Exercises and Supervision Instruction for Information Retrieval

Simone Teufel

Lent Term 2014/15

Contents

General Instructions

Thank you for supervising! You are free to assemble your own supervisions from the material given below. You will have to select a number of exercises for your students. Some of these are exam questions, some exercises, some are little experiments the students can do with a search engine to demonstrate a point to them.

Please point out to the students the following:

- The course is closely modelled to the textbook; it is absolutely essential that they study with the book
- The course has been remodelled in 2013/14; it was previously much less closely related to the textbook.
- That is the reason why I am also listing here the exercises from the book, but please note that some exercises go beyond the material taught in the lectures. These are marked with a star.
- In particular, the following topics have not been treated on this course before 2013/14: Text Classification, Tolerant retrieval.
- Previously, there was less emphasis on practicalities such as exact data structures used.
- These changes mean that while some previous exam questions *can* still be used for preparation, if they fit into the changed course, it is certainly not sufficient to rely only on these.

1 Boolean Model [Lecture 1]

1.1 Exercises from book

- **Exercise 1.2** Draw the term-document incidence matrix and the inverted index representation for the following document collection:
  Doc 1 breakthrough drug for schizophrenia
  Doc 2 new schizophrenia drug
  Doc 3 new approach for treatment of schizophrenia
  Doc 4 new hopes for schizophrenia patients

- **Exercise 1.3** For the document collection shown in Exercise 1.2, what are the returned results for these queries:
  - schizophrenia AND drug
  - for AND NOT(drug OR approach)

- **Exercise 1.4 *** For the queries below, can we still run through the intersection in time O(x + y), where x and y are the lengths of the postings lists for Brutus and Caesar? If not, what can we achieve?
  - Brutus AND NOT Caesar
  - Brutus OR NOT Caesar

- First make sure they understand the merge algorithm for AND, then, if they seem particularly intersted, you can make them do:

- **Exercise 1.5 *** Extend the postings merge algorithm to arbitrary Boolean query formulas. What is its time complexity? For instance, consider:

  (Brutus OR Caesar) AND NOT (Antony OR Cleopatra)

  Can we always merge in linear time? Linear in what? Can we do better than this?

- **Exercise 1.7** Recommend a query processing order for

  (tangerine OR trees) AND (marmalade OR skies) AND (kaleidoscope OR eyes)

  given the following postings list sizes:

Term	Postings size
eyes	213312
kaleidoscope	87009
marmalade	107913
skies	271658
tangerine	46653
trees	316812

- **Exercise 1.9** For a conjunctive query, is processing postings lists in order of size guaranteed to be optimal? Explain why it is, or give an example where it isn't.

- **Exercise 1.11 *** How should the Boolean query x AND NOT y be handled? Why is naive evaluation of this query normally very expensive? Write out a postings merge algorithm that evaluates this query efficiently.

- **Exercise 1.12** Write a query using Westlaw syntax which would find any of the words professor, teacher, or lecturer in the same sentence as a form of the verb explain.

1.2 Other Exercises/Discussion Points

1. Why don't we use grep for information retrieval?
2. Why don't we use a relational database for information retrieval?
3. In constructing the index, which step is most expensive/complex?
4. Name Westlaw operations that go beyond strictly Boolean operators.
5. Googlewhack is a game started in 2002. The task is to find a pair of search terms that return exactly *one* document in a Google search.

    - *anxiousness scheduler*
    - *squirreling dervishes*

    Deceptively hard! Spend some time trying to find a new googlewhack – it will give you an idea what kinds of words might qualify, and why this is so hard.

2 Indexing and document normalisation [Lecture 2]

2.1 Exercises from the Book

- **Exercise 2.1** Are the following statements true or false?
  - In a Boolean retrieval system, stemming never lowers precision.
  - In a Boolean retrieval system, stemming never lowers recall.
  - Stemming increases the size of the vocabulary.
  - Stemming should be invoked at indexing time but not while processing a query.
- **Exercise 2.4** For the top Porter stemmer rule group shown on slide 69:
  - What is the purpose of including an identity rule such as SS →SS?
  - Applying just this rule group, what will the following words be stemmed to?
    circus canaries boss
  - What rule should be added to correctly stem *pony*?
  - The stemming for *ponies* and *pony* might seem strange. Does it have a deleterious effect on retrieval? Why or why not?
- **Exercise 2.5** Why are skip pointers not useful for queries of the form x OR y?
- **Exercise 2.6 *** We have a two-word query. For one term the postings list consists of the following 16 entries:

