

---

# IR&TM Review I

## Part 1: Questions

### Chapter 1

#### Question

Why don't we use grep for information retrieval?

**Answer:** grep doesn't support ranking, line-oriented, slow

#### Question

Why don't we use a relational database for information retrieval?

**Answer:** Relational databases don't support ranking.

#### Question

Google does not always interpret the query as a boolean conjunction of its terms. Give examples.

**Answer:** anchor text, variants of terms (morphology, synonyms), long queries

#### Question

What is a term-document incidence matrix?

**Answer:** Entry  $(i,j)$  is 1 if term  $i$  occurs in document  $j$ , 0 otherwise

#### Question

In constructing the index, which step is most expensive/complex?

**Answer:** sorting the postings

#### Question

Complex Boolean retrieval systems like Westlaw use many operations that go beyond strictly Boolean operators. Name some of them.

**Answer:** within  $n$  words/sentence/paragraph proximity, truncation, phrases

### Chapter 2

#### Question

Define the number of types/tokens in a sentence.

**Answer:** Number of tokens = length of sentence in words. Number of types = number of distinct words.

---

**Question**

An IR system can normalize terms by defining equivalence classes. E.g., “suit” and “suits” could be in an equivalence class. What is the limitation of this model in IR?

**Answer:** A query term t1 may be a good match for document term t2, even though query term t2 is not a good match for document term t1. Example t1 = windows, t2 = Windows

**Question**

What is tokenization? **Answer:** splitting a sentence (or text) into tokens

**Question**

Give an example in English where tokenization is nontrivial **Answer:** page 21 of lecture 2, e.g., Boston-Chicago flight vs Hewlett-Packard

**Question**

Give an example in German where tokenization is nontrivial **Answer:** compounds

**Question**

What is a stop list? **Answer:** A list of “stop words”, very common words that are of little help in (non-phrase) information retrieval

**Question**

What is lemmatization? Give an example. **Answer:** Reducing inflected form to the base dictionary form, the lemma. Example: giving → give

**Question**

What is stemming? Give an example that is not also a lemmatization example. **Answer:** Reducing a word to a “stem” (a distinctive prefix) by heuristic rules. increment → incr

**Question**

Name a particular stemmer. **Answer:** Porter stemmer

**Question**

Give an example of a pair of words that a typical stemmer would put in one equivalence class and we would expect improved performance of the IR system. **Answer:** sweater/sweaters, tour/tours

**Question**

Give an example of a pair of words that a typical stemmer would put in one equivalence class and we would expect decreased performance of the IR system. **Answer:** operational/operating/operative/operates

**Question**

Name two data structures that support phrase queries.

**Answer:** biword index, positional inverted index

**Question**

Name a data structure that supports proximity queries.

**Answer:** positional inverted index

**Chapter 3****Question**

Which data structures are typically used for locating the entry for a term in the dictionary?

**Answer:** either a hash or a tree

**Question**

Which data structure is best used for locating the entry for a term in the dictionary if the collection is static?

**Answer:** a hash

**Question**

Which data structure is best used for locating the entry for a term in the dictionary if prefix search must be supported?

**Answer:** a tree

**Question**

Which special strings are stored in the permuterm index for the word “car”?

**Answer:** car\$, ar\$c, r\$ca, \$car

**Question**

What sequence of letters is looked up in the permuterm index for the following wildcard queries?

X, X\*, \*X, \*X\*, X\*Y

**Answer:**

X\$, \$X\*, X\$, X\*, Y\$X\*

**Question**

What is the difference between the regular inverted index used in IR and the k-gram index?

**Answer:** a postings list in the regular index is the list of documents that a particular term occurs in; a postings list in a k-gram index is a list of words that a k-gram occurs in

**Question**

Give an example of a query that cannot be corrected using isolated-word spelling correction.

**Answer:** flights form london

**Question**

Define Levenshtein edit distance.

**Answer:** the minimum number of inserts, deletes, replaces needed to transform one string into another

**Question**

Define Damerau-Levenshtein edit distance.

**Answer:** the minimum number of inserts, deletes, replaces, transposes needed to transform one string into another

**Chapter 6****Question**

What is the feast or famine problem?

**Answer:** In Boolean search it's often difficult to write a query that produces a small number of results. Often you get back either nothing or a large number of hits.

**Question**

Define the Jaccard coefficient

**Answer:**  $\frac{|A \cap B|}{|A \cup B|}$

**Question**

What is the bag of words model?

**Answer:** The representation of a document is a bag of words: the order of words is ignored and only the count of the word in the document considered

**Question**

What is the advantage of idf weighting compared to inverse-collection-frequency weighting?

**Answer:** Of two terms with the same collection frequency, the one that occurs in fewer documents receives a higher idf weight. This indicates "clustered" occurrence in a documents, which is a content word property.

**Question**

What is the tf-idf weight of term t in document d?

**Answer:**  $w_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t}$

**Question**

What is the relationship between term frequency and collection frequency?

**Answer:** The sum of all term frequencies of a term in a collection is its collection frequency.

**Question**

Why don't we use Euclidean distance of tf-idf vectors to rank documents with respect to a query?

**Answer:** Because document length strongly influences such a ranking and length should not be considered (or only very slightly) in relevance ranking

**Question**

Write down the formula for cosine similarity between query  $q$  and document  $d$ .

**Answer:** 
$$\cos(\vec{q}, \vec{d}) = \text{SIM}(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}| |\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2} \sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

**Question**

Explain the notation ddd.qqq

**Answer:** The six letters stand for term frequency weighting, document frequency weighting and normalization of the document (first three) and the query (last three)

**Chapter 7****Question**

What is the advantage of pivot normalization compared to regular cosine normalization?

**Answer:** Cosine normalization ranks short documents too high and long documents too low. Pivot normalization treats short and long documents more equally.

