Utilizing Machine Learning in Information Retrieval:

- Text Classification
- Learning to Rank

(COSC 488)

Nazli Goharian
nazli@cs.georgetown.edu

Literatures used to prepare the slides: See last page!

What is Text Classification?

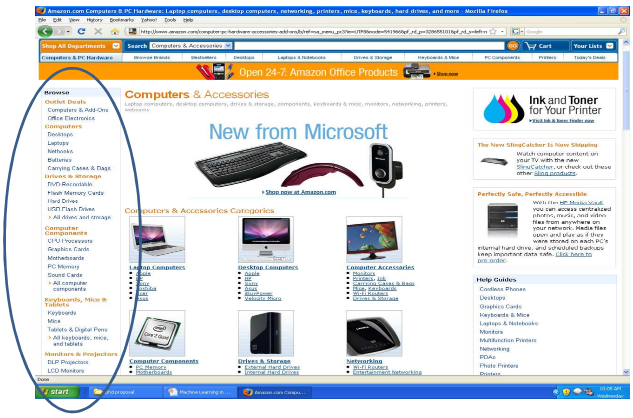
Text classification also known as *text categorization*, *topic classification*, or *topic spotting* is the process of assigning predefined categor(ies)/topic(s)/class(e)s/label(s) to a document that reflect its overall contents.

Application of Text Classification

- News Classification
  - *"Politics", "Sports", "Business"*

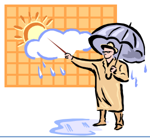
Application of Text Classification

- Shopping Products Classification
  - *"Electronics", "Home Appliances", "Books"*



Application of Text Classification

- News Routing/Filtering

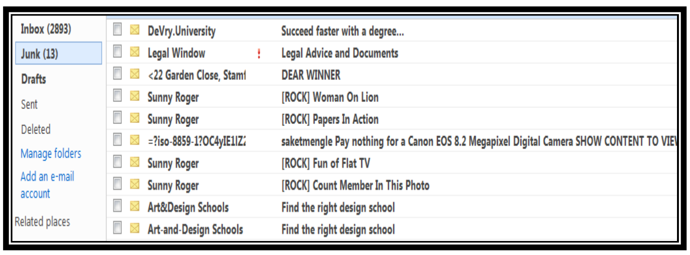


Tropical storms are building up in the south Pacific due to high pressure belts. The rains may continue for few more days.

Users interested in weather news *(standing queries)*

Application of Text Classification

- Spam Filtering
  - *"Spam", "Not Spam"*



Improving Search Results via Text Classification

- Query is searched in the user selected categories in web directories
- Categorized result set is presented to user
- Learning to rank -- (more recent efforts)
  Using various document features such as document length, age, etc. and their relevance to a query, build a model to rank/re-rank the documents
- Query category is searched against categorized pages (vertical search, advertisement search,…)

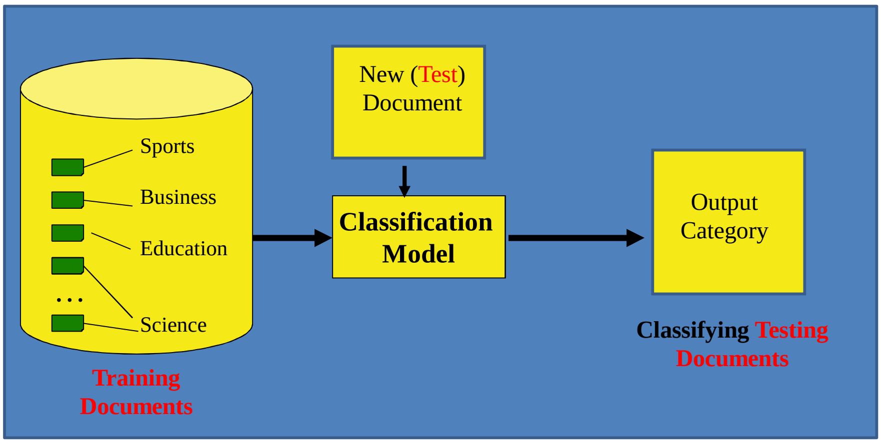
Web Directories

Constructing Web directories to be able to browse information via predefined set of categories:

- Yahoo
- dmoz Open Directory Project (ODP)
- Existing directories are based on human efforts
  - 80,000 editors involved to maintain ODP; www.dmoz.org

Using Web directories (Yahoo,ODP, Wikipedia,…) as training data, the classifier classifies new web pages into categories

Supervised Learning (Classification)



Supervised Learning (Classification)

