot .04 + \frac{3}{4} \cdot .32 + \frac{4}{4} \cdot .64 = .02 + .24 + .64 = 90\%.$$

Exercise 6

With the same data of Exercise 5, suppose that the system uses algorithm (1).

6.1) Compute the expected precision and recall scores from the point of view of the IR system, who only knows the probabilities p_i for document i to be relevant.

Solution — In this case the IR system’s answer is known, but the actual document relevance is a random variable with the given probabilities. Therefore, the average values must be computed against probabilities of the unknown \mathbf{r} :

$$E_r(\text{Precision}) = \sum_{\mathbf{r}} \Pr(\mathbf{r}) \text{Precision}_r(\mathbf{x}), \quad E_r(\text{Recall}) = \sum_{\mathbf{r}} \Pr(\mathbf{r}) \text{Recall}_r(\mathbf{x}).$$

We know the answer \mathbf{x} of the IR system, which is $(1, 0, 0, 0, 0, 1)$, therefore we can compute a table which is similar to that of Exercise 5:

		$x_2 + x_4$					
		0 (.04)		1 (.32)		2 (.64)	
x_3	0 (.8)	$\frac{2}{2} \frac{2}{2}$ (.032)	$\frac{2}{2} \frac{2}{3}$ (.256)	$\frac{2}{2} \frac{2}{4}$ (.512)			
	1 (.2)	$\frac{2}{2} \frac{2}{3}$ (.008)	$\frac{2}{2} \frac{2}{4}$ (.064)	$\frac{2}{2} \frac{2}{5}$ (.128)			

Therefore, as expected,

$$E_r(\text{Precision}) = 100\%$$

because we are sure that only relevant documents are returned. On the other hand,

$$E_r(\text{Recall}) = \frac{2}{2} \cdot .032 + \frac{2}{3} \cdot .256 + \frac{2}{4} \cdot .512 + \frac{2}{3} \cdot .008 + \frac{2}{4} \cdot .064 + \frac{2}{5} \cdot .128 \approx .032 + .171 + .256 + .005 + .032 + .051 \approx 55\%.$$

Exercise 7

Write in your favorite high-level language a function that implements the FastMap algorithm. In particular, define what input must be provided and which output shall be returned.

Solution — Let matrix d be the input data (mutual distances between couples of items). The matrix is symmetric, so many optimizations are possible. Let x be the output matrix with one column per document and one row per extracted coordinate. We assume that the number of documents n and the number of extracted dimensions m are encoded into matrix sizes; otherwise, we can pass them as two additional integer parameters.

```

1. void FastMap (double d[], double x[])
2. {
3.     int n = d.length, m = x.length;
4.     for (int s = 0; s < m; s++) {
5.         i, j ← arg maxi,j d[i][j];
6.         for (int k = 0; k < n; k++)
7.             x[s][k] ←  $\frac{d[i][k]^2 + d[j][k]^2 - d[i][j]^2}{2d[i][j]}$ ;
8.         for (int i1 = 0; i1 < n; i1++)
9.             for (int j1 = 0; j1 < n; j1++)
10.                d[i1][j1] ←  $\sqrt{d[i_1][j_1]^2 - (x[s][i_1] - x[s][j_1])^2}$ ;
11.     }
12. }
```

Repeat for the desired number of coordinates
Find the two farthest points
Compute the s-th coordinate

Recompute the mutual distances

Note that the term within the square root sign at line 10 might be negative, so a bit of care must be taken when actually implementing the algorithm...

Exercise 8

The columns of the following matrix represent the coordinates of a set of documents in a TFIDF space:

$$A = \frac{1}{\sqrt{6}} \begin{pmatrix} 2 & 0 & 2 \\ 0 & 1 & 0 \\ 2 & 1 & 2 \\ 2 & -1 & 2 \end{pmatrix}$$

Let document similarity be defined by the cosine measure (dot product).

8.1) Compute the rank of matrix A .

8.2) Let $\mathbf{q} = (1, 3, 0, -2)^T$ be a query. Find the document in the set that best satisfies the query.

8.3) Given the matrices

$$U = \alpha \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad V = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$

determine coefficient α and the diagonal matrix Σ so that U is column-orthonormal and $A = U\Sigma V^T$.

8.4) Project the query \mathbf{q} onto the LSI space defined by this decomposition and verify the answer to question 8.2. Why isn't the requirement that V be column-orthonormal important in our case?

8.5) Suppose that we want to reduce the LSI space to one dimension. Show how the new approximate document similarities to \mathbf{q} are computed.

Solution —

8.1) Notice that A has two linearly dependent (actually equal) columns (thus $\text{rk } A < 3$), while the first two columns are independent (thus $\text{rk } A \geq 2$), therefore $\text{rk } A = 2$.

8.2) Similarities are computed by dot products, let's do it in a single shot for all documents:

$$A^T \mathbf{q} = \frac{1}{\sqrt{6}} \begin{pmatrix} -2 \\ 5 \\ -2 \end{pmatrix};$$

The most similar is document 2.