  [4,6,10,12,14,16,18,20,22,32,47,81,120,122,157,180] and for the other it is the one entry postings list: [47]. Work out how many comparisons would be done to intersect the two postings lists with the following two strategies. Briefly justify your answers:
  - Using standard postings lists
  - Using postings lists stored with skip pointers, with a skip length of P, as suggested in the lecture.
- **Exercise 2.8** Assume a biword index. Give an example of a document which will be returned for a query of *New York University* but is actually a false positive which should not be returned.
- **Exercise 2.9** Shown below is a portion of a positional index in the format:

  term: doc1: <position1, position2, . . . >; doc2: <position1, position2, . . . >; etc.

  angels: 2: <36,174,252,651>; 4: <12,22,102,432>; 7: <17>;
  fools: 2: <1,17,74,222>; 4: <8,78,108,458>; 7: <3,13,23,193>;
  fear: 2: <87,704,722,901>; 4: <13,43,113,433>; 7: <18,328,528>;
  in: 2: <3,37,76,444,851>; 4: <10,20,110,470,500>; 7: <5,15,25,195>;
  rush: 2: <2,66,194,321,702>; 4: <9,69,149,429,569>; 7: <4,14,404>;
  to: 2: <47,86,234,999>; 4: <14,24,774,944>; 7: <199,319,599,709>;
  tread: 2: <57,94,333>; 4: <15,35,155>; 7: <20,320>;
  where: 2: <67,124,393,1001>; 4: <11,41,101,421,431>; 7: <16,36,736>;

  Which document(s) if any match each of the following queries, where each expression within quotes is a phrase query?
  - fools rush in

  - fools rush in AND angels fear to tread

- **Exercise 2.12 *** Consider the adaptation of the basic algorithm for intersection of two postings lists (Figure 1.6, page 11) to the one in Figure 2.12 (page 42), which handles proximity queries. A naive algorithm for this operation could be O($PL_{max}{}^2$), where P is the sum of the lengths of the postings lists (i.e., the sum of document frequencies) and L_{max} is the maximum length of a document (in tokens).
  - Go through this algorithm carefully and explain how it works.
  - What is the complexity of this algorithm? Justify your answer carefully.

2.2 Other Exercises/Discussion Points

- Define the number of types and tokens in a sentence. How many tokens does the following verse contain?

  Come as you are
  as you were
  as I want you to be
  as a friend
  as a friend
  as an old enemy

- Download the Porter stemmer, e.g. from `http://tartarus.org/martin/PorterStemmer/` and play around with it on a text of your choice.

- Discuss the limitations of the equivalence classing approach with an example. Solution: suit/suits is a good example. Cf. example in book. Any proper names that happens to be a common noun should not be matched (equivalence classed) to common name class if uppercased. Unsymmetrical expansion.

- What is a stop list?

- Porter stemmer questions:
  1. Show which stems *rationalisations, rational, rationalizing* result in, and which rules they use.
  2. Explain why *sander* and *sand* do not get conflated.
  3. What would you have to change if you wanted to conflate them?
  4. Find five different examples of incorrect stemmings.
  5. Can you find a word that gets reduced in every single step (of the 5)?

- Show cases where the porter stemmer is too aggressive.

- Try to find counterexamples for rules of your choice in the porter stemmer.

- Build (or plan how you would build) your own Boolean index and search engine – as an added thrill, using only unix tools on the command line (if you want)...

- Name two data structures that support phrase queries, and explain how they do it.

- Name a data structure that supports proximity queries.

2.3 Exam questions

- 2004 P7Q12a-c

3 Tolerant Retrieval [Lecture 3]

3.1 Exercises from the book

- **Exercise 3.1** In the permuterm index, each permuterm vocabulary term points to the original vocabulary term(s) from which it was derived. How many original vocabulary terms can there be in the postings list of a permuterm vocabulary term?
- **Exercise 3.3** If you wanted to search for s*ng in a permuterm wildcard index, what key(s) would one do the lookup on?
- **Exercise 3.4** Refer to Figure 3.4 in the textbook; it is pointed out in the caption that the vocabulary terms in the postings are lexicographically ordered. Why is this ordering useful?
- **Exercise 3.5** Consider again the query fi*mo*er from Section 3.2.1. What Boolean query on a bigram index would be generated for this query? Can you think of a term that matches the permuterm query in Section 3.2.1, but does not satisfy this Boolean query?
- **Exercise 3.6** Give an example of a sentence that falsely matches the wildcard query `mon*h` if the search were to simply use a conjunction of bigrams.
- **Exercise 3.7** If $|s_i|$ denotes the length of string s_i, show that the edit distance between s_1 and s_2 is never more than $\max(|s_1|, |s_2|)$.