- Learning a model (classifier), using annotated training samples (documents) to classify any new incoming document into pre-defined set of topics
- Each Training document has one/more label(s)
- Various learning algorithms exists, examples:
  - Example: *Naïve Bayes, decision tree, support vector machine, neural network, regression, K-nearest neighbor,…*
- Model/Classifier is used to classify incoming (test) documents

Example: Single-labeled Document

The Dow Jones industrial average lost 26 points, or 0.3%. The S&P 500 index fell 6 points, or 0.6%. The Nasdaq composite was little changed. Stocks slipped through most of the session as investors mulled the implications of a weaker-than-expected reading on the services sector of the economy, and mixed reports on the jobs market, ahead of Friday's big monthly payrolls report.

Source: CNN (http://money.cnn.com/2010/02/03/markets/markets_newyork/index.htm?postversion=2010020318)

- Politics
- Business
- Sports
- Entertainment

Example: Multi-labeled Document

President Obama, in his proposed 2011 budget, is calling on Congress to make a number of tax changes for individuals. Some ideas are new. Many others were made last year, but not enacted by Congress. So the estimates of the revenue that may be raised by his proposals may be overly optimistic.

Source: CNN (http://money.cnn.com/2010/02/01/pf/taxes/obama_budget_tax_changes/index.htm)

- Politics
- Business
- Sports
- Entertainment

Hard Categorization vs. Ranking Categorization

Hard Categorization

Complete decision of True or False for each pair $< d_j, c_i >$

Document	Category Assigned
d_1	c_1, c_2
d_2	c_2
d_3	c_3, c_4
d_4	c_4

Ranking (Soft) Categorization

Given $d_j \in D$, rank the categories according to their estimated appropriateness to d_j

Document	Category	Estimated appropriateness
d_1	c_2	0.6
	c_1	0.3
	c_3	0.05
	c_4	0.05

Types of Classification

from: X. Qi and B. Davison , ACM Computing Surveys, 2009

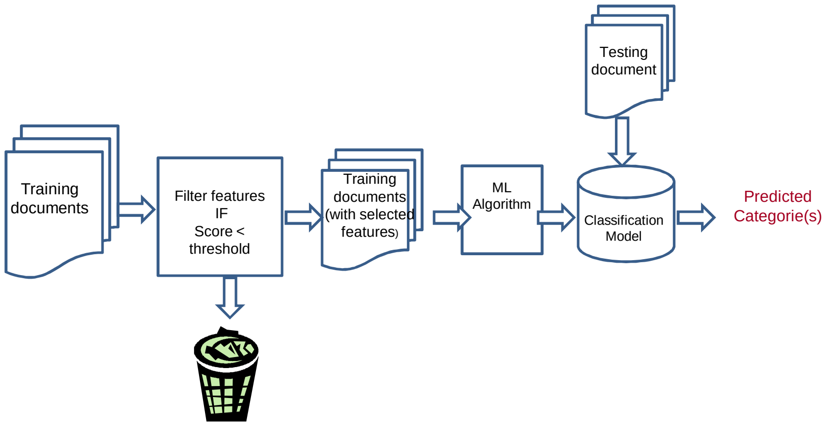
Binary → Classifier → *Class C* OR *Class* $\bar{C}$

Hard, Multi-class, Single-label → Classifier → *Class* C_1 OR *Class* C_2 OR ….*Class* C_n

Hard, Multi-class, Multi-label → Classifier → *Class* C_1 and *Class* C_3

Soft, Multi-class → Classifier → C_1: 0.2, C_2: 0.2, …………C_n: 0.6,

Text Classification Process

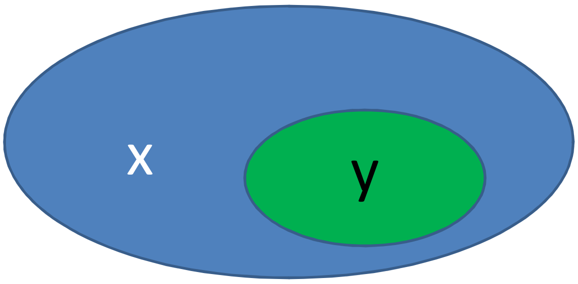


Feature Selection

- Feature Selection in text classification refers to selecting a subset of the collection terms and utilize them in the process of text classification.
- Good features are better indicators of a class label
- Feature reduction tends to:
  - Reduce *overfitting*
  - Improve performance due to reducing dimensionality
- Feature Extraction provides more detailed features and feature relationships ***(not covered in this course)***