8.3) The column normality condition for matrix U implies $\beta = 1/\sqrt{3}$. By expliciting the calculation of some entries of matrix A , we obtain

$$\Sigma = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}.$$

8.4) Projection onto the document LSI space is achieved via $\Sigma^{-1}U^T$:

$$\hat{\mathbf{q}} = \Sigma^{-1}U^T \mathbf{q} = \frac{1}{\sqrt{3}} \begin{pmatrix} -1/2 \\ 5 \end{pmatrix}.$$

Similarity to the documents is computed via the $V\Sigma^2$ matrix. If all computations are right,

$$V\Sigma^2\hat{\mathbf{q}} = A^T \mathbf{q}.$$

Exercise 9

Specify in the MapReduce framework the Map and Reduce functions to find the number of occurrences of one/more given pattern/s in a collection of documents.

Solution — Let us define the two functions.

$$\begin{array}{lll} \text{Map:} & \mathbb{N} \times T^* & \longrightarrow (T \times \mathbb{N})^* \\ & (\text{offset, line}) & \mapsto [(\text{match}, 1)] \\ \text{Reduce:} & T \times \mathbb{N}^* & \longrightarrow (T \times \mathbb{N})^* \\ & (\text{match}, [n_1, \dots, n_k]) & \mapsto [(\text{match}, \sum n_i)] \end{array}$$

Function *Map* receives a key (related to the document ID or line offset), which we can disregard, and a sequence of terms (a line or a full document). It gives as output a list of pairs (match, 1), one for each match of the pattern in the received value.

Function *Reduce* takes as input a pair (match, [n₁, ..., n_k]) where the value part is a list of previously computed occurrences (originally all 1's) and returns the list of matching patterns (only one element in this case) with the number of occurrences for each match.

The pseudo-code for the Map and Reduce functions is the following:

1. map (*offset, line*)
2. $\left[\begin{array}{l} \text{while } \text{pattern.matches}(\text{line}) \\ \text{emit}(\text{pattern}, 1); \end{array} \right.$
3. $\left. \right]$

1. reduce (*match, values*)
2. $\left[\begin{array}{l} \text{result} = 0; \\ \text{for each } v \text{ in } \text{values} \\ \text{result} += v; \\ \text{emit}(\text{match}, \text{result}); \end{array} \right.$
3. $\left. \right]$
4. $\left. \right]$
5. $\left. \right]$

values is an iterator over counts

Exercise 10

Given the following relevance ranking vector in response to a query q :

$$d_1, \underline{d_2}, \underline{d_3}, d_4, \underline{d_5}, \underline{d_6}$$

(the underlined documents are exactly all the relevant ones)

10.1) Determine the interpolated precision at level $\rho = 0.5$ of recall,

10.2) Determine the “global” F_1 – *measure* (for the system returning all the six documents),

10.3) Determine the Break Even Point (BEP), which is the point of equivalence between (interpolated) precision and recall.

Solution — 10.1) We are given a ranked list of documents returned in response to a query q with their associated relevance values. In this ranked retrieval context, precision and recall can be computed by considering as the set of retrieved documents the top k ranked documents:

k	r_k	R	P
0	0	0	1
1	0	0	0
2	1	0.25	0.5
3	1	0.5	0.66
4	0	0.5	0.5
5	1	0.75	0.6
6	1	1	0.66

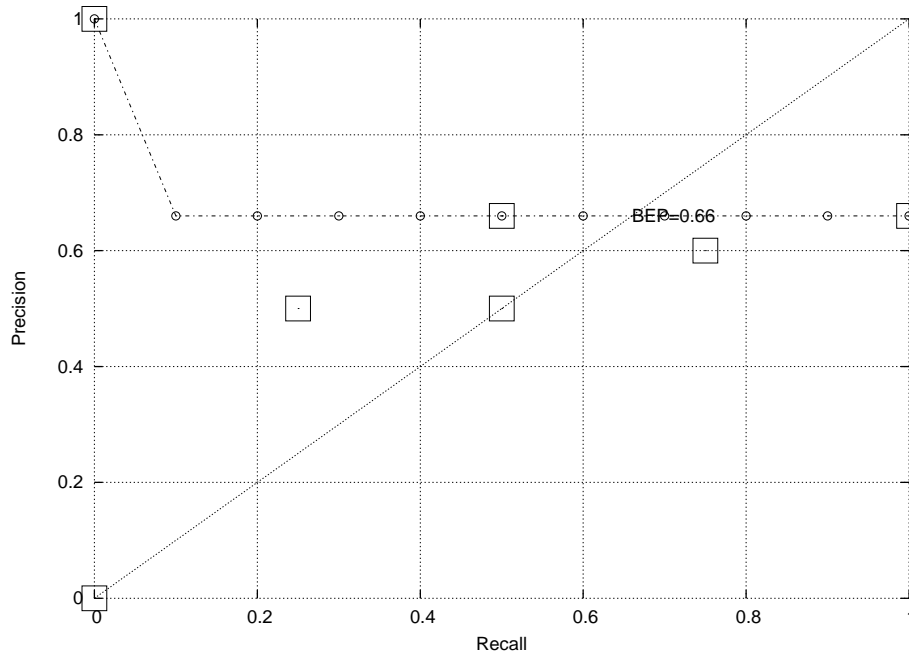
The interpolated precision P_{interp} at a certain level ρ of recall is defined as the highest precision found for any recall level $\rho' \geq \rho$:

$$P_{interp}(\rho) = \max_{\rho' \geq \rho} P(\rho')$$

Thus the interpolated precision at level $\rho = 0.5$ of recall is $P_{interp}(0.5) = 0.66$

10.2) The F_1 – *measure* = $\frac{2 \times P \times R}{P + R}$ when all the documents are returned is $F_1 = 0.795$ (by taking the P and R values computed in the last row of the table).