3.2 Other Exercises/Discussion Points

1. Which data structures are typically used for locating the entry for a term in the dictionary, and why?
2. Which of these is best if the collection is static?
3. What sequence of letters is looked up in the permuterm index for the following wildcard queries? X, X*, *X, *X* X*Y
4. What is the difference between the regular inverted index used in IR and the k-gram index?
5. Give formulae for Zipf's law and Heap's law.
6. Draw a trie which encodes the following terms: *Hawai'i, hare, hiss, hissing, hissed, he, hunger, honey, hello, hallo, Hungary.*
7. Now draw a hash in which the same terms are stored. Consider a word $word = l_1 l_2 \dots l_n$. Let code(l_i) be the number of letter l_i in the alphabet (e.g., code(a)=1). Use the hash function of h(word) = [code(l_1) + code(l_2)] mod 26
8. Caculate the edit distance between *cat* – *catcat.*

   Solution: Edit distance is 3.

			c		a		t		c		a		t	
		0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**	**5**	**5**	**6**	**6**
c		**1**	**0**	2	2	3	3	4	**3**	5	5	6	6	7
		1	2	**0**	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**	**5**	**5**
a		**2**	2	**1**	**0**	2	2	3	3	4	**3**	5	5	6
		2	3	**1**	2	**0**	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
t		**3**	3	**2**	2	**1**	**0**	2	2	3	3	4	**3**	5
		3	4	**2**	3	**1**	2	**0**	**1**	**1**	**2**	**2**	**3**	**3**

9. How many transformations exist to turn “cat” into “catcat”? How can these be read off the edit distance matrix?

Solution: 4. To be read off as follows:

		c	a	t	c	a	t
	0	**1** **1**	**2** **2**	**3** **3**	**4** **4**	**5** **5**	**6** **6**
c	**1** **1**	**0** 2 2 **0**	2 3 **1** **1**	3 4 **2** **2**	**3** 5 **3** **3**	5 6 **4** **4**	6 7 **5** **5**
a	**2** **2**	2 **1** 3 **1**	**0** 2 2 **0**	2 3 **1** **1**	3 4 **2** **2**	**3** 5 **3** **3**	5 6 **4** **4**
t	**3** **3**	3 **2** 4 **2**	2 **1** 3 **1**	**0** 2 2 **0**	2 3 **1** **1**	3 4 **2** **2**	**3** 5 **3** **3**

cost	operation	input	output
1	insert	*	c
1	insert	*	a
1	insert	*	t
0	(copy)	c	c
0	(copy)	a	a
0	(copy)	t	t

		c	a	t	c	a	t
	0	**1** **1**	**2** **2**	**3** **3**	**4** **4**	**5** **5**	**6** **6**
c	**1** **1**	**0** 2 2 **0**	2 3 **1** **1**	3 4 **2** **2**	**3** 5 **3** **3**	5 6 **4** **4**	6 7 **5** **5**
a	**2** **2**	2 **1** 3 **1**	**0** 2 2 **0**	2 3 **1** **1**	3 4 **2** **2**	**3** 5 **3** **3**	5 6 **4** **4**
t	**3** **3**	3 **2** 4 **2**	2 **1** 3 **1**	**0** 2 2 **0**	2 3 **1** **1**	3 4 **2** **2**	**3** 5 **3** **3**

cost	operation	input	output
0	(copy)	c	c
1	insert	*	a
1	insert	*	t
1	insert	*	c
0	(copy)	a	a
0	(copy)	t	t

		c	a	t	c	a	t
	0	**1** **1**	**2** **2**	**3** **3**	**4** **4**	**5** **5**	**6** **6**
c	**1** **1**	**0** 2 2 **0**	2 3 **1** **1**	3 4 **2** **2**	**3** 5 **3** **3**	5 6 **4** **4**	6 7 **5** **5**
a	**2** **2**	2 **1** 3 **1**	**0** 2 2 **0**	2 3 **1** **1**	3 4 **2** **2**	**3** 5 **3** **3**	5 6 **4** **4**
t	**3** **3**	3 **2** 4 **2**	2 **1** 3 **1**	**0** 2 2 **0**	2 3 **1** **1**	3 4 **2** **2**	**3** 5 **3** **3**