Feature Selection

- Given a feature set $X=\{x_i \mid i=1\ldots N\}$, find a subset $Y=\{x_{i1}, x_{i2}, \ldots, x_{iM}\}$, with $M<N$, that increases the probability of correct classification



Text Features

- Feature space in text may include:
  - Lexical features (words, phrases)
  - Part-of-Speech (POS)
  - N-grams
  - Synonyms
  - ….
- General feature types may be:
  - Numeric
  - Nominal
  - Ordinal
  - Ratio

Web Page Features

- Additional features are utilized in Web page classification task:
  - On-Page Features
  - Neighboring Page Features (External Links)

On-Page Features

HTML tags:

- title
- headings
- metadata
- main text

HTML tags usually removed in pre-processing; the content of tags preserved

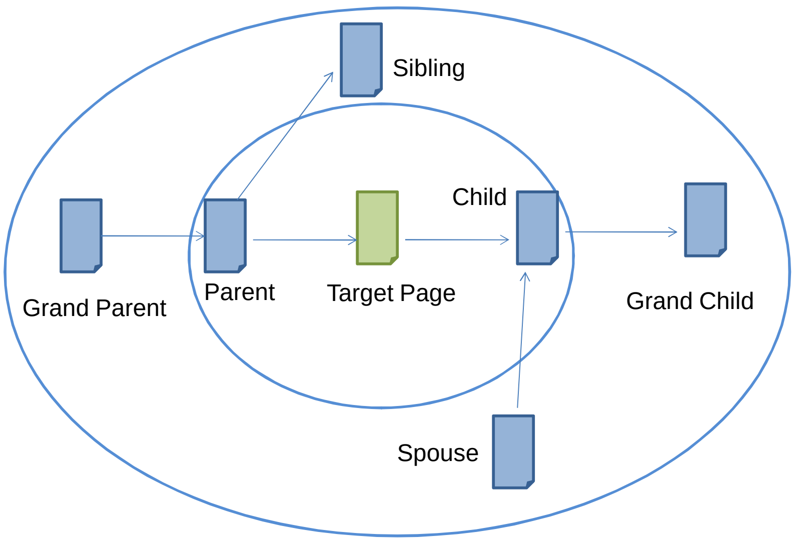
URL – classify without using page content

Neighboring-Page Features

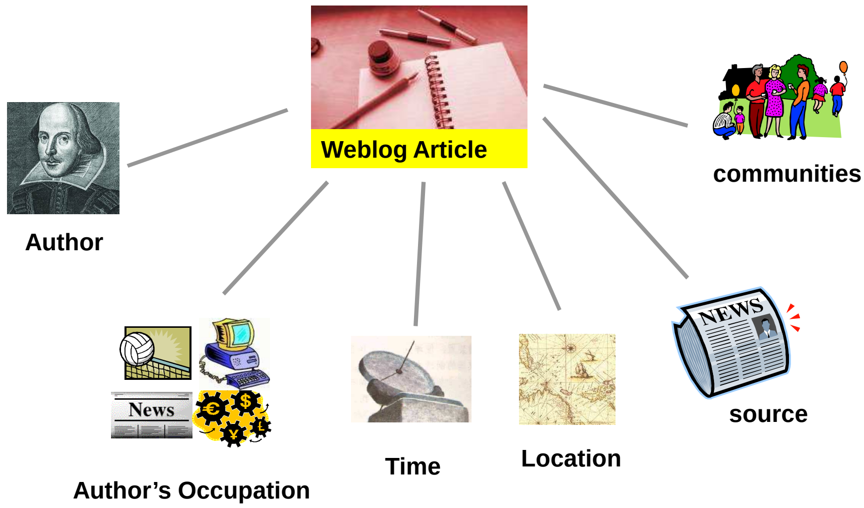
- Neighbors (linked pages) have similar topics and categories
- Number of steps from a page --shown as 2 (parent, child, sibling, grand parent, grand child); more steps more expensive & less effective
- Although all useful, but sibling is shown to be more effective
- Using only portion of neighboring content: title, anchor text, text closer to hyperlink to train a classifier
- Voting -- majority class of neighbors used

Neighboring-Page Features

from: X. Qi and B. Davison , ACM Computing Surveys, 2009



Context Features of a Document



Slide from: Cheng Xiang Zhai, keynote, SIGIR, 2011

Feature Selection Algorithms

- Frequency based FS:
  - df
  - tf-idf
  - Tf-icf
- Commonly used Information Theoretic based FS:
  - Mutual Information
  - Information Gain
  - χ^2 Statistic (CHI)
  - Odds Ratio

    (Note: There are some more FS algorithms!)