**Question**

What is document-at-a-time processing?

**Answer:** Computation of a score of one document is completed before computation of the score of the next document is started.

**Question**

What index organization does document-at-a-time processing require?

**Answer:** There must be one consistent global ordering that all postings lists adhere to.

**Question**

What is term-at-a-time processing?

**Answer:** The postings list of one query term is completely processed before processing of the postings list of the next query term is started.

**Question**

What data structure does term-at-a-time processing require that document-at-a-time processing does not require?

**Answer:** An array of accumulators, one for each document, that stores document scores.

**Question**

What is a tiered inverted index?

**Answer:**

A small proportion of documents that collectively is likely to satisfy most queries is put in tier 1, so that most queries can be served from a small index (the tier 1 index).

**Question**

Name two criteria that can be used for deciding as to whether to put a document  $d$  in tier 1 of a tiered index.

**Answer:** (i) Term occurs in title of  $d$ . (ii) Term occurs in anchor text of a document pointing to  $d$ .

**Chapter 8****Question**

Name three criteria for evaluating a search engine.

**Answer:** relevance of results, response time, comprehensiveness of index, cost of running a query ...

**Question**

What are the components of an information retrieval benchmark?

**Answer:** set of queries, set of documents, set of query-document relevance judgments

**Question**

What is the difference between the concepts of query and information need?

**Answer:** A query on the web usually is a set of keywords. An information need is a detailed description of what the user is looking for. Most queries correspond to several very different information needs.

**Question**

Define precision

**Answer:**

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant}|\text{retrieved})$$

**Question**

Define recall

**Answer:**

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved}|\text{relevant})$$

**Question**

Define  $F_1$

**Answer:**

$$\frac{2PR}{P+R}$$

**Question**

What is the harmonic mean of two numbers?

**Answer:**

$$\frac{1}{F} = \frac{1}{2} \left( \frac{1}{P} + \frac{1}{R} \right)$$

**Question**

Why is  $F_1$  defined as the harmonic mean?

**Answer:** The harmonic mean is a kind of “soft” minimum. Roughly, the performance of a system is as good as the minimum of precision and recall

**Question**

What is an easy way of maximizing the recall of an IR engine?

**Answer:**

return the entire collection to the user

**Question**

What is an easy way of maximizing the precision of an IR engine?

**Answer:**

return only one document – often it is relatively easy to find one relevant document

**Question**

What is a precision-recall curve?

**Answer:**

Each point on the curve corresponds to a position  $n$  in the ranked list. The x-coordinate of the point is the recall at position  $n$ , the y-coordinate is the precision at position  $n$ .

**Question**

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?

**Answer:**

the total number of relevance judgments is the product of the number of queries and the number of documents. relevance judgments are expensive and cannot be done exhaustively for larger collections

**Question**

Define the kappa measure

**Answer:**

$P(A)$  = proportion of time judges agree

$P(E)$  = what agreement would we get by chance

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

**Question**

What is the minimum and maximum of the kappa measure?

**Answer:**

$(-\infty, 1]$

**Question**

What is the significance of kappa being less than / greater than 0?

**Answer:**

The judgements of the two judges are negatively / positively correlated. Or: the judges agree less / more than would be expected by chance.

**Question**

What is A/B testing?

**Answer:**

To test a new feature, implement it in a new system  $S'$  and then redirect a small fraction of your traffic from the old system  $S$  to  $S'$ . If users like  $S'$  better than  $S$  in this experiment, then roll out the new feature to all users.

**Question**

What does marginal relevance measure?

**Answer:**

The additional information that the document at position  $n$  in the ranked list provides to the user, i.e., the information that is not present in the  $n - 1$  more highly ranked documents.

**Question**

What distinguishes a dynamic from a static summary?

**Answer:**

The dynamic summary of document  $d$  is computed for a particular query  $q$ . For two different queries, the dynamic summaries of  $d$  will in general be different. The static summary is always the same.

**Question**

What is a simple heuristic for computing a dynamic summary if you can display  $n$  characters?

**Answer:**

If all query terms occur in a window of size  $n$  in the document display that window. Otherwise display discontinuous windows around the query terms.

**Chapter 9****Question**

What is the difference between adhoc retrieval and relevance feedback?

**Answer:**

Adhoc retrieval: user enters query, IR engine returns results.

Relevance feedback: user enters query, IR engine returns results, user marks results as relevant/nonrelevant, IR engine returns better results.

**Question**

Give the mathematical definition of the centroid

**Answer:**  $\vec{\mu}(D) = \frac{1}{|D|} \sum_{d \in D} \vec{v}(d)$

**Question**

In Rocchio's algorithm, what weight setting for  $\alpha/\beta/\gamma$  does a 'Find pages like this one' search correspond to?

**Answer:** 'Find pages like this one' ignores the query and no negative judgments are used. Hence  $\alpha = \gamma = 0$ . This implies  $\beta = 1$ .

**Question**

Why is relevance feedback not used by most search engines?

**Answer:**

(i) Users don't like to give time-consuming feedback on individual documents. (ii) Relevance feedback queries are long and expensive for the search engine to service.

**Question**

What is the difference between relevance feedback and manual query expansion?

**Answer:**

In relevance feedback, the user judges documents and the search engine automatically computes a new query. In manual query expansion, the user judges terms or alternative queries and directly "authors" the new query.

**Question**

Give an example of ineffective automatic query expansion

**Answer:**

"interest rate"  $\rightarrow$  "interest rate fascinate"

**Question**

Search engines log the sequence of queries that a user issues during a session. How can this be exploited for query expansion?

**Answer:**

If users often issue  $q_2$  after issuing  $q_1$ , then  $q_2$  is a good candidate to suggest after query  $q_1$ .