10.3) To find the BEP we plot the *interpolated precision curve*:



Exercise 11

The singular value decomposition of a term-document matrix $A = U\Sigma V^T$ is

$$U = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & \sqrt{2} \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad V = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 0 & -1 \\ 1 & 1 & 1 \\ 0 & 1 & -1 \\ 1 & -1 & 0 \end{pmatrix}$$

- 11.1) What is the rank of the matrix A ?
- 11.2) Perform a reduction of the LSI space by one dimension.
- 11.3) Given the new representation of matrix \hat{A} , apply an agglomerative clustering procedure to the collection. Merge the clusters at each step according to the *self-similarity* measure by using as a measure of inter-document similarity simply the dot-product $\langle d_1, d_2 \rangle$.
- 11.4) Draw the resulting dendrogram. How many clusters can you find at a level of similarity of 2?
- 11.5) Check the clustering results you get by cutting across the dendrogram, by plotting them.

Solution —

11.1) Matrix A was originally a 3×4 matrix. The three elements in the diagonal matrix Σ are non-null, therefore matrix $A^T A$ (hence, matrix A) has full rank (3).

11.2) Let us obtain \hat{U} , \hat{V} and $\hat{\Sigma}$ by removing the third column from U and V , and the third row and column from Σ , corresponding to the smallest eigenvalue of $A^T A$:

$$\hat{U} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \\ 0 & 0 \end{pmatrix}, \quad \hat{\Sigma} = \begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix}, \quad \hat{V} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & -1 \end{pmatrix}.$$

11.3) After rank reduction, we can compute the similarity by

$$\hat{A}^T \hat{A} = \hat{V} \hat{\Sigma}^2 \hat{V}^T = \frac{1}{3} \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 9 & 0 \\ 0 & 4 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & -1 \end{pmatrix}$$

Therefore, we obtain the following table of unnormalized dot product similarities:

	2	3	4
1	3	0	3
2		$\frac{4}{3}$	$\frac{5}{3}$
3			$-\frac{4}{3}$

11.4) $\{1, 2\}$ and $\{1, 4\}$ are both candidates as the first cluster. Let us choose the first pair. Therefore, at level 3 the first clustering step yields

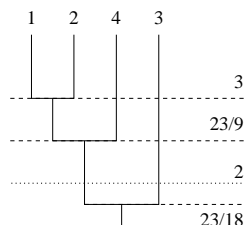
	3	4
$\{1, 2\}$	$\frac{13}{9}$	$\frac{23}{9}$
3		$-\frac{4}{3}$

Now, the highest self-similarity value is achieved by cluster $\{1, 2, 4\}$ at level $\frac{23}{9}$, so that the similarity matrix becomes

	3
$\{1, 2, 4\}$	$\frac{23}{18}$

Therefore, at similarity level 2 we have two clusters: $\{1, 2, 4\}$ and 3.

11.5) The corresponding dendrogram is



Exercise 12

Consider the query: “love Mary” and the set of returned documents of a information retrieval system:

- d1: John gives a book to Mary.
- d2: John who reads a book loves Mary.
- d3: Whom does John think Mary loves?
- d4: John thinks a book is a good gift.

12.1) Define the set of keywords and give documents the corresponding representation, after applying the stop word elimination and the stemming processes. Assume that we are using direct term frequency (with no scaling and no document frequency). Do not normalize vectors.

12.2) Suppose that document 2 has been judged as relevant, and document 4 as not-relevant. Using Rocchio relevance feedback what would the revised query vector be after relevance feedback?

Assume $\alpha = 1$, $\beta = 0.5$, $\gamma = 0.5$.

Exercise 13

A large set of documents, each containing a large number of terms, is given. The aim of this exercise is to create an index that maps each term to the document where it occurs in the earliest position (ties may be broken at will). For example, given the three following documents:

Filename	Content
random.doc	Zigzag bumblebee slash acorn
nonsense.txt	Bumblebee acorn zigzag slash
useless.pdf	Acorn dot bumblebee slash zigzag

the index should be:

```

acorn ↦ useless.pdf
bumblebee ↦ nonsense.txt
dot ↦ useless.pdf
slash ↦ random.doc
zigzag ↦ random.doc

```

In fact, the word “bumblebee” appears in position 2 of file random.doc, in position 1 of file nonsense.txt and in position 3 of file useless.pdf, therefore it is mapped to nonsense.txt.

13.1) Outline a MapReduce-based solution to the problem. In particular, describe the input and output records of the mapper and reducer functions.

13.2) Could a combiner be useful? Provide a short motivation to your answer.

13.3) Implement the mapper and the reducer; assume that the data are already tokenized and use any language or high-level pseudo-code.