Feature Selection: Frequency based

- DF (Document Frequency): *Frequency of a term in the collection*
  - Retain terms that are not *stop terms* (high *df*) and do not have very low *df* (noise, not of interest)
- TF-IDF

  tf: frequency of a term in a document -- *commonly normalized*

  idf: inverse document frequency $tfidf(t_k, d_i) = TF(t_k, d_i) * \log\left(\frac{|D|}{df(t_k)}\right)$
  - Retain terms with high *tf-idf* in a document
- TF-ICF
  - Analogous to *tf-idf* but considering the frequency of term in the category. $tficf(t_k, c_i) = TF(t_k, c_i) * \log\left(\frac{|C|}{cf(t_k)}\right)$

Feature Selection (FS)

Consider the Term-Class incidence table:

Case	Docs in class: c_p	Docs not in class: $\overline{c_p}$	Total
Docs that contain term k_i	$n_{i,p}$	$n_i - n_{i,p}$	n_i
Docs that do not contain term $\overline{k_i}$	$n_p - n_{i,p}$	$N_t - n_i - (n_p - n_{i,p})$	$N_t - n_i$
All docs	n_p	$N_t - n_p$	N_t

The notations used in this table are used in the FS algorithms of the next few pages!

From: *Modern Information retrieval, R. Baeza-Yates & B. Ribeiro-Neto, 2011*

FS: Mutual Information (MI)

Measuring the amount of information the presence of a term contributes to the classification

MI between term k_i and set of classes C is expressed as expected value of:

$$I(k_i, c_p) = \log \frac{P(k_i, c_p)}{p(k_i)P(c_p)} = \log\left(\frac{\frac{n_{i,p}}{N_t}}{\frac{n_i}{N_t}\cdot\frac{n_p}{N_t}}\right)$$

Two alternates: 1) across all classes; 2) maximum term information:

$$MI(k_i, C) = \sum_{p=1}^{L} p(c_p)\, I(k_i, c_p) = \sum_{p=1}^{L} \frac{n_p}{N_t} \log\left(\frac{\frac{n_{i,p}}{N_t}}{\frac{n_i}{N_t}\cdot\frac{n_p}{N_t}}\right)$$

$$I_{\max}(k_i, C) = \max_{p=1}^{L}\ I(k_i, c_p) = \max_{p=1}^{L}\ \log\left(\frac{\frac{n_{i,p}}{N_t}}{\frac{n_i}{N_t}\cdot\frac{n_p}{N_t}}\right)$$

FS: Information Gain (IG)

Measuring the amount of information both the presence and the absence of a term contribute to the classification.

Terms with *IG >= threshold* are kept.

$$IG(k_i, C) = -\sum_{p=1}^{L} P(c_p)\log P(c_p)$$
$$-(-\sum_{p=1}^{L} P(c_p, k_i)\log P(c_p \mid k_i))$$
$$-(-\sum_{p=1}^{L} P(c_p, \bar{k}_i)\log P(c_p \mid \bar{k}_i))$$

$$IC(k_i, C) = -\sum_{p=1}^{L}\left((\frac{n_p}{N_t}\log\frac{n_p}{N_t}) - (\frac{n_{i,p}}{N_t}\log\frac{n_{i,p}}{n_i}) - (\frac{n_p - n_{i,p}}{N_t}\log\frac{n_p - n_{i,p}}{N_t - n_i}) \right)$$

FS: Chi Square (χ^2)

- Chi Square measures the *dependency* between the term and the class (*value of zero indicates independency*)

$$\chi^2(k_i, c_p) = \frac{N_t\left(P(k_i, c_p)P(\bar{k}_i, \bar{c}_p) - P(k_i, \bar{c}_p)P(\bar{k}_i, c_p)\right)^2}{P(c_p)P(\bar{c}_p)P(k_i)P(\bar{k}_i)}$$

- Calculate χ^2 of a term over all categories and retain the term if the value meets a threshold. Two alternatives:

1) Averaging over all categories: $\chi^2_{avg}(k_i) = \sum_{p=1}^{L} P(c_p)\chi^2(k_i, c_p)$

2) Considering the largest value: $\chi^2_{\max}(k_i) = \max_{p=1}^{L} \chi^2(k_i, c_p)$

FS: Chi Square (χ^2) (Cont'd)