cost	operation	input	output
0	(copy)	c	c
0	(copy)	a	a
1	insert	*	t
1	insert	*	c
1	insert	*	a
0	(copy)	t	t

		c	a	t	c	a	t
	 0	 **1** **1**	 **2** **2**	 **3** **3**	 **4** **4**	 **5** **5**	 **6** **6**
c	**1** **1**	**0** 2 2 **0**	2 3 **1** **1**	3 4 **2** **2**	**3** 5 **3** **3**	5 6 **4** **4**	6 7 **5** **5**
a	**2** **2**	2 **1** 3 **1**	**0** 2 2 **0**	2 3 **1** **1**	3 4 **2** **2**	**3** 5 **3** **3**	5 6 **4** **4**
t	**3** **3**	3 **2** 4 **2**	2 **1** 3 **1**	**0** 2 2 **0**	2 3 **1** **1**	3 4 **2** **2**	**3** 5 **3** **3**

cost	operation	input	output
0	(copy)	c	c
0	(copy)	a	a
0	(copy)	t	t
1	insert	*	c
1	insert	*	a
1	insert	*	t

10. Why is the assymptotic complexity of edit distance calculation quadractic?

11. Show that a naive implementation of the recursive definition of edit distance is assymptotically worse (how bad?)

12. Give an algorithm to read out one optimal transformation, and state its assymptotic complexity.

13. Give an algorithm to read out all optimal transformations, and state its assumptotic complexity.

3.3 Exam questions

- 2006 P7Q11 (note that this question was asked when tolerant retrieval was NOT specifically treated in the lectures).

4 Term Weighting and VSM [Lecture 4]

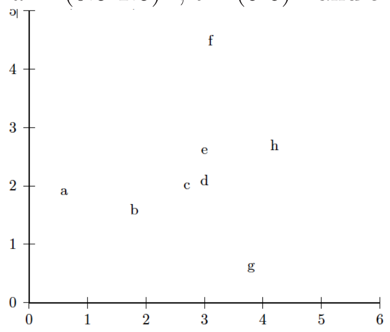
4.1 Exercises from the Book

- **Exercise 6.8** Why is the idf of a term always finite?
- **Exercise 6.9** What is the idf of a term that occurs in every document? Compare this with the use of stop word lists.
- **Exercise 6.10** Consider the table of term frequencies for 3 documents denoted Doc1, Doc2, Doc3 in Figure 6.9. Compute the tf-idf weights for the terms car, auto, insurance, best, for each document, using the idf values from Figure 6.8.
- **Exercise 6.15** Recall the tf-idf weights computed in Exercise 6.10. Compute the Euclidean normalized document vectors for each of the documents, where each vector has four components, one for each of the four terms.
- **Exercise 6.16** Verify that the sum of the squares of the components of each of the document vectors in Exercise 6.15 is 1 (to within rounding error). Why is this the case?
- **Exercise 6.17** With term weights as computed in Exercise 6.15, rank the three documents by computed score for the query car insurance, for each of the following cases of term weighting in the query:
  - The weight of a term is 1 if present in the query, 0 otherwise.
  - Euclidean normalized idf.
- **Exercise 6.19** Compute the vector space similarity between the query digital cameras and the document digital cameras and video cameras by filling out the empty columns in Table 6.1. Assume N = 10,000,000, logarithmic term weighting (wf columns) for query and document, idf weighting for the query only and cosine normalization for the document only. Treat and as a stop word. Enter term counts in the tf columns. What is the final similarity score?
- **Exercise 6.20** Show that for the query “affection”, the relative ordering of the scores of the three documents in Figure 6.13 is the reverse of the ordering of the scores for the query “jealous gossip”.
- **Exercise 6.23** Refer to the tf and idf values for four terms and three documents in Exercise 6.10. Compute the two top scoring documents on the query best car insurance for each of the following weighing schemes: (i) nnn.atc; (ii) ntc.atc.