**Question**

Search engines log the documents users click on in response to a query. How can this be exploited for query expansion?

**Answer:**

If users frequently click on document  $d$  after issuing  $q_1$  and also frequently click on  $d$  after issuing  $q_2$ , then  $q_1$  and  $q_2$  are good expansions of each other.

**Chapter 13****Question**

What is the machine learning approach to text classification?

**Answer:**

A classifier is trained on a training set where each document is labeled with one of the classes in a set of classes. When applied to a new document, the classifier returns the predicted class of this document.

**Question**

Give one advantage and one disadvantage of rule-based classifiers compared to machine-learned classifiers.

**Answer:**

Advantages: can be carefully tuned to do exactly what you want it do to; don't need training set

Disadvantages: complicated classification rules are difficult to program and debug; you need an expert do build this type of system

**Question**

What is bad about maximum likelihood estimates of the parameters  $P(t|c)$  in Naive Bayes.

**Answer:**

These will often be zero. A zero parameter results in a zero probability for the class  $c$ , effectively giving a single rare word veto power for class  $c$ .

**Question**

What is the time complexity of training a Naive Bayes classifier and why?

**Answer:**

linear in the size of the training set; counts can be computed while scanning the corpus. the actual computation of the parameters is also linear since it is bounded by the number of classes times the size of the training set

**Question**

What is the main independence assumption of Naive Bayes.

**Answer:**

Given a class, the occurrence of terms in a document is independent.

**Question**

What is feature selection?

**Answer:**

Feature selection evaluates each term (word) and assigns it a utility. It then selects the top  $k$  terms according to this utility and ignores the rest.

**Question**

What is feature selection used? Give the two main reasons?

**Answer:**

(i) to get rid of noise features (ii) to increase efficiency

**Question**

In words: what is the meaning of mutual information when used for features selection in text classification?

**Answer:**

MI measures the amount of information a term occurrence has about document class membership and vice versa.

**Chapter 12****Question**

What is the basic idea of the language model approach to adhoc IR?

**Answer:**

---

Estimate a language model for each document. The top-ranked document is then the one that is most likely to have generated the query.

**Question**

How is length normalization performed in the vector space model vs the language model approach to IR?

**Answer:**

vectors space: cosine/pivot normalization; language model approach: probabilities estimated as relative frequencies are inherently length normalized

**Question**

What is the number of classes in the Naive Bayes approach to classification? What is the number of classes in the language model approach to IR?

**Answer:** classification: the number of classes is given by the application (e.g., spam/ham)

language models: the number of classes equals the number of documents

**Chapter 15****Question**

What is the definition of a linear classifier?

**Answer:**

A linear classifier computes a linear combination or weighted sum  $\sum_i w_i x_i$  of the feature values.

Classification decision:  $\sum_i w_i x_i > \theta$ ?

**Question**

For linearly separable problem, how many different linear decision boundaries are there that separate the two classes of the training set perfectly?

**Answer:**

infinitely many

**Question**

Which decision boundary does the linear SVM choose?

**Answer:**

The hyperplane with the largest margin

**Question**

What is a support vector?

**Answer:**

A training set vector on the margin

**Question**

How does an SVM classify a test set point in the margin

**Answer:**

either: classify based on decision boundary; or: make no decision

**Chapter 16****Question**

What is the difference between classification and clustering?

**Answer:**

The classes are defined by a human. The clusters are induced from the data.

**Question**

Why is result set clustering useful?

**Answer:**

If a search term is ambiguous, then clustering the search result may group all hits of a particular sense together. This is easier for the user to understand than a list of links.

**Question**

What is hard/soft clustering?

**Answer:**

hard clustering: a document is in exactly one cluster

soft clustering: a document can be a “soft” (weighted) member of several clusters

**Question**

Does K-means always converge and why?

**Answer:**

It always converges because RSS decreases in each step and there is only a finite number of clusterings, so a minimum must be reached in finite time.

**Question**

Does K-means find the global optimum and why?

**Answer:**

No, it sometimes converges to a local optimum. To get from the local optimum to the global optimum, RSS would have to be increased temporarily. K-means doesn't do that.

**Chapter 17****Question**

Name three properties of a dendrogram

**Answer:**

- (i) a horizontal line corresponds to a merger of two clusters
- (ii) the height of a horizontal line is the similarity of the corresponding merger
- (iii) the dendrogram records the temporal sequence of mergers

**Question**

What two different types of hierarchical clustering were introduced in class?

**Answer:**

top-down: start with one big cluster, keep splitting

bottom-up: start with each doc being its own cluster, keep merging

**Question**

What is the difference single-link and complete-link HAC?

**Answer:**

single-link: similarity of two clusters is computed as maximum similarity between pairs of members of the two clusters

complete-link: similarity of two clusters is computed as minimum similarity between pairs of members of the two clusters

**Question**

What is chaining?

**Answer:**

The tendency of single-link clustering to form long, elongated clusters

**Question**

In general, is single-link or complete-link better to use and why?

**Answer:**

complete-link because it's not afflicted by chaining

**Chapter 21****Question**

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

**Answer:**

(i) A hyperlink is a quality signal. (ii) The anchor text describes the content of the page the hyperlink points to.

**Question**

What is a Google bomb? Give an example

**Answer:**

A bad search result due to maliciously manipulated anchor text. Examples: evil empire - Microsoft, dumb motherf\*\*\*\*r - George W Bush, dangerous cult - Scientology

**Question**

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

**Answer:**

A simple count does not weight the quality of the pointing pages. A page pointed to by 10 high-quality pages is better than a page pointed to by 10 low-quality pages.

**Question**

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

**Answer:**

$q$  is the probability that the random surfer will be on page  $d$  at a particular point in time.

**Question**

What is ergodicity and why is it important for PageRank?

