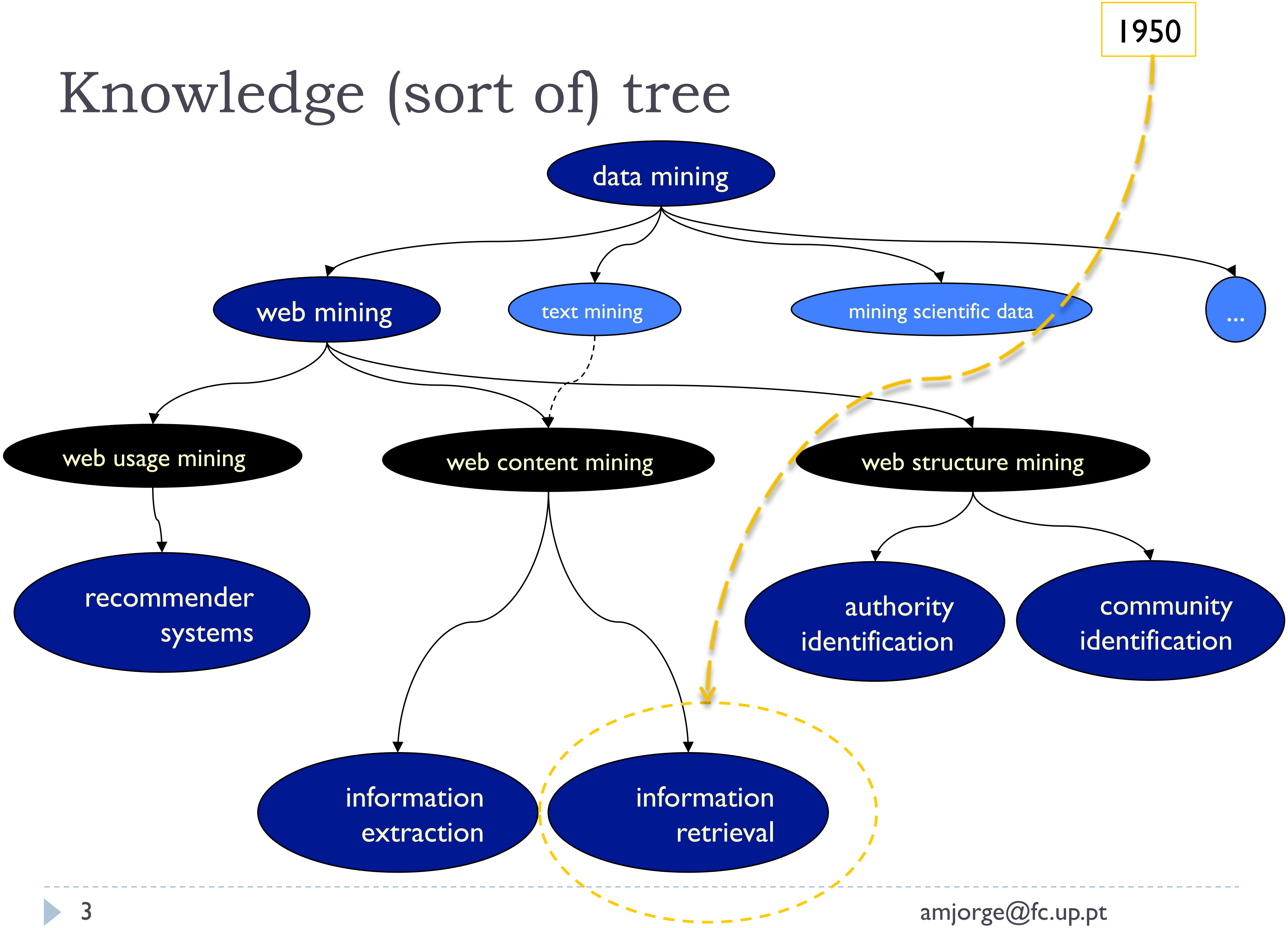
Information Retrieval

Alípio Jorge, DCC-FC, Universidade do Porto
amjorge@fc.up.pt

Overview

- Information Retrieval
  - basic concepts
  - models
  - relevance feedback
  - evaluation measures