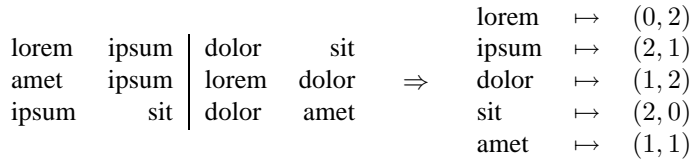
Exercise 14

A directed graph is a pair $G = (V, E)$, where V , the *vertex set* is a finite set of terms, and $E \subseteq V \times V$ is the *edge set*. The *indegree* of a vertex $v \in V$ is the number of incoming edges (the cardinality of $E \cap (V \times \{v\})$), while its *outdegree* is the number of outgoing edges (the cardinality of $E \cap (\{v\} \times V)$).

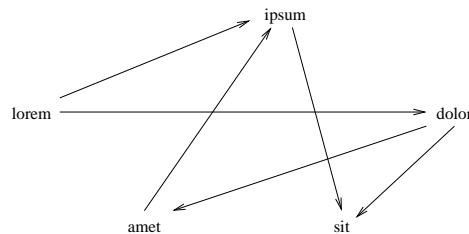
An edge in E can be represented by a line of text containing two terms (the first is the origin, the second the destination of the edge). The edge set E is therefore represented by a collection of lines of text.

Given a collection of lines describing the edge set E (either coming from a single file or split among several files), we want to design a MapReduce system to produce a list of vertices, each associated with a pair of integers representing their indegree and outdegree.

For example, consider the set of lines on the left. The resulting mapping is shown on the right.



The corresponding graph is the following:



14.1) What are the domain and codomain of the Map and Reduce functions? Is it possible to use the Reduce function as a combiner as well?

14.2) Write a pseudo-code implementation of the relevant functions.

Exercise 15

Below is a table showing how a human judge rated the relevance of a set of 15 documents with respect to a particular information need (0 = nonrelevant, 1 = relevant).

docID	relevance
1	0
2	1
3	1
4	1
5	0

docID	relevance
6	1
7	0
8	0
9	0
10	1

docID	relevance
11	1
12	1
13	1
14	1
15	1

Let us assume that two different information retrieval engines (S1 and S2) compute for this query the following rankings:

S1: (5, 8, 9, 1, 3, 4, 2, 10, 12, 13, 15, 6, 7, 14, 11) and S2: (7, 8, 1, 10, 12, 2, 3, 5, 13, 15, 14, 4, 6, 9, 11).

15.1) Does intuition suggest that one of the two IR system is better than the other?

Show that your guess is right by analysing the performance of the two systems and by comparing them.

Use the most suitable performance measures and methods among those we have seen during the course, giving as much evidence as you can.

15.2) Suppose the IR engine can return only the first 8 documents to the user. Compare the performance of the systems in this case.

Which IR system is the best? Justify your answer.

Exercise 16

Consider the following set of documents, where the vocabulary is composed of three words and we have two categories A and B :

$(5, 6, 0) \mapsto A$
 $(2, 1, 3) \mapsto A$
 $(7, 7, 0) \mapsto A$
 $(2, 2, 5) \mapsto A$
 $(0, 8, 4) \mapsto B$
 $(2, 0, 8) \mapsto B$
 $(7, 1, 3) \mapsto B$

16.1) Perform one iteration of the k -means algorithm assuming that the initial clustering corresponds to the provided categorization. Show the final clustering.

16.2) Suppose that the document set is used as a training set for a supervised k -Nearest-Neighbors classifier. Given the new document $(4, 4, 1)$, how would the classifier categorize it for $k = 1$? What for $k = 3$?

Exercise 17

Below is a table showing how two human judges rated the relevance of a set of 12 documents to a particular information need ($0 = \text{nonrelevant}$, $1 = \text{relevant}$).

Let us assume that you have written an information retrieval engine that for this query returns the set of documents $\{4, 5, 6, 7, 8\}$.

docID	Judge1	Judge2
1	0	0
2	0	0
3	1	1
4	1	1
5	1	0
6	1	0
7	1	0
8	1	0
9	0	1
10	0	1
11	0	1
12	0	1

17.1) Calculate the precision and recall of your system if a document is considered relevant only if both judges agree.

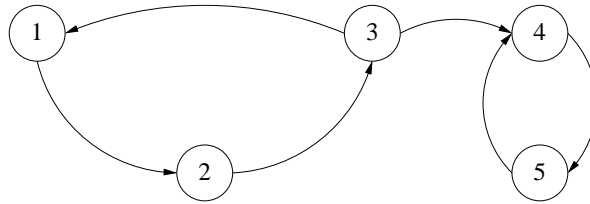
17.2) Calculate the precision and recall of your system if a document is considered relevant if either judge thinks it is relevant.

17.3) Suppose the documents are returned by your IR engine in the ID order as in the table.

- Plot the Precision versus Recall graph for the first case (a document is considered relevant only if both judges agree) when varying the number of documents returned (1 document returned, 2 documents returned, etc).
- Determine the interpolated precision at level $\rho = 0.5$ of recall.
- How many documents should the system return in order to maximize its performance? Justify your answer.