**Answer:**

---

ergodic = aperiodic (no periodic behavior) and irreducible (roughly: there is a path from every page to every other page)

PageRank is well-defined if surfing the web graph is ergodic

## Chapter 19

### Question

What was the Goto model and what was bad about it?

#### Answer:

Goto ranked web pages for a query  $q$  according to how much an advertiser was willing to pay for a click on their web page. This is bad because users use web search engines to get relevant information about a query. They will stop using a search engine that instead gives them information somebody has paid for to be shown.

### Question

What are the advantages of a search engine ad compared to other types of ads (radio, television, newspaper)?

#### Answer:

(i) We know the user's search (e.g., "buy ford") – this is an important indicator of what the user may be willing to buy (a car, car insurance, etc). (ii) The advertiser only pays if the user actively interacts with the ad. (= clicks on the ad)

### Question

What is the problem with duplicates and near duplicates in terms of relevance to the user?

#### Answer:

A duplicate can be relevant in absolute terms: if the user saw it in isolation, it would be a good page to return. But the near duplicate has zero marginal relevance if the same content occurs higher up in the search result ranking.

### Question

What is the advantage of shingling/sketches for near duplicate identification compared to computing tf-idf similarity scores between documents?

#### Answer:

(i) Sketches are small, so we need a lot less space to identify duplicates. (ii) tf-idf similarity is computed on words – n-grams (shingles) are a much more selective measure of similarity than words.

### Question

How would you eliminate exact duplicates?

**Answer:**

(i) Compute a fingerprint (or hash code) of each document. (ii) Only show one document with given fingerprint  $j$  in the search result – or only include one document with given fingerprint  $j$  in the index.

**Part 2: Exercises****Chapter 2****Exercise**

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 at which positions, where each expression within quotes is a phrase query? (i) “fools rush in” (ii) “fools rush in” AND “angels fear to tread”.

**Answer:**

(i) doc2:1, doc4:8, doc7:3,13 (ii) doc4:8&12

**Chapter 3****Exercise**

Compute the Levenshtein matrix for the distance between the strings “apfel” (input) and “poems” (output). Use this format (as introduced in class):

		f	a	s	t
	0	1 1	2 2	3 3	4 4
c	1 1	1 2 2 1	2 3 2 2	3 4 3 3	4 5 4 4
a	2 2	2 2 3 2	1 3 3 1	3 4 2 2	4 5 3 3
t	3 3	3 3 4 3	3 2 4 2	2 3 3 2	2 4 3 2
s	4 4	4 4 5 4	4 3 5 3	2 3 4 2	3 3 3 3

**Answer:**

		p	o	e	m	s
	0	1 1	2 2	3 3	4 4	5 5
a	1 1	1 2 2 1	2 3 2 2	3 4 3 3	4 5 4 4	5 6 5 5
p	2 2	1 2 3 1	2 3 2 2	3 4 3 3	4 5 4 4	5 6 5 5
f	3 3	3 2 4 2	2 3 3 2	3 4 3 3	4 5 4 4	5 6 5 5
e	4 4	4 3 5 3	3 3 4 3	2 4 4 2	4 5 3 3	5 6 4 4
l	5 5	5 4 6 4	4 4 5 4	4 3 5 3	3 4 4 3	4 5 4 4

**Exercise**

We saw in class that the Levenshtein sequence of operations for converting strings into each other is not unique. For example, “cat” can be transformed into “catcat” either by insert, insert, insert, copy, copy, copy or by copy, copy, copy, insert, insert, insert. In contrast, the minimum number Levenshtein operations with cost 1 for converting one string to another is fixed since the minimum is unique. Let  $n_i, n_d, n_r$  be the number of inserts, deletes and replaces in a sequence of operations. Give an example of a pair of strings and two different sequences of operations  $\sigma_1$  and  $\sigma_2$  that convert the first string into the second such that  $n_i(\sigma_1) \neq n_i(\sigma_2)$  or  $n_d(\sigma_1) \neq n_d(\sigma_2)$  or  $n_r(\sigma_1) \neq n_r(\sigma_2)$ . Or prove that this is not possible.

**Answer:**

The Levenshtein distance of the strings “ab” and “ba” is 2. Two possible sequences of operations are (i) replace a with b, replace b with a (ii) delete a, copy b, insert a. In this example,  $0 = n_i(\sigma_1) \neq n_i(\sigma_2) = 1$ ,  $0 = n_d(\sigma_1) \neq n_d(\sigma_2) = 1$ ,  $2 = n_r(\sigma_1) \neq n_r(\sigma_2) = 0$ . Thus, the number of inserts, deletes and replaces need not be the same.

**Chapter 6/7**

**Exercise**

Compute the Inc.ltn similarity between the query “digital phones” and the document “digital phones and video phones and other phones” by filling out the empty columns in the table below. Assume  $N = 10,000,000$ . Treat *and* and *other* as stop words. What is the final similarity score? What is the corresponding Jaccard score?

word	query					document				product
	tf-raw	tf-wght	df	idf	weight	tf-raw	tf-wght	weight	n'lized	
digital			10,000							
video			100,000							
phones			50,000							

**Answer:**

$$Jaccard(Q, D) = \frac{|Q \cap D|}{|Q \cup D|} = \frac{| \{digital, phones\} |}{| \{digital, video, phones\} |} = \frac{2}{3}$$

word	query					document				product
	tf-raw	tf-wght	df	idf	weight	tf-raw	tf-wght	weight	n'lized	
digital	1	1	10,000	3	3	1	1	1	0.49	1.47
video	0	0	100,000	2	0	1	1	1	0.49	0
phones	1	1	50,000	2.3	2.3	3	1.5	1.5	0.73	1.68

Length of document vector =  $\sqrt{1^2 + 1^2 + 1.5^2} \approx 2.06$

$1/2.06 \approx 0.49$ ,  $1.5/2.06 \approx 0.73$ ,

Similarity score:  $1.47 + 1.68 = 3.15$

### Exercise

One measure of the similarity of two vectors is the Euclidean distance between them:  $|\vec{x} - \vec{y}| = \sqrt{\sum_{i=1}^M (x_i - y_i)^2}$ . Given a query  $q$  and documents  $d_1, d_2, \dots$ , we may rank the documents  $d_i$  in order of increasing Euclidean distance from  $q$ . Show that if  $q$  and the  $d_i$  are all normalized to unit vectors, then the rank ordering produced by Euclidean distance is identical to that produced by cosine similarities.