Knowledge (sort of) tree



Introduction

- Information retrieval
  - helping users find information that matches their needs
    - From **repositories/sources** with documents or similar
      - □ Web, document database

  - acquisition
  - organization
  - storage
  - retrieval
  - distribution

of information

- Classical IR: document retrieval

General architecture

- user poses a **query** (e.g. "turing")
- the query is sent to the **retrieval system**
- which uses the **document index**
- to get **docs** with query terms
- compute **relevance** of documents
- **rank** results

How documents are represented

Information Retrieval Models

- How are documents and queries best represented?
  - Reflecting user's intent
  - Reflecting document content
  - Computationally appealing

- Query "data mining"
  - "mining is important for finding gold"
  - "classification and regression are data mining"
  - "economical data is missing"
  - "data mining is important for marketing"
  - "new release of the R statistical package"

Information Retrieval Models

- Bag of words
  - Each document is seen as the set of its terms
  - A term is not necessarily a word
  - Weights can be associated to terms
  - Different models find weights in different ways

- "mining is important for finding gold“

```
> doc<-"mining is important for finding gold"
> bagofwords<-unlist(strsplit(doc," "))
> bagofwords
[1] "mining"    "is"        "important" "for"       "finding"   "gold"
> q<-"mining"
> is.element(q,bagofwords)
[1] TRUE
```

Information Retrieval Models

- Vector representation

$$\mathbf{d}_j = (w_{1j}, w_{2j}, \ldots, w_{|V|j})$$

- Models
  - Boolean Model
    - binary
  - Vector Space Model
    - Uses frequencies
    - tf-idf
  - Statistical Language Model
    - Uses Bayesian reasoning

Information Retrieval Models

- Boolean model
  - Document representation
    - Each weight is one or zero (True or False)

```
> docs<-c("mining is important for finding gold",
+ "classification and regression are data mining")

> bagofwords<-function(x) unlist(strsplit(x," "))

> vocab<-union(bagofwords(docs[1]),bagofwords(docs[2]))
> vocab
 [1] "mining"         "is"             "important"      "for"
 [5] "finding"        "gold"           "classification" "and"
 [9] "regression"     "are"            "data"
```

Information Retrieval Models

- Boolean model
  - Document representation
    - Each weight is one or zero (True or False)

```
> booleanvec<-function(doc,vocab)
+                  sapply(vocab,function(x) is.element(x,bagofwords(doc)))
> booleanvec(docs[1],vocab)
        mining             is      important            for        finding
          TRUE           TRUE           TRUE           TRUE           TRUE
          gold classification            and     regression            are
          TRUE          FALSE          FALSE          FALSE          FALSE
          data
         FALSE
> as.numeric(booleanvec(docs[1],vocab))
 [1] 1 1 1 1 1 1 0 0 0 0 0
```

Information Retrieval Models

- Boolean model
  - Boolean queries
    - (Data AND mining) OR classification
  - Retrieval
    - Exact match
    - No ranking
  - Seldom used alone in practice
    - Cf "data +mining -gold"

Information Retrieval Models

- Vector Space Model
  - Weights are any number
  - Weights reflect importance of term in document

- Term frequency scheme:
  - Weight(t,doc) = frequency of t in doc
  - May be normalized