Exercise 18

The network of references for a set of five hypertexts is given in figure:

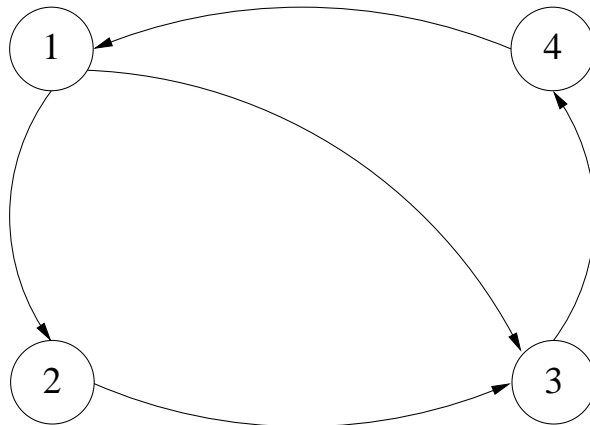


Compute the first 5 iterations of the PageRank and HITS algorithms in the following hypotheses:

- No damping factor.
- Initial PageRank vector gives probability 1 to node 1.
- Initial hub and authority vectors are uniformly 1 over all nodes.
- No normalization required.

Exercise 19

The network of references for a set of four hypertexts is given in figure:



19.1) Execute the first four steps of the PageRank algorithm starting from user being with certainty at node 1 (no damping factor).

19.2) Compute the stationary PageRank scores of the documents.

Exercise 20

Suppose that a query, executed on the same network as Exercise 19, returns nodes 1 and 2, as the *root set* and that we want to use the HITS algorithm in order to rank the pages.

20.1) Define the expanded set and the base set for the given query.

20.2) Compute the first five iterations of the HITS algorithm for the base set.

20.3) Which hub and authority values will asymptotically dominate?

Exercise 21

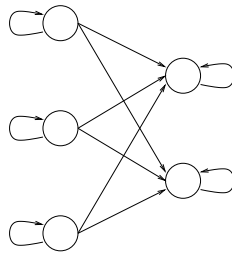
Let a hypertext system be a complete bipartite graph with 3 hubs and 2 authorities.

21.1) For every node in the system, draw a link from the node to itself. Write the adjacency matrix of the system, and normalize it for use with the PageRank algorithm.

21.2) What is the PageRank score of the nodes in the system? Provide both an analytical proof and an intuitive explanation. Assume no damping factor.

21.3) Now add a link from one of the authorities to one of the hubs. What is the PageRank score of the nodes now? Provide both an analytical proof and an intuitive explanation.

Solution — **21.1)** The requested graph is the following:



The adjacency matrix, assuming that hubs are the first three nodes, and its normalized version is

$$E = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad L = \begin{pmatrix} 1/3 & 0 & 0 & 1/3 & 1/3 \\ 0 & 1/3 & 0 & 1/3 & 1/3 \\ 0 & 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

21.2) We must find the principal eigenvector, corresponding to eigenvalue 1 of matrix E^T :

$$L^T \mathbf{v} = \mathbf{v}$$

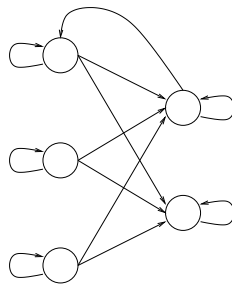
Let us make the system explicit:

$$\begin{aligned} v_1 &= \frac{1}{3}v_1 \\ v_2 &= \frac{1}{3}v_2 \\ v_3 &= \frac{1}{3}v_3 \\ v_4 &= \frac{1}{3}(v_1 + v_2 + v_3) + v_4 \\ v_5 &= \frac{1}{3}(v_1 + v_2 + v_3) + v_5 \end{aligned}$$

Therefore, principal eigenvectors are of the form $(0, 0, 0, v_4, v_5)$. The vector must be normalised, so that $v_4 + v_5 = 1$, and by symmetry considerations we get the final score: $(0, 0, 0, 1/2, 1/2)$.

By intuition, after one step (at most) the user will be trapped in one of the authorities, and will never go back; thus, the PageRank score of the hubs is 0 (after a transient period the user will never visit them). By symmetry, the probability of the user being in any authority is equal.

21.3) The graph becomes:



corresponding to the following adjacency matrix (on the right, the normalized version):

$$E' = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad L' = \begin{pmatrix} 1/3 & 0 & 0 & 1/3 & 1/3 \\ 0 & 1/3 & 0 & 1/3 & 1/3 \\ 0 & 0 & 1/3 & 1/3 & 1/3 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The eigenvector equation

$$L'^T \mathbf{v} = \mathbf{v}$$

becomes:

$$\begin{aligned} v_1 &= \frac{1}{3}v_1 + \frac{1}{2}v_4 \\ v_2 &= \frac{1}{3}v_2 \\ v_3 &= \frac{1}{3}v_3 \\ v_4 &= \frac{1}{3}(v_1 + v_2 + v_3) + \frac{1}{2}v_4 \\ v_5 &= \frac{1}{3}(v_1 + v_2 + v_3) + v_5 \end{aligned}$$

Therefore, principal eigenvectors are of the form $(0, 0, 0, 0, v_5)$, and by normalization we get the final score: $(0, 0, 0, 0, 1)$.

By intuition, sooner or later the user will be trapped in the pure authority, and will never go out.