**Answer:**

$$\sum (q_i - w_i)^2 = \sum q_i^2 - 2 \sum q_i w_i + \sum w_i^2 = 1 - 2 \sum q_i w_i + 1 = 2(1 - \sum q_i w_i)$$

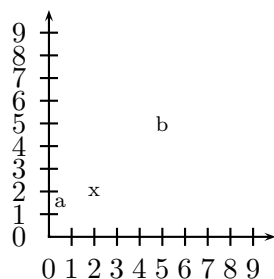
(Note that for a normalized vector  $\vec{x}$ , we have:  $\sum x_i^2 = 1$ .)

$$\text{Thus: } |\vec{q} - \vec{v}| < |\vec{q} - \vec{w}| \Leftrightarrow |\vec{q} - \vec{v}|^2 < |\vec{q} - \vec{w}|^2 \Leftrightarrow \sum (q_i - v_i)^2 < \sum (q_i - w_i)^2 \Leftrightarrow 2(1 - \sum q_i v_i) < 2(1 - \sum q_i w_i) \Leftrightarrow \sum q_i v_i > \sum q_i w_i \Leftrightarrow \cos(\vec{q}, \vec{v}) > \cos(\vec{q}, \vec{w})$$

This proves that ordering normalized vectors according to increasing distance is the same as ordering them according to decreasing cosine similarity.

### Exercise

In the figure below, which of the three vectors  $\vec{a}$ ,  $\vec{b}$ , and  $\vec{c}$  is (i) most similar to  $\vec{x}$  according to dot product similarity ( $\sum_i x_i \cdot y_i$ ), (ii) most similar to  $\vec{x}$  according to cosine similarity ( $\sum_i x_i \cdot y_i / (|x||y|)$ ), (iii) closest to  $\vec{x}$  according to Euclidean distance? The vectors are  $\vec{a} = (0.5 \ 1.5)^T$ ,  $\vec{x} = (2 \ 2)^T$ ,  $\vec{b} = (5 \ 5)^T$ , and  $\vec{c} = (11 \ 8)^T$ . Compute the relevant dot products, cosines and distances. Assume that higher dot product indicates higher similarity.



**Answer:**

(i)  $\vec{c}$  (dot products: 4, 20, 38)

(ii)  $\vec{b}$  (cosines: 0.8944, 1.0, 0.9878)

(iii)  $\vec{a}$  (distances: 1.58, 4.24, 10.82)

## Chapter 8

### Exercise

Below is a table showing how two human judges assigned documents to the class “English” (0 = is not written in English, 1 = is written in English). Let us assume that you’ve written a classifier that assigns the documents {2, 5, 6, 7, 8} to “English”.

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

(i) Calculate precision, recall, and  $F_1$  of your system if a document is considered relevant only if the two judges agree it is relevant. (ii) Calculate precision, recall, and  $F_1$  of your system if a document is considered relevant if either judge thinks it is relevant.

**Answer:**

(i)  $P = R = F_1 = 0$  (ii)  $P = 4/5$

$R = 4/10 = 2/5$

$F_1 = 2 \cdot 4/5 \cdot 4/10 / (4/5 + 4/10) = 16/25 / (6/5) = 8/15$

## Chapter 9

### Exercise

Suppose that a user’s initial query is “cheap CDs cheap DVDs extremely cheap CDs”. The user examines two documents,  $d_1$  and  $d_2$ . She judges  $d_1$ , with the content “CDs software cheap CDs” relevant and  $d_2$  with content “cheap thrills DVDs” nonrelevant. Assume that we are using direct term frequency (with no scaling and no document frequency). There is no need to length-normalize vectors. Using Rocchio relevance feedback as in Equation 9.3 what would the revised query vector be after relevance feedback? Assume  $\alpha = 1, \beta = 0.8, \gamma = 0.2$ .

**Answer:**

word	$q$	$d_1$	$d_2$	$\alpha q$	$\beta d_1$	$\gamma d_2$	rocchio
CDs	2	2	0	2	1.6	0	3.6
cheap	3	1	1	3	0.8	0.2	3.6
DVDs	1	0	1	1	0	0.2	0.8
extremely	1	0	0	1	0	0	1.0
software	0	1	0	0	0.8	0	0.8
thrills	0	0	1	0	0	0.2	0.0

## Chapter 13

### Exercise

Based on the data below, estimate a multinomial Naive Bayes classifier (the type of NB classifier we introduced in class) and apply the classifier to the test document.

	docID	words in document	in $c = \text{Japan?}$
training set	1	Kyoto Osaka Taiwan	yes
	2	Japan Kyoto	yes
	3	Taipei Taiwan	no
	4	Macao Taiwan Shanghai	no
test set	5	Taiwan Taiwan Kyoto	?