4.2 Other Exercises/Discussion Points

1. What is the bag-of-words model?
2. What is the advantage of idf weighting compared to inverse-collection-frequency weighting?
3. What is the relationship between term frequency and collection frequency?
4. Compute the Jaccard matching score and the tf matching score for the following query-document pairs.
   - q: [information on cars] d: “all you’ve ever wanted to know about cars”
   - q: [information on cars] d: “information on trucks, information on planes, information on trains”
   - q: [red cars and red trucks] d: “cops stop red cars more often”

5. In the figure below, which of the three vectors $\vec{a}, \vec{b}$ and $\vec{c}$ is most similar to $\vec{x}$ according to (i) dot-product similarity ($\sigma_i x_i \cdot y_i$), (iii) cosine similarity ($\frac{\sigma_i x_i \cdot y_i}{|\vec{x}||\vec{y}|}$), (iii) Euclidean distance ($|x - y|$)? The vectors are $\vec{a} = (0.5\ 1.5)^T$, $\vec{b}$ =$(6\ 6)^T$ and $\vec{c} = (12\ 9)^T$, and $\vec{x} = (2\ 2)^T$.



6. Consider the following situation in a collection of N=100,000,000 documents. The query is "John Miller" and the document "John Miller, John Fisher, and one other John". Treat "and" "one" and "other" as stop words. df_{John} = 50,000; df_{Fisher} = 100,000; df_{Miller} = 10,000.

   - What is the Jaccard similarity between query and document?
   - What is the lnc.ltn similarity between query and document?

7. Compute the similarity between query "smart phones" and document "smart phones and video phones at smart prices" under lnc.ltn similarity. Assume N=10,000,000. Treat "and" and "at" as stop words. df_{smart} = 5,000; df_{video} = 50,000; df_{phones} = 25,000; df_{prices} = 30,000;

   When computing length-normalised weights, you can round the length of a vector to the nearest integer.

   Here are some log values you may need: (assumption: use in exam without calculator allowed)

x	1	2	3	4	5	6	7	8	9
$log_{10}(x)$	0	0.3	0.5	0.6	0.7	0.8	0.8	0.9	1.0

8. Ask the students to choose three very short documents (of their own choice, or they can make them up!) and then

   - Build a binary document-term matrix
   - Build an TFIDF weighted query (you may estimate the IDF)
   - Write a suitable query
   - Calculate document–query similarity, using
     - cosine
     - inner product (i.e. cosine without normalisation)

     (Of course, having to mark arbitrary documents and queries is creating more work for you as a supervisor).
   - What effect does normalisation have?
   - What effect does stemming have?
   - What effect does equivalence classing by prefix have (e.g., same first 4 letters)?

9. If we were to have only one-term queries, explain why the use of weight-ordered postings lists (i.e., postings lists sorted according to weight, instead of docID) truncated at position k in the list suffices for identifying the k highest scoring documents. Assume that the weight w stored for a document d in the postings list of t is the cosine-normalised weight of t for d.

10. Play around with your own implementation:

    - Modify your implementation from lecture 2 so that your search model is now a VSP – several parameters can be varied

4.3 Exam questions

- 2003 P7Q11a
- 2010 P8Q9a
- 2013 P8Q9ab

5 IR evaluation [Lecture 5]

5.1 Exercises from book

- **Exercise 8.3**

  Derive the equivalence between the two formulae given for the F-measure in the lectures (F_α vs F_β)

- **Exercise 8.4**

  What are the possible values for interpolated precision at a recall level of 0?

- **Exercise 8.5**

  Must there always be a break-even point between Precision and Recall? Either show there must or provide a counterexample.

  The type of graph where we can observe P/R break-even point has not been introduced in the lectures, but it's an obvious extension of the one where the Y-axis is P and the X-axis is R. Now, we plot both P and R on the Y-axis, as two independent functions. The X-axis is now the rank of positions in the ranked return list. In a typical curve, P tends to start high but then decrease, whereas R starts low and then increases. At some point in a typical curve, the two functions will cross, but is a situation possible where this does not happen?

- **Exercise 8.8** (avg 11 point prec instead of R-precision)

  Consider an information need for which there are 4 relevant documents in the collection. Contrast two systems run on this collection. Their top ten results are judged for relevance as follows (with the leftmost being the top-ranked search result):

System 1	R N R N N N N N R R
System 2	N R N N R R R N N N

  a) What is the MAP of each system?
  b) Does this result intuitively make sense? What does it say about what is important to get a high MAP score?
  c) What is the avg- 11 point precision of the systems? Observations?

- **Exercise 8.9**

  The following list of Rs and Ns represents relevant (R) and non-relevant (N) returned documents (as above) in a ranked list of 20 documents retrieved in response to a query from a collection of 10,000 documents. The top of the ranked list is on the left of the list. The list shows 6 relevant documents. Assume that there are 8 relevant documents in the collection.