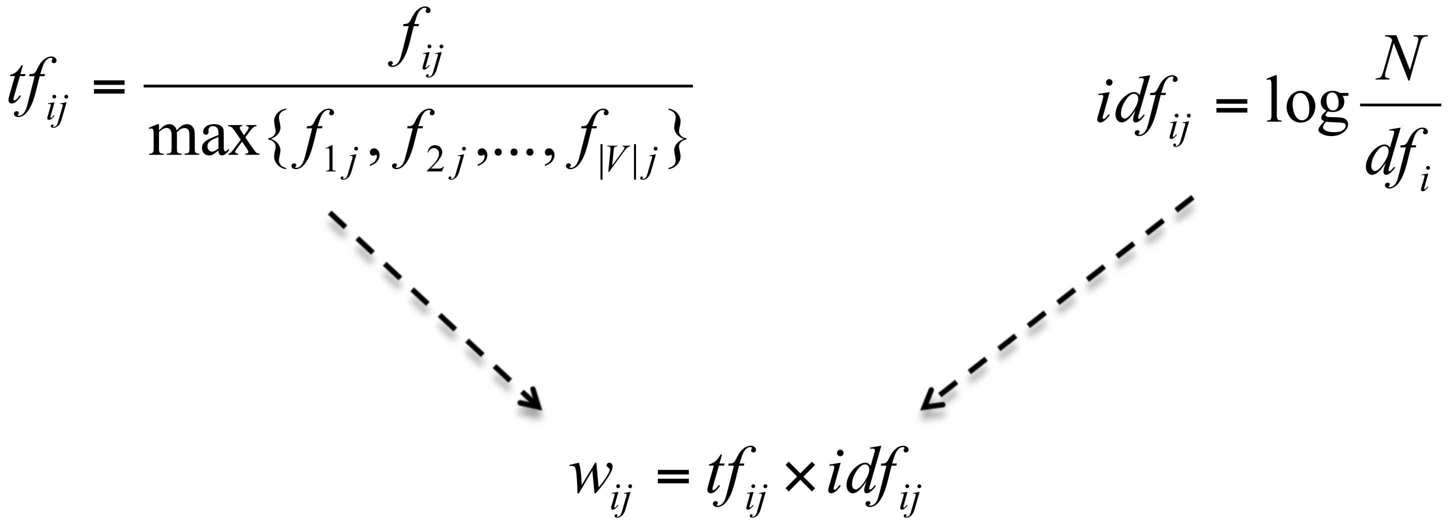
```
> doc<-"data mining deals with data"
> table(bagofwords(doc))

  data   deals mining   with
     2       1      1      1
```

  - Problem: popular terms have high frequencies in many docs

Information Retrieval Models

- TF-IDF scheme:
  - Term frequency / inverse document frequency
  - Weight(t,doc) values freq in doc but also **discriminant** power
    - Grows with term frequency in the doc
    - Decreases with number of docs where term appears



$$tf_{ij} = \frac{f_{ij}}{\max\{f_{1j}, f_{2j}, \ldots, f_{|V|j}\}}$$

$$idf_{ij} = \log \frac{N}{df_i}$$

$$w_{ij} = tf_{ij} \times idf_{ij}$$

Information Retrieval Models

- TF-IDF scheme:
  - Docs
    - "mining is important for finding gold“
    - "classification and regression are data mining"
    - "data mining deals with data"

$$tf_{ij} = \frac{f_{ij}}{\max\{f_{1j}, f_{2j}, \ldots, f_{|V|j}\}}$$

$$idf_{ij} = \log \frac{N}{df_i}$$

$$w_{ij} = tf_{ij} \times idf_{ij}$$

  - TfIdf(classification,d2)
  - TfIdf(data,d3)
  - TfIdf(mining,d3)

Information Retrieval Models

- Query representation:
  - Same way as documents
  - Or modifying the TF part

$$tf_{iq} = 0.5 + \frac{0.5 f_{iq}}{\max\{f_{1q}, f_{2q}, \ldots, f_{|V|q}\}}$$

  - Example: query "data mining"

Information Retrieval Models

- Document retrieval:
  - Relevant docs are the ones closer to the query
  - Similarity metric
    - e.g. *cosine*

$$\cos(\mathbf{d}_j, \mathbf{q}) = \frac{\mathbf{d}_j \otimes \mathbf{q}}{\left\|\mathbf{d}_j\right\| \times \left\|\mathbf{q}\right\|}$$

$$\cos(\mathbf{d}_j, \mathbf{q}) = \frac{\sum_{i=1}^{|V|} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{|V|} w_{ij}{}^2} \times \sqrt{\sum_{i=1}^{|V|} w_{iq}{}^2}}$$