**Answer:**

$$P(c|d) \propto \hat{P}(c) \prod_k \hat{P}(t_k|c)$$

$$\hat{P}(t_k|c) := \frac{C(t_k, c) + 1}{\sum_t C(t, c) + |V|}$$

(i)  $\hat{P}(c) = \hat{P}(\bar{c}) = 1/2$ . The vocabulary has 7 terms: Japan, Macao, Osaka, Kyoto, Shanghai, Taipei, Taiwan. There are 5 tokens in the concatenation of all  $c$  documents. There are 5 tokens in the concatenation of all  $\bar{c}$  documents. Thus, the denominators have the form  $(5+7)$ .

$$\hat{P}(Taiwan|c) = (1 + 1)/(5 + 7) = 2/12$$

$$\hat{P}(Taiwan|\bar{c}) = (2 + 1)/(5 + 7) = 3/12$$

$$\hat{P}(Kyoto|c) = (2 + 1)/(5 + 7) = 3/12$$

$$\hat{P}(Kyoto|\bar{c}) = (0 + 1)/(5 + 7) = 1/12$$

(ii) We then get

$$\hat{P}(c|d) \propto 1/2 \cdot (2/12)^2 \cdot 3/12 = 12/(12 \cdot 12 \cdot 12)$$

$$\hat{P}(\bar{c}|d) \propto 1/2 \cdot (3/12)^2 \cdot (1/12) = 9/(12 \cdot 12 \cdot 12)$$

$$\frac{\hat{P}(c|d)}{\hat{P}(\bar{c}|d)} = 4/3$$

Thus, the classifier assigns the test document to  $c = \text{Japan}$ .

## Chapter 12

### Exercise

Rank the documents in collection  $\{d_1, d_2\}$  for query  $q$  using the language model approach to IR introduced in class. Use Jelinek-Mercer smoothing with the mixture coefficient  $\lambda = 0.4$ .

- $d_1$ : Scottish sheep getting smaller due to climate change study says
- $d_2$ : The analysis has shown a dramatic shift in the natural ranges for US Bird species in response to climate change
- Query  $q$ : climate change

**Answer:**

$$P(q|d) = \prod_l [\lambda P(t_K|M_d) + (1 - \lambda)P(t_k|M_c)]$$

$$P(q|d_1) = [0.4 \cdot 1/10 + 0.6 \cdot 2/30] \cdot [0.4 \cdot 1/10 + 0.6 \cdot 2/30] = [0.4 \cdot 1/10 + 0.2 \cdot 0.2]^2 = 0.08^2 = 0.0064$$

$$P(q|d_2) = [0.4 \cdot 1/20 + 0.6 \cdot 2/30] \cdot [0.4 \cdot 1/20 + 0.6 \cdot 2/30] = [0.4 \cdot 1/20 + 0.2 \cdot 0.2]^2 = 0.06^2 = 0.0036$$

Ranking:  $d_1 > d_2$

## Chapter 15

### Exercise

The decision boundary of the support vector machine  $S$  is defined by:

$$(2 \quad -1)\vec{x} - 4 = 0$$

In this exercise, use the labels +1 and -1 for the two classes.

(i) Let  $S$  be an SVM that doesn't make a decision for points in the margin. Which of the following points does  $S$  make a decision on and what is the decision?

$$\vec{a} = (1.01 \quad 2)^T, \vec{b} = (1 \quad 1.99)^T, \vec{c} = (2 \quad 2)^T, \vec{d} = (0 \quad 0)^T$$

(ii) Let  $S$  be an SVM that always makes a decision, even for points in the margin. In this case, what is the decision for the four points  $\vec{a}$ ,  $\vec{b}$ ,  $\vec{c}$ , and  $\vec{d}$ ?

**Answer:** (i)  $\vec{a}$ : no decision,  $\vec{b}$ : no decision,  $\vec{c} \rightarrow +1$ ,  $\vec{d} \rightarrow -1$

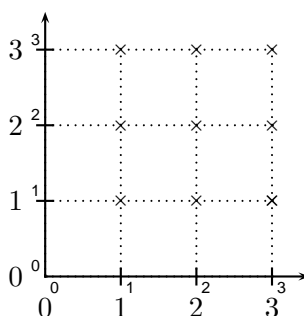
(ii)  $\vec{a} \rightarrow +1$ ,  $\vec{b} \rightarrow -1$ ,  $\vec{c} \rightarrow +1$ ,  $\vec{d} \rightarrow -1$

## Chapter 16

### Exercise

a) Perform a 3-means clustering for the points below. 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.

b) There are several clusterings that 3-means can converge to in this case. Give an example of such a clustering that is different from the one in a.



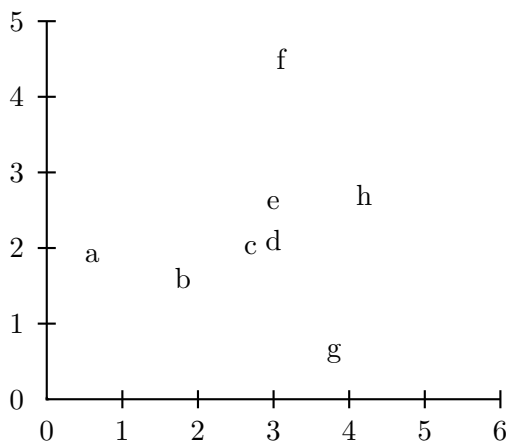
**Answer:**

i) Choose as seeds: (2, 3), (2, 2), (2, 1). Reassignment:  $\{(1, 3), (2, 3), (3, 3)\}$ ,  $\{(1, 2), (2, 2), (3, 2)\}$ ,  $\{(1, 1), (2, 1), (3, 1)\}$ . Recomputation: (2, 3), (2, 2), (2, 1). *K*-means has converged.

ii) A different clustering is:  $\{(3, 1), (3, 2), (3, 3)\}$ ,  $\{(2, 1), (2, 2), (2, 3)\}$ ,  $\{(1, 1), (1, 2), (1, 3)\}$ .