  R R N N N   N N N R N   R N N N R   N N N N R

  a) What is the precision of the system in the top twenty?
  b) What is the F_1 on the top twenty?
  c) What is the (uninterpolated) precision of the system at 25% recall? d) What is the interpolated precision at 33% recall?
  e) Assume that these twenty documents are the complete result set of the system. What is the MAP for the query?
  f) What is the largest possible MAP that this system could have?
  g) WHat is the smallest possible MAP that this system could have?
  h) In a set of experiments, only the top 20 results are evaluated by hand. The result in (e) is used to approximate the range (f) to (g). For this example, how large in absolute terms can the error for MAP be by calculating (e) instead of (f) and (g) for this query?

- **Exercise 8.10bc** Below is a table showing how two human judges rated the relevance of a set of 12 documents to a particular information need (0 = nonrelevant, 1 = relevant). Let us assume that you have written and IR system for this query that returns the set of documents {4,5,6,7,8}.

docID	Judge 1	Judge 2
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

5.2 Other Exercises/Discussion Points

1. Name three criteria for evaluating a search engine.
2. What are the components of an information retrieval benchmark?
3. Explain the difference between the concepts of a query and an information need.
4. What is an easy way of maximising the recall of an IR engine?
5. What is an easy way of maximising the precision of an IR engine?
6. Precision and recall – make sure they understand the difference to accuracy. This is somehow really hard to understand for some of them.
7. Ask them to give you precision out of the first 10, 20, 30 documents on their own Google query to the TREC information need "hair loss".
8. Discuss; how did they express their query (synonymy etc).
9. What is the so-called 'recall problem' in IR?
10. If you feel that asking them to go through the worked examples for MAP and avg-11-pt precision is somehow still not enough, you can give them relevance tables for several made-up queries (ideally more than 2) similar to those on slide 2 3, play through MAP and avg-11-point prec with interpolation. What is important is that they can describe to you why averaging over queries with a different number of relevant documents makes it impossible to use the simple one-number metrics (such as P at rank X)
11. An evaluation benchmark ideally should tell us for any document–query pair whether the document is relevant to the query.
    - Why is Cranfield the only collection that actually satisfies this desideratum?
    - Explain the difference between a ROC and a precision–recall curve. Can you transform one into the other?
    - How do modern benchmarks solve this problem, i.e., provide document–query judgements?

5.3 Exam questions

- 2004 P7Q12d
- 2007 P7Q5ab
- 2011 P8Q9

6 Clustering [Lecture 6]

6.1 Exercises from Textbook

- **Exercise 16.1** Define two documents as similar if they have at least two proper names such as Clinton and Sarkozy in common. Give an example of an information need and two documents, for which the cluster hypothesis does *not* hold for this notion of similarity.

- **Exercise 16.4** Why do documents that do not use the same term for the concept *car* (but that are about cars) still tend to end up in the same cluster in K-means clustering?

- **Exercise 16.5** Two of the possible termination conditions for K-means were (i) assignment does not change, (ii) centroids do not change (page 332). Do these two conditions imply each other?

- **Exercise 16.13*** Prove that $RSS_{min}(K)$ is monotonically decreasing in K. (In other words, the more K centers we have, the higher our final RSS will be, in comparison to cases where K is lower.)

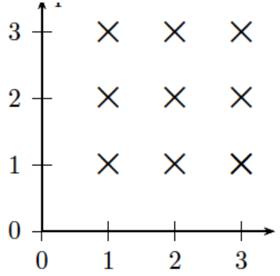
- **Exercise 16.17** Perform a K-means clustering for the documents in the table below. After how many iterations does K-means converge?

docID	document text
1	hot chocolate cocoa beans
2	cocoa ghana africa
3	beans harvest ghana
4	cocoa butter
5	butter truffles
6	sweet chocolate
7	sweet sugar
8	sugar cane brazil
9	sweet sugar beet
10	sweet cake icing
11	cake black forest

- **Exercise 17.12** How many different clusterings of N points into K flat clusters are there? (Note to supervisors – Answer on page 366; $\geq N^k$ different flat clusterings). What is the number of different hierarchical clusterings (dendrograms) of N documents? Are there more flat or more hierarchical clusterings for any given K and N?