Exercise 22

Let V_1 and V_2 be two finite sets. Then the set of edges in a complete directed bipartite graph having V_1 as source nodes and V_2 as destination nodes is the Cartesian product of the two sets:

$$V_1 \times V_2 = \{(i, j) : i \in V_1 \wedge j \in V_2\}$$

Let us define graph $G = (V, E)$ where:

$$\begin{aligned} V &= \{1, \dots, 12\} \\ E &= (\{1, 2, 3\} \times \{4, 5, 6\}) \cup (\{5, 6\} \times \{7, 8\}) \cup (\{9, 10\} \times \{11, 12\}). \end{aligned}$$

The three subsets of E identify three bipartite components of G :

$$\begin{aligned} G_1 &= (\{1, \dots, 6\}, \{1, 2, 3\} \times \{4, 5, 6\}) \\ G_2 &= (\{5, \dots, 8\}, \{5, 6\} \times \{7, 8\}) \\ G_3 &= (\{9, \dots, 12\}, \{9, 10\} \times \{11, 12\}) \end{aligned}$$

Note that the three components are not disjoint, but the graph is not connected.

For every node n , according to the HITS scoring system, let $h(n)$ be its *hub score* and let $a(n)$ be its *authority score*. Moreover, if $B = (V_B, E_B)$ is a bipartite graph, its *importance* $I(B)$ as the sum of hub scores of its source nodes plus the sum of the authority scores of its destination nodes:

$$I(B) = \sum_{i:\exists j(i,j)\in E_B} h(i) + \sum_{i:\exists j(j,i)\in E_B} a(i).$$

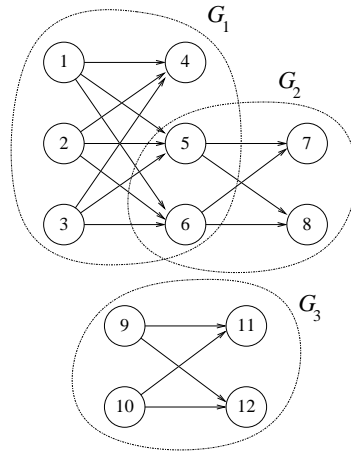
22.1) Which bipartite component (among G_1 , G_2 and G_3) will asymptotically achieve the maximum importance, and why?

22.2) Simulate three iterations of the HITS system starting with a uniform value of 1 to all hub and authority scores. What is the importance of each bipartite component, at the end?

22.3) If the edge $(3, 9)$ is added to G , how do you expect the importance scores of the three components to change, and why?

22.4) With the further addition of edge $(10, 3)$ to the graph, how do you expect the importance scores of the three components to change, and why?

Solution — The initial graph is the following (the three bipartite components are also shown):



22.1) The HITS ranking system favors the largest bipartite component, which corresponds to the principal eigenvector of $E^T E$. Therefore, component G_1 will asymptotically prevail.

22.2) Authority scores:

Node	1	2	3	4	5	6	7	8	9	10	11	12
Initial value	1	1	1	1	1	1	1	1	1	1	1	1
Step 1	0	0	0	3	3	3	2	2	0	0	2	2
Step 2	0	0	0	9	9	9	4	4	0	0	4	4
Step 3	0	0	0	27	27	27	8	8	0	0	8	8

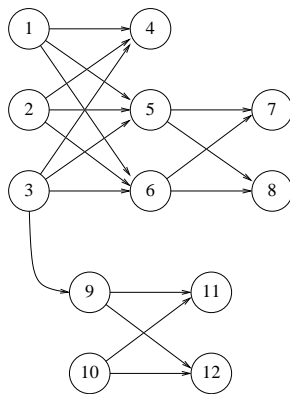
Hub scores:

Node	1	2	3	4	5	6	7	8	9	10	11	12
Initial value	1	1	1	1	1	1	1	1	1	1	1	1
Step 1	3	3	3	0	2	2	0	0	2	2	0	0
Step 2	9	9	9	0	4	4	0	0	4	4	0	0
Step 3	27	27	27	0	8	8	0	0	8	8	0	0

Hub scores:

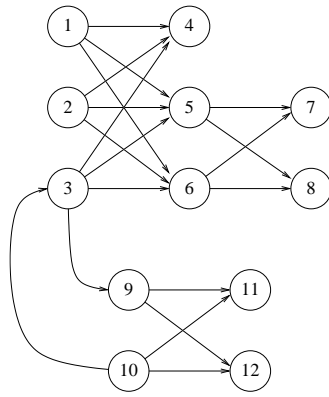
Component	G_1	G_2	G_3
Initial value	6	4	4
Step 1	18	8	8
Step 2	54	16	16
Step 3	162	32	32

22.3) After the new edge, the graph is the following:



Due to the current hub value of node 3, the authority value of node 9 increases, and in turn also the hub value of node 3 will increase, and therefore the authority values of nodes 4, 5, and 6. Therefore, the new edge causes $I(G_1)$ to increase. On the other hand, the authority value of node 9 does not impact on $I(G_3)$, where it is a source, and for the same reason the new edge has no impact on $I(G_2)$.

22.4) Finally, after the addition of the last edge:



The edge impacts on the authority score of node 3, therefore $I(G_1)$ and $I(G_2)$ do not change, and the hub score of node 10 (and hence the authorities of nodes 11 and 12) is increased. Therefore, the new edge only impacts on $I(G_3)$.

Exercise 23

A set of four web pages (A, B, C and D) is completely connected: all pages contain links to every other page, while no page contains links to itself.

23.1) Compute the PageRank score of all pages.