**Chapter 17**

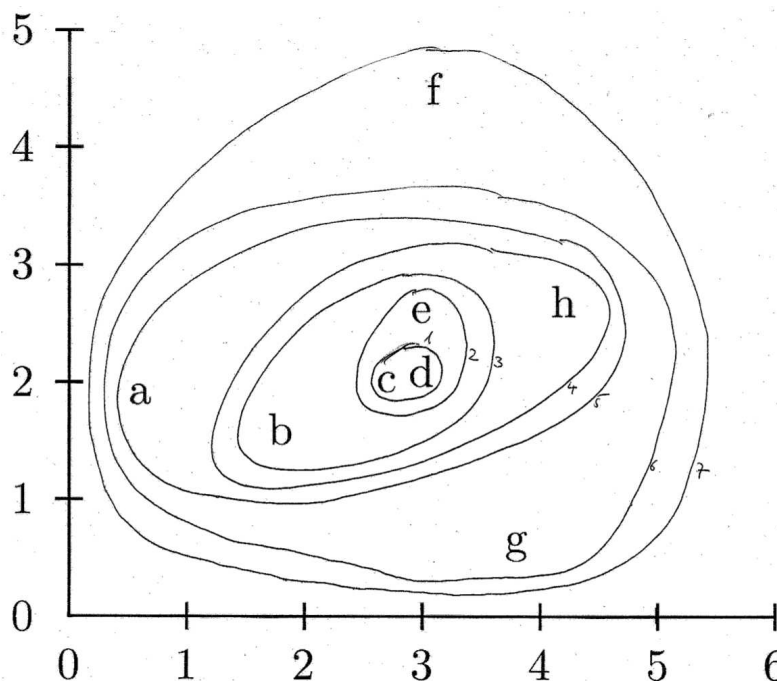
**Exercise**



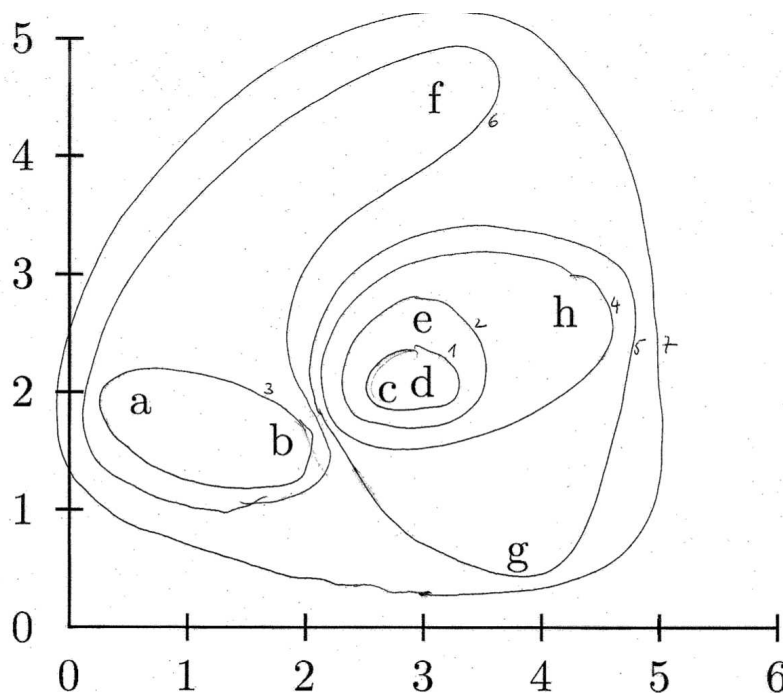
The points have the following coordinates: a: (0.6,1.9), b: (1.8,1.6), c: (2.7,2.0), d: (3.0,2.1), e: (3.0,2.6), f: (3.1,4.5), g: (3.8,0.6), h: (4.2,2.7). Define the similarity of two points as  $-(x_1 - x_2)^2 - (y_1 - y_2)^2$ .

Compute (i) single-link and (ii) complete-link clusterings of this set of points. You can do this visually by drawing circles/ellipses. Mark each circle/ellipse with a number indicating the temporal sequence of mergers.

**Answer:**

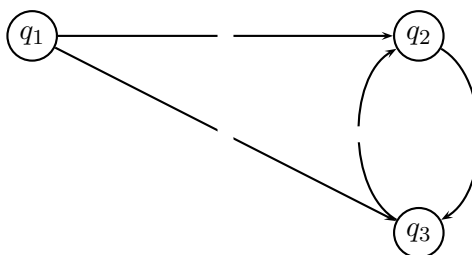


single-link clustering:



complete-link clustering:

### Chapter 21



Compute PageRank for the web graph in Figure for each of the three pages. Also give the relative ordering of the 3 nodes indicating any ties.

Assume that at each step of the PageRank random walk, we teleport to a random page with probability 0.1, with a uniform distribution over which particular page we teleport to.

Using symmetries to simplify and solving with linear equations might be easier than using iterative methods.

**Answer:** Since the in-degree of A is 0, the steady-visit rate (or rank) of A is  $0.1 \cdot 1/3 = 1/30$  (from teleport). By symmetry,  $\text{rank}(B) = \text{rank}(C)$ . Thus,  $\text{rank}(B)=\text{rank}(C) = 29/60$ .