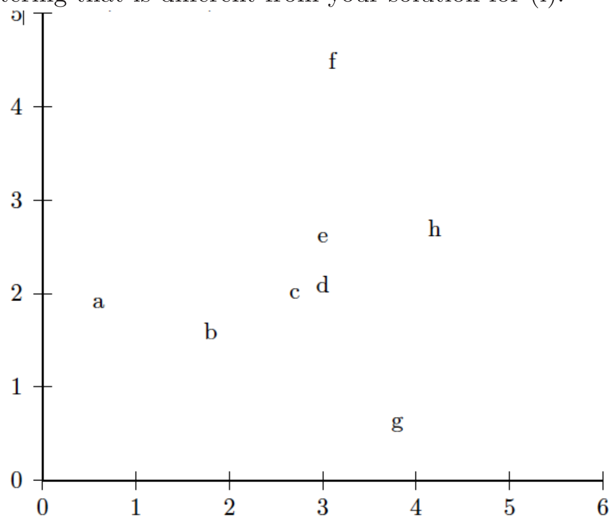
6.2 Other Exercises/Discussion Points

1. Perform a 3-means clustering of the points below:



(i) Draw a different diagram for each iteration to show the assignments and the centroids. If a tie occurs during an assignment step, you can freely choose any of the possible assignments.

(ii) There are several clusterings that 3-means can converge to in this case. Give an example of one such clustering that is different from your solution for (i).



2. Compute single link and complete-link clusterings of the set of points shown below, and depict them as dendrograms. Make sure to indicate the merge value of each horizontal "merge" line (i.e., the similarity of the two clusters that are being merged in this step). Define the similarity of two points as $-(x_1 - x_2)^2 - (y_1 - y_2)^2$. The coordinates of the points are:

Point	X	Y
a	0.6	1.9
b	1.8	1.6
c	2.7	2.0
d	3.0	2.1
e	3.0	2.6
f	3.1	4.5
g	3.8	0.6
g	4.2	2.7

3. Supervisors, you can invent another example of 2-dimensional clustering in Euclidean space, or possibly even one with documents, and see how they cluster under single and complete link. For instance, use the documents in the table from Exercise 16.17 above.

4. Give the mathematical definition of the centroid.

5. Why is result set clustering useful?

6. What does it mean that K-means is suboptimal? Give an example where K-means converges to an unintuitive clustering.

7. Play with Cludo:

   - Download and install the open-source clustering toolkit Cludo
   - Download rat gene data from IR course website
   - See if you can replicate the clusterings from the lecture (they were not created using Cludo, but with another toolkit)

6.3 Exam questions

- 2012 P8Q9cd

7 Text Classification [Lecture 7]

[Please note that this lecture has been "cut down" in size from the original plan and slides due to overrun; I will discuss neither the derivation of the NB formula nor text classification evaluation. This should be taken into account when choosing exercises for the supervision.]

7.1 Exercises from book

- **Exercise 13.1** Why is $\mathbb{C}||V| < |\mathbb{D}|L_{ave}$ expected to hold for most test collections?
- **Exercise 13.3** The rationale for the positional independence assumption is that there is no useful information in the fact that a term occurs in position k of a document. Find exceptions. Consider formulaic documents with a fixed document structure.

7.2 Other Exercises/Discussion Points

1. Based on the data below, estimate a multinomial Naive Bayes classifier and apply the classifier to the test document.

	docID	words in document	in c = China?
training set	1	Kyoto Osaka Taiwan	yes
	2	Japan Kyoto	yes
	3	Taipei Taiwan	no
	4	Macao Shanghai Taiwan	no
	5	London	no
test set	6	Taiwan Taiwan Kyoto	?

   You only need to provide the subset of the parameters that you need to classify the test set (e.g., it's not necessary to estimate the lexical probabilities for "London").

2. What is bad about maximum likelihood estimates of the parameters $P(t|c)$ in Naive Bayes?
3. What is the time complexity of a Naive Bayes classifier, and why?
4. What is the main independence assumption of Naive Bayes (insist on them giving you a formula, not a natural language description).
5. What is the difference between clustering and classification? How can they be used in a complete IR system?

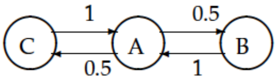
8 Link Analysis [Lecture 8]

8.1 Exercises from book

- **Exercise 21.1** Is it always possible to follow directed edges (hyperlinks) in the web graph from any node (web page) to any other? Why or why not?

- **Exercise 21.2** Find an instance of misleading anchor text on the web.