Package tm

R package tm

```
> library(tm)
> docs<-c("mining is important for finding gold","classification and
regression are data mining","data mining deals with data")
> corpus<-Corpus(VectorSource(docs))
> dtm <- DocumentTermMatrix(corpus)

# Term frequency - inverse document frequency Scheme
> tfidf<-weightTfIdf(dtm)
> as.matrix(tfidf)
    Terms
Docs       and       are classification       data     deals   finding
   1 0.0000000 0.0000000      0.0000000 0.00000000 0.0000000 0.3169925
   2 0.2641604 0.2641604      0.2641604 0.09749375 0.0000000 0.0000000
   3 0.0000000 0.0000000      0.0000000 0.23398500 0.3169925 0.0000000
    Terms
Docs       for      gold important mining regression      with
   1 0.3169925 0.3169925 0.3169925      0  0.0000000 0.0000000
   2 0.0000000 0.0000000 0.0000000      0  0.2641604 0.0000000
   3 0.0000000 0.0000000 0.0000000      0  0.0000000 0.3169925
```

R package tm

```
# Term frequency Scheme
> tf<-weightTf(dtm)
> as.matrix(tf)
    Terms
Docs and are classification data deals finding for gold important mining
   1   0   0              0    0     0       1   1    1         1      1
   2   1   1              1    1     0       0   0    0         0      1
   3   0   0              0    2     1       0   0    0         0      1
    Terms
Docs regression with
   1          0    0
   2          1    0
   3          0    1
```

R package tm

```
# Boolean model
> boolean<-weightBin(dtm)
> as.matrix(boolean)
    Terms
Docs and are classification data deals finding for gold important mining
   1   0   0              0    0     0       1   1    1         1      1
   2   1   1              1    1     0       0   0    0         0      1
   3   0   0              0    1     1       0   0    0         0      1
    Terms
Docs regression with
   1          0    0
   2          1    0
   3          0    1
```

R package tm

```
"mining is important for finding gold‶
"classification and regression are data mining"
"data mining deals with data"
```

```
# Measuring dissimilarity between documents
> mycosdist<-function(x,y) 1-x%*%y/(sqrt(x%*%x)*sqrt(y%*%y))
> proxy::dist(as.matrix(dtm),method=mycosdist)
          1         2
2 0.8174258
3 0.8309691 0.5370900
> proxy::dist(as.matrix(weightBin(dtm)),method=mycosdist)
          1         2
2 0.8174258
3 0.7763932 0.5917517
> proxy::dist(as.matrix(weightTf(dtm)),method=mycosdist)
          1         2
2 0.8174258
3 0.8309691 0.5370900
> proxy::dist(as.matrix(weightTfIdf(dtm)),method=mycosdist)
          1         2
2 1.0000000
3 1.0000000 0.9160317
```

R package tm

```
"mining is important for finding gold‟
"classification and regression are data mining"
"data mining deals with data"
```

```
# Ranking docs given a query
> cq<-Corpus(VectorSource("data mining"))
> dtmq<-DocumentTermMatrix(cq)
> qv<-NULL
> qv[colnames(dtm)]<-0
> qv[colnames(dtmq)]<-1
> dtmqd<-rbind(as.matrix(dtm),qv)
> proxy::dist(dtmqd,method=mycosdist)
            1         2         3
2   0.8174258
3   0.8309691 0.5370900
qv  0.6837722 0.4226497 0.1982163
```

R package tm

```
"mining is important for finding gold‶
"classification and regression are data mining"
"data mining deals with data"
```

```
# Ranking docs given a query using Tf-Idf scheme
> dict<-colnames(dtm) # fetch the vocabulary of the corpus
> dtmq<-DocumentTermMatrix(Corpus(VectorSource("data
mining")),list(dictionary=dict))
> tfidfq<-as.matrix(dtmq)[1,][dict]*log(3/apply(as.matrix(weightBin(dtm)),
2,sum))[dict]
> tfidfd<-as.matrix(weightTfIdf(dtm))
> mycosdist(tfidfq,tfidfd[1,])
     [,1]
[1,]    1
> mycosdist(tfidfq,tfidfd[2,])
          [,1]
[1,] 0.8185288
> mycosdist(tfidfq,tfidfd[3,])
          [,1]
[1,] 0.5372911
```

Information Retrieval Models

- Statistical Language Model
  - Rank documents by the likelihood of the query
  - Prob(doc | query)

$$\Pr(d_j \mid q) = \frac{\Pr(q \mid d_j)\Pr(d_j)}{\Pr(q)}$$

Pr(dj) can be uniform
Pr(q) does not affect rank

$$\Pr(q = q_1 q_2 \ldots q_n \mid d_j) = \prod_{i=1}^{m} \Pr(q_i \mid d_j) = \prod_{i=1}^{|V|} \Pr(t_i \mid d_j)^{f_{iq}}$$

$$\Pr(t_i \mid d_j) = \frac{f_{ij}}{|d_j|}$$

Information Retrieval Models: activity

- Consider
  - d1 = “data mining is cool”
  - d2 = “coal mining is hot”
  - q = “coal mining”