23.2) Now add web page E, and two links: one from C to E, the second from E to D (so that E has exactly an incoming link and an outgoing link). Compute the PageRank score of all pages.

Exercise 24

Consider a document corpus with $m = 6$ documents, $n = 5$ terms. Suppose that documents have been clustered into $m' = 2$ clusters and terms have been clustered into $n' = 2$ clusters. The following document-term matrix and cluster attribution has been determined:

		1	2	3	4	5
		1	1	2	1	2
1	1		X			
2	2			X		X
3	1	X	X		X	
4	1		X		X	
5	2			X	X	
6	2		X	X		

24.1) Consider the Jaccard index as similarity measure. Suppose that all we know about a document is that it contains term 2. Which other term is most likely to occur in the same document?

24.2) Compute the following probabilities for all suitable index values:

- the probability $p_{i'}$ that a random document belongs to cluster i' ;
- the probability $p_{j'}$ that a random item belongs to cluster j' ;
- the probability $p_{i'j'}$ that a document in cluster i' contains a term in cluster j' .

24.3) Perform a step of the Gibbs Sampling technique on document 4 by computing the posterior probabilities $\pi_{4 \rightarrow i'}$ for $i' = 1, 2$. Was the proposed cluster attribution likely, or will it be probably changed?

Exercise 25

Given the following three documents (each row is a document and each cell corresponds to a term and contains its term id)

1	1	2	1	5	2	2
2	4	3	3	1	2	1
3	2	2	5	4	3	3

assume a multinomial model for the document generation and estimate the parameters of the term distribution by using the maximum likelihood estimation method. (*Show all the steps to obtain the best parameter estimation*) As all the documents have the same length, assume $P(L = l_d|\Theta) = 1$ in the multinomial model $P(l_d, n(d, t)|\Theta)$.

Solution — The multinomial model for a document generation is the following:

$$P(d|\Theta) = P(l_d, n(d, t)|\Theta) = P(L = l_d|\Theta) \binom{l_d}{\{n(d, t)\}} \prod_{t \in d} \theta_t^{n(d, t)} \quad (1)$$

where $\Theta = (\theta_t \forall t \in T)$ and T is our vocabulary.

We have a set $D = (d_1, d_2, d_3)$ of iid observations, thus $P(D|\Theta) = \prod_{d \in D} P(d|\Theta)$

In this case, as we assume $P(L = l_d|\Theta) = 1$, then:

$$P(D|\Theta) = \left(\binom{l_d}{\{n(d, t)\}} \right) \prod_d \prod_t \theta_t^{n(d, t)} \quad (2)$$

This model corresponds to the likelihood function $L(\Theta|D)$.

We want to estimate the Θ parameters which maximize the likelihood function. We can do that by computing the partial derivatives with respect to each one of the parameters θ_t and putting them equal to zero.

From now on we will consider the log likelihood function which is easier to derive:

$$\log L(\Theta|D) = \log \left(\binom{l_d}{\{n(d, t)\}} \right) + \sum_d \sum_t n(d, t) \log(\theta_t) \quad (3)$$

moreover there is one constraint to the maximization, namely $\sum_t \theta_t = 1$, thus we perform a standard Lagrangian optimization:

$$\frac{\partial}{\partial \theta_t} \left[\log \left(\binom{l_d}{\{n(d, t)\}} \right) + \sum_d \sum_t n(d, t) \log(\theta_t) - \lambda (\sum_t \theta_t - 1) \right] = 0 \quad (4)$$

$$\frac{\partial \log L}{\partial \theta_t} = \frac{\sum_d n(d, t)}{\theta_t} - \lambda = 0 \quad (5)$$

then the estimation of our parameters is:

$$\theta_t = \frac{\sum_d n(d, t)}{\lambda} \quad (6)$$

In order to compute the Lagrangian multiplier λ , we consider the constraint $\sum_t \theta_t = 1$ which becomes $\sum_t \frac{\sum_d n(d, t)}{\lambda} = 1$ and thus $\lambda = \sum_d \sum_t n(d, t) = \sum_d l_d = n \cdot l_d$, where $n = |D|$.

The parameters are:

$$\theta_t = \frac{\sum_d n(d, t)}{n \cdot l_d} \quad (7)$$

Substituting the values we obtain $\theta_1 = \frac{4}{3 \times 6} = \frac{2}{9}$, $\theta_2 = \frac{7}{18}$, $\theta_3 = \frac{5}{18}$, $\theta_4 = \frac{1}{9}$, $\theta_5 = \frac{1}{9}$.

Exercise 26

Solve the previous exercise by using the least squares method. (*Show all the steps to obtain the best parameter estimation*)

Exercise 27

Given the following three documents (each row is a document and each cell corresponds to a term and contains its term id)

1	1	2	1	5	2	2	3	2
2	1	3	1	5	2	2		
3	2	2	5	4	3	3	2	

assume a multinomial model for the document generation and estimate the parameters of the term distribution by using the least square method. (Show all the steps to obtain the best parameter estimation)

Exercise 28

A document set has been partitioned into two clusters. For each cluster, 100 2-shingles have been sampled randomly. Shingles were divided into four categories: “2, 4” (term 2 followed by term 4), “2, $\bar{4}$ ” (term 2 followed by any term different from 4), “ $\bar{2}$, 4” (any term different from 2 followed by term 4) and “ $\bar{2}$, $\bar{4}$ ”.