- **Exercise 21.5** What is the transition probability matrix for the following example:

  How strange of the authors to choose this as an exercise, as the solution to this exercise is actually in the book, right there on p. 425. So, maybe not very sensible as is. You could go with Ex. 21.6 instead, or modify this question by saying that you want to see the transition prob. matrix with a teleportation rate of $\alpha = 0.1$.

- **Exercise 21.6** Consider a web graph with three nodes 1, 2 and 3. The links are as follows: $1 \leftarrow 2$, $3 \leftarrow 2$, $2 \leftarrow 1$, $2 \leftarrow 3$. Write down the transition prob. matrix for the surfer's walk with teleporting, for the following 3 values of teleportation: (i) $\alpha = 0$, (ii) $\alpha = 0.5$ (iii) $\alpha = 1$.

- **Exercise 21.7*** A user of a browser can, in addition to clicking a hyperlink on the page x she is currently browsing, use the back button to go back to the page from which she arrived at x. Can such a use of back buttons be modelled as a Markov chain? How would we model repeated invocations of the back button?

- **Exercise 21.8** Consider a Markov chain with three states, A, B and C, and transition probabilities as follows. From state A, the next state is B with probability 1. From B, the next state is either A with probability p_A, or state C with probability $1 - p_A$. From C, the next state is A with probability 1. For what values of $p_A \in [0, 1]$ is this Markov chain ergodic?

- **Exercise 21.11** Verify that the pagerank of the data in the following transition matrix (from book and lectures)

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0.02	0.02	0.88	0.02	0.02	0.02	0.02
d_1	0.02	0.45	0.45	0.02	0.02	0.02	0.02
d_2	0.31	0.02	0.31	0.31	0.02	0.02	0.02
d_3	0.02	0.02	0.02	0.45	0.45	0.02	0.02
d_4	0.02	0.02	0.02	0.02	0.02	0.02	0.88
d_5	0.02	0.02	0.02	0.02	0.02	0.45	0.45
d_6	0.02	0.02	0.02	0.31	0.31	0.02	0.31

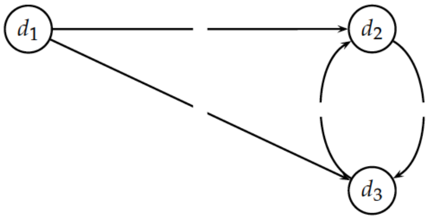
  is indeed

  $\vec{x} = (0.05\,0.04\,0.11\,0.25\,0.21\,0.04\,0.31)$

  Write a small routine or use a scientific calculator to do so. [Please additionally check that they know how to produce a transition with teleportation matrix in general.]

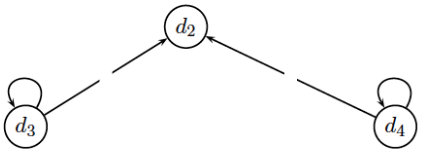
- **Exercise 21.19** If all the hub and authority scores are initialised to 1, what is the hub/authority score of a node after one iteration?

- **Exercise 21.22** For the web graph in the following figure, compute PageRank, hub and authority scores for each of the three pages.

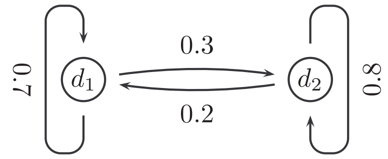


8.2 Other Exercises/Discussion Points

1. Compute PageRank for the web graph below for each of the three pages. Teleportation probability is 0.6.



2. Compute PageRank of the following network using the power method



Solution:

	x_1	x_2		
	$P_t(d_1)$	$P_t(d_2)$		
			$P_{11} = 0.7$	$P_{12} = 0.3$
			$P_{21} = 0.2$	$P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2	0.3	0.7	0.35	0.65
t_3	0.35	0.65	0.375	0.625
				...
t_∞	0.4	0.6	0.4	0.6

PageRank vector $= \vec{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$
$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

3. Make them invent and calculate their own network example for HITS.

4. When using PageRank for ranking, what assumptions are we making about the meaning of hyperlinks?

5. Why is PageRank a better measure of quality than a simple count of inlinks?

6. What is the meaning of the PageRank q of a page d in the random surfer model?

7. Make sure they understand the two main differences between PageRank and HITS (namely 1. offline vs. online, 2. different underlying model of importance/of the inherent properties of a web page). You can use MatLab or make them program it in perl.
8. Discuss with them exactly how anchor text is used for queries.

8.3 Exam questions

- 2009 P8Q9
- 2010 P8Q9b