- Rank d1 and d2
  - Pr(d1|q)
  - Pr(d2|q)

Information Retrieval Models

- Statistical Language Model
  - If term is not in doc → Pr(d|q) = 0
  - This is too drastic and degrades retrieval
    - Needs smoothing
  - Example
    - d1 = “data mining is cool”
    - d2 = “coal mining is hot”
    - d3 = “classification is mining”
    - q = “data mining classification”
  - All docs would have score zero

Information Retrieval Models

- Statistical Language Model
  - Smoothing
    - Prevent zero probability
    - Assign a non-zero residual probability to unobserved events
    - At the cost of higher probabilities

$$\Pr_{smoothed}(t_i \mid d_j) = \frac{\lambda + f_{ij}}{\lambda |V| + |d_j|}$$

- If λ=1 we have Laplace smoothing

Information Retrieval Models

- Statistical Language Model: smoothing
  - Example
    - d1 = “data mining is cool”
    - d2 = “coal mining is hot”
    - d3 = “classification is mining”
    - q = “data mining classification”

$$\Pr_{smoothed}(t_i \mid d_j) = \frac{\lambda + f_{ij}}{\lambda|V| + |d_j|}$$

    - λ=1
      - □ Pr(di | q) ?
        - □ Pr(data | d2) = (1+0)/(1x7+4) = 1/11

Relevance Feedback

- improve retrieval effectiveness
  - user labels retrieved documents
    - relevant / not relevant
  - system expands query with terms from relevant docs
  - or
  - system produces a classification model

  - … may iterate

Relevance Feedback

- The Rocchio Method
  - q is the original query
  - D_r is the set of relevant documents selected by the user
  - D_{ir} is the set of irrelevant documents
  - the expanded query q_e is:

$$\mathbf{q}_e = \alpha\mathbf{q} + \frac{\beta}{|D_r|}\sum_{\mathbf{d}_r \in D_r} \mathbf{d}_r - \frac{\gamma}{|D_{ir}|}\sum_{\mathbf{d}_{ir} \in D_{ir}} \mathbf{d}_{ir}$$

Relevance Feedback: activity

- Example (boolean model)
- **q**={data mining}
- D_i
  - "mining is important for finding gold"
  - "classification and regression are data mining"
  - "economical data is missing"
  - "data mining is important for marketing"
- parameters= 0.4, 0.3, 0.3
- w(data) = 0.4 + 0.3*2/2 - 0.3/2
- w(classification) = 0+0.3/2-0
- w(important) = 0

$$\mathbf{q}_e = \alpha \mathbf{q} + \frac{\beta}{|D_r|} \sum_{\mathbf{d}_r \in D_r} \mathbf{d}_r - \frac{\gamma}{|D_{ir}|} \sum_{\mathbf{d}_{ir} \in D_{ir}} \mathbf{d}_{ir}$$

Relevance Feedback

- Machine learning (classification)
  - classes are relevant and irrelevant
  - cases are documents (e.g. tf-idf vectors)
  - apply SVM or naive Bayes

- even simpler:
  - find a prototype vector for each class
    - adapting Rocchio

$$c_r = \frac{\beta}{|D_r|} \sum_{\mathbf{d} \in D_r} \frac{\mathbf{d}}{\|\mathbf{d}\|} - \frac{\gamma}{|D_{ir}|} \sum_{\mathbf{d} \in D_{ir}} \frac{\mathbf{d}}{\|\mathbf{d}\|}$$

  - use cosine distance to assign new docs to classes

Evaluation Measures

- How to evaluate the results of IR ?
  - relevant doc retrieved / not retrieved
  - irrelevant docs retrieved
  - how high/low in the rank?