The following tables show the results of our sampling in clusters 1 and 2 respectively:

C_1	4	$\bar{4}$	C_2	4	$\bar{4}$
2	20	10	2	10	10
$\bar{2}$	10	60	$\bar{2}$	0	80

28.1) Based on the above contingency tables, what are the relative frequencies of terms 2 and 4 within the two clusters? What are the relative frequencies of terms different from 2 and 4?

28.2) Consider the following term-based generative model for documents:

1. Choose the cluster by an unbiased coin throw.
2. Document length is always 6.
3. Choose every term of the document independently with probability equal to the frequency of the term within the chosen cluster.

(Hint: the model depends on four *free* parameters, i.e., probability of term 2 and probability of term 4 within each cluster).

Use this model to determine the maximum-likelihood clustering of the three following documents:

- $d_1 = 1, 2, 4, 2, 3, 5$
- $d_2 = 3, 2, 1, 3, 5, 4$
- $d_3 = 1, 2, 4, 2, 4, 2$

28.3) Use a similar generative model based on shingles instead of terms, considering every shingle as independent of the others (so that every document is made of 5 independent shingles).

Use this model to determine the maximum-likelihood clustering of the same documents.

Nota bene: this is not an exercise about parameter estimation. Parameters are already given, only document attribution to clusters must be decided.

Solution — **28.1)** We just count the frequency of shingles containing term 2 in the two clusters and divide by the total number of samples. In cluster 1, for example, 30 samples out of 100 contain term 2. We do similarly for term 4, then compute the remaining probability:

	Term 2	Term 4	Any other term
C_1	.3	.3	.4
C_2	.2	.1	.7

28.2) We must compute the probability for each document to be generated within each cluster. Since every term is generated independently, the probability of the document is just the probability of each term being selected in its position. Let us define as p_{ij} the probability for document i to be generated in cluster j . For example:

$$p_{21} = .4 \times .3 \times .4 \times .4 \times .4 \times .3 = .002304$$

Probabilities are:

	C_1	C_2
d_1	.001728	.001372
d_2	.002304	.004802
d_3	.000972	.000056

Therefore, documents d_1 and d_3 are attributed to cluster C_1 , document d_2 to cluster C_2 .

28.3) Same computation with shingles:

	C_1	C_2
d_1	.00432	.00512
d_2	.00216	0
d_3	.00864	.00512

Notice that the probability that d_2 is generated in cluster C_2 is null because it contains the shingle “ $\bar{2}, 4$ ”. Therefore, documents d_2 and d_3 are attributed to cluster C_1 , while d_1 is attributed to cluster C_2 .

Exercise 29

As a part of a clustering method, we decide to compute for each document d the number of occurrences of the most frequent term in that document:

$$N(d) = \max_{t \in T} n(t, d).$$

We want to model $N(d)$ as a Poisson random variable with parameter λ , so that given a random document d in our corpus we have

$$\Pr(N(d) = n; \lambda) = \frac{\lambda^n e^{-\lambda}}{n!}$$

29.1) Given a sampled document set d_1, \dots, d_k , show that the maximum likelihood estimate of λ in the Poisson model is the average of $N(d_i)$.

29.2) Determine λ based on the following sample:

- $d_1 = (1, 2, 3, 4, 3, 2, 3, 3, 2, 1, 5, 6, 3)$
- $d_2 = (3, 2, 4, 4, 2, 3, 2, 4, 5, 1, 6)$
- $d_3 = (6, 4, 3, 5, 2, 6, 1, 7)$
- $d_4 = (6, 5, 4, 3, 2, 1, 2, 6, 2, 2)$
- $d_5 = (4, 3, 4, 2, 5, 4, 1, 6, 3)$

Solution —

29.1) The likelihood of λ with respect to the sample set is

$$L(\lambda; N(d_1), \dots, N(d_k)) = \Pr(N(d_1), \dots, N(d_k); \lambda) = \prod_{i=1}^k \Pr(N(d_i); \lambda) = \prod_{i=1}^k \frac{\lambda^{N(d_i)} e^{-\lambda}}{N(d_i)!}.$$

Therefore, the log-likelihood is

$$\log L(\lambda; N(d_1), \dots, N(d_k)) = \sum_{i=1}^k (N(d_i) \log \lambda - \lambda - \log N(d_i)!),$$

and its derivative with respect to λ is

$$\frac{d}{d\lambda} \log L(\lambda; N(d_1), \dots, N(d_k)) = \sum_{i=1}^k \left(\frac{N(d_i)}{\lambda} - 1 \right) = \frac{1}{\lambda} \sum_{i=1}^k N(d_i) - k.$$

Equating the derivative to zero, we get

$$\lambda = \frac{1}{k} \sum_{i=1}^k N(d_i).$$

29.2) By counting:

$$N(d_1) = 5, \quad N(d_2) = 3, \quad N(d_3) = 2, \quad N(d_4) = 4, \quad N(d_5) = 3,$$

the maximum likelihood estimate is the average of these values over the 5-document sample:

$$\hat{\lambda} = \frac{5 + 3 + 2 + 4 + 3}{5} = \frac{17}{5}.$$