Solution using power method:

		q1	q2	q3
Transition matrix without teleport:	q1	0	$1/2$	$1/2$
	q2	0	0	1
	q3	0	1	0
		q1	q2	q3
Transition matrix with teleport:	q1	$1/30$	$29/60$	$29/60$
	q2	$1/30$	$1/30$	$14/15$
	q3	$1/30$	$14/15$	$1/30$

code:

```

q1 = 0
q2 = 1
q3 = 0
for n in range(101):
    q1new = q1*1/30.+q2*1/30.+q3*1/30.
    q2new = q1*29/60.+q2*1/30.+q3*14/15.
    q3new = q1*29/60.+q2*14/15.+q3*1/30.
    q1 = q1new
    q2 = q2new
    q3 = q3new
    print n,q1,q2,q3

```

For initialization: (0,1,0)

```

0 0.03333333333333 0.03333333333333 0.93333333333333
1 0.03333333333333 0.88833333333333 0.07833333333333
2 0.03333333333333 0.11883333333333 0.84783333333333
3 0.03333333333333 0.81138333333333 0.15528333333333
4 0.03333333333333 0.18808833333333 0.77857833333333
5 0.03333333333333 0.74905383333333 0.21761283333333
6 0.03333333333333 0.24418488333333 0.72248178333333
7 0.03333333333333 0.69856693833333 0.26809972833333
8 0.03333333333333 0.28962308883333 0.67704357783333
9 0.03333333333333 0.65767255338333 0.30899411328333
...
96 0.03333333333333 0.483315115639 0.483351551028
97 0.03333333333333 0.483349729259 0.483316937408
98 0.03333333333333 0.483318577001 0.483348089666
99 0.03333333333333 0.483346614033 0.483320052634
100 0.03333333333333 0.483321380704 0.483345285963

```

For initialization: (1/3,1/3,1/3)

Convergence in the first iteration!

```

0 0.03333333333333 0.48333333333333 0.48333333333333

```

## Chapter 19

### Exercise

advertiser	bid	CTR
A	\$5.00	0.04
B	\$1.00	0.1
C	\$0.50	0.06
D	\$1.00	0.03

Compute how much advertisers A, B, C, D have to pay for each click in a second price auction as described in class. The minimum amount per click is 0.005.

**Answer:**

```
[('a', 5, 0.040000000000000001), ('b', 1,
 0.100000000000000001), ('c', 0.5, 0.059999999999999998),
 ('d', 1, 0.029999999999999999)]
price per click paid by a is 2.51
price per click paid by b is 0.31
price per click paid by c is 0.51
price per click paid by d is 0.005
```

Code:

```
def secondpriceauction(myargs):
    minimum = 0.005
    names = ['a','b','c','d']
    #bids = [4,3,2,1]
    #ctrs = [.01,.03,.06,.08]
    bids = [5,1,.5,1]
    ctrs = [.04,.1,.06,.03]
    tuples = ((names[i],bids[i],ctrs[i]) for i in range(4))
    tuples2 = sorted(tuples,key=lambda tuple: tuple[1]*tuple[2],reverse=True)
    print tuples2
    for i,tuple in enumerate(tuples2):
        if i+1>=len(tuples2):
            print 'price per click paid by',tuple[0],'is',minimum
        else:
            auctionmin = tuples2[i+1][1]*tuples2[i+1][2]/tuple[2]
            print 'price per click paid by',tuple[0],'is',auctionmin+0.01
```

### Exercise

The shingle representations of three documents are as follows:  $d_3 = (0, 0, 1, 0, 0, 0, 1)^T$ ,  $d_4 = (0, 0, 1, 0, 0, 0, 0)^T$ ,  $d_5 = (1, 1, 1, 0, 1, 1, 1)^T$

We will use sketches of size 2. The two elements of a sketch are defined by the permutations.  $(3 \times n + 2) \bmod 7$  and  $(5 \times n + 1) \bmod 7$ . Based on this setup what are the estimates of the three Jaccard coefficients  $J(d_3, d_4)$ ,  $J(d_3, d_5)$ , and  $J(d_4, d_5)$ ?

**Answer:**

sketches:  $d_3 = (2, 1)$ ,  $d_4 = (4, 2)$ ,  $d_5 = (1, 1)$

$\hat{J}(d_3, d_4) = 0$ ,  $\hat{J}(d_3, d_5) = 1/2$ , and  $\hat{J}(d_4, d_5) = 0$ ?

to compute sketches run permutationmin for selectedperm=2 and selectedperm=3

```
def permutationmin(myargs):
    selectedperm = myargs['selectedperm']
    def myperm0(n):
        return n % 5
    def myperm1(n):
        return (2*n+1) % 5
    def myperm2(n):
        return (3*n+2) % 7
    def myperm3(n):
        return (5*n+1) % 7
    myperm = {}
```

```
i = 0
myperm[i] = myperm0
i += 1
myperm[i] = myperm1
i += 1
myperm[i] = myperm2
i += 1
myperm[i] = myperm3
d1 = [1,0,1,1,0]
d2 = [0,1,1,0,1]
d3 = [0,0,1,0,0,0,1]
d4 = [0,0,1,0,0,0,0]
d5 = [1,1,1,0,1,1,1]
myperm = myperm[selectedperm]
for mydoc in [d1,d2,d3,d4,d5]:
    mymin = 1000000
    for i in range(len(mydoc)):
        if mydoc[i]==0: continue
        myindex = i+1
        permuted = myperm(myindex)
        if permuted<mymin:
            mymin = permuted
    print 'min for doc',mydoc,'is',mymin
```

Output:

```
python munge.py permutationmin 2
parameters: {'selectedperm': 2}
min for doc [1, 0, 1, 1, 0] is 0
min for doc [0, 1, 1, 0, 1] is 1
min for doc [0, 0, 1, 0, 0, 0, 1] is 2
min for doc [0, 0, 1, 0, 0, 0, 0] is 4
min for doc [1, 1, 1, 0, 1, 1, 1] is 1
```

```
python munge.py permutationmin 3
parameters: {'selectedperm': 3}
min for doc [1, 0, 1, 1, 0] is 0
min for doc [0, 1, 1, 0, 1] is 2
min for doc [0, 0, 1, 0, 0, 0, 1] is 1
min for doc [0, 0, 1, 0, 0, 0, 0] is 2
min for doc [1, 1, 1, 0, 1, 1, 1] is 1
```