- Recall at position i
- Precision at position i
- Average Precision
- Precision-Recall curve

Rank i	+/-	
1	+	*rel*
2	+	
3	-	*irrel*
4	+	
5	-	
6	-	
7	+	
8	-	

Evaluation Measures

- Recall for i documents retrieved
  - The fraction of relevant (+) docs in the first i retrieved wrt the total number of docs relevant to q

$$r(i) = \frac{rel_i}{|D_q|}$$

- Precision for i documents retrieved
  - Retrieved relevants wrt total retrieved

$$p(i) = \frac{rel_i}{i}$$

Rank i	+/-
1	+
2	+
3	-
4	+
5	-
6	-
7	+
8	-

Evaluation Measures: activity

- Examples (suppose 4 docs are relevant)
  - p(2)
  - p(4)
  - p(8)
  - r(2)
  - r(4)
  - r(8)

$$r(i) = \frac{rel_i}{|D_q|}$$

$$p(i) = \frac{rel_i}{i}$$

Rank i	+/-
1	+
2	+
3	-
4	+
5	-
6	-
7	+
8	-

Evaluation Measures

- Average Precision
  - wrt to all retrieved docs
  - provides a single number that summarizes results
    - *d↓i↑q* is the i-th relevant retrieved doc

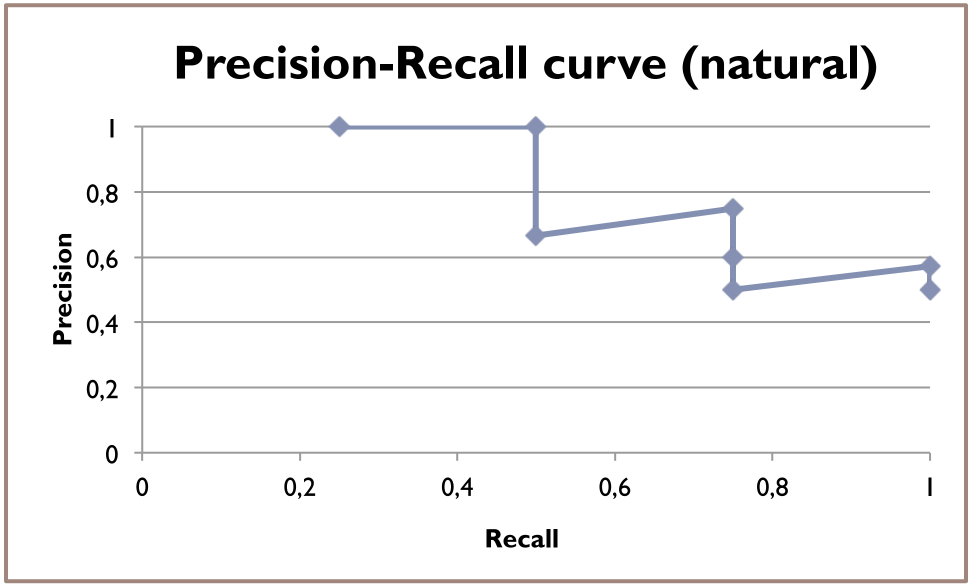
$$p_{avg} = \frac{\sum_{d_i^q \in D_q} p(i)}{\left|D_q\right|}$$

- Example
  - p_avg = (1+1+.75+4/7) / 4

Rank i	+/-
1	+
2	+
3	-
4	+
5	-
6	-
7	+
8	-

Evaluation Measures

- Precision-Recall curve
  - plot each document as a point (p(i),r(i))
    - recall is the x-axis, precision is the y-axis
  - or plot at each "round" value of recall
    - interpolate precision if necessary



Rank i	+/-
1	+
2	+
3	-
4	+
5	-
6	-
7	+
8	-

Evaluation Measures

- Comparing algorithms
  - overlay precision-recall curves

- Precision and recall interplay
  - usually there is a tradeoff

- Recall in practice
  - D_q is not known
  - precision is critical

- F-score
  - Combining precision and recall = 2.P.R/(P+R)

Resources

- Books
  - Web Data Mining, Bing Liu, Springer, 2007
  - Mining the World Wide Web, Chang, G., Healey, M., McHugh, J., Wang, J., Kluwer Academic Press, 2001.
  - Modern Information Retrieval, Ricardo Baeza-Yates and Berthier Ribeiro-Neto