- Chi Square measures the *dependency* between the term and the class (*value of zero indicates independency*)

$$\chi^2(k_i, c_p) = \frac{N_t\left(P(k_i, c_p)P(\bar{k}_i, \bar{c}_p) - P(k_i, \bar{c}_p)P(\bar{k}_i, c_p)\right)^2}{P(c_p)P(\bar{c}_p)P(k_i)P(\bar{k}_i)}$$

$$\chi^2(k_i, c_p) = \frac{N_t\left(n_{i,p}\left(N_t - n_i - n_p + n_{i,p}\right) - \left(n_i - n_{i,p}\right)\left(n_p - n_{i,p}\right)\right)^2}{n_p\left(N_t - n_p\right)n_i\left(N_t - n_i\right)}$$

$$= \frac{N_t\left(n_{i,p}N_t - n_p n_i\right)^2}{n_p n_i\left(N_t - n_p\right)\left(N_t - n_i\right)}$$

FS: Odds Ratio

- Odds Ratio reflects the odds of the word occurring in the positive class normalized by that of the negative class.
- Odds Ratio for a term t_k in category c_i

$$OR(t_k, c_i) = \frac{P(t_k \mid c_i).[1 - P(t_k \mid \overline{c_i})]}{P(\overline{t_k} \mid \overline{c_i}).[1 - P(\overline{t_k} \mid c_i)]}$$

Supervised Learning Algorithms

- Naïve Bayes
- K-Nearest Neighbor (KNN)

Only these two are covered in this course!

- Support Vector Machines (SVM)
- Decision-tree
- Decision-Rule classifiers
- Neural Networks
- Rocchio
- HMM
- CRF

Representation of Text

This week, the United Nations created the position of czar in the global fight against a possible avian influenza pandemic. Meanwhile, officials here in the United States acknowledged the country is unprepared if this never-before-seen strain of flu, known to scientists as H5N1 virus, were to hit this winter.

- Commonly used pre-processing: stop word removal, stemming,…

d1:<*week, united, nations, create, position, czar, global, fight, against, possible,…..*>

Term	Frequency
Week	1
united	2
nation	1
………	

Phrases:
United nations
Avian influenza
………..

Naïve Bayes Text Classifier

- Text as "bag-of-words"
- Independent assumption -- *occurrence of terms and their positions*
- Building Model:
  - For each category c_i build a probabilistic model

T: text in class c_i
n: size of the vocabulary

$$P(T: t_1, t_2, \ldots. t_n \mid c_i)$$

  - Calculate the prior probability $P(C_i)$

Naïve Bayes Text Classifier

- Classify Text:
  - Calculate probability of each category for a given text

$$P(c_i \mid d_j) = p(c_i)P(d_j \mid c_i)$$

  - The *category* c_i with the highest score among all categories C is the one that is most probable to generate the text d_j

$$c_{\max\ a\ posteriori} = \arg\max_{c_i \in C}\ p(c_i)P(d_j \mid c_i)$$

Naïve Bayes Text Classifier

$$P(c_i \mid d_j) = p(c_i)\ \underbrace{P(d_j \mid c_i)}$$

$$\prod_{k=1}^{|T|} P(t_{kj} \mid c_i) = \sum_{i=1}^{|T|} \log\ P(t_{kj} \mid c_i)$$

Naïve Bayes Text Classifier

- Need to estimate the probability: $P(t_{kj} \mid c_i)$
  - Multinomial model:

$$\frac{\textit{number of times term } t_{kj} \textit{ appears in category } c_i + 0.5}{\textit{total terms in } c_i + 1}$$

  - binomial or Bernoulli model:

$$\frac{\textit{number of documents in category } c_i \textit{ that term } t_{kj} \textit{ appears}}{\textit{total documents in } c_i}$$

Naïve Bayes Text Classifier

Multinomial model:

$$P(c_i \mid d_j) = \underbrace{p(c_i)}\ \underbrace{P(\vec{d}_j \mid c_i)}$$

$$\log\left(\frac{\textit{docs in } c_i}{\textit{total docs}}\right)$$

$$\prod_{k=1}^{|T|} P(t_{kj} \mid c_i) = \sum_{i=1}^{|T|} \log\ \underbrace{P(t_{kj} \mid c_i)}_{\frac{\textit{number of times term } t_{kj} \textit{ appears in category } c_i + 0.5}{\textit{total terms in } c_i + 1}}$$

To avoid a zero if a new term appears → Smoothing

- Various approaches: *Dirchelet prior, Laplace,..*

Example

Doc-1
Category: Computers
The sales of laptops in 2009 was high as many OS were released

Doc-2
Category: Computers
Many OS provide varying level of securities for laptops as they tend to switch networks. This makes the laptops more secure from computer viruses

Doc-3
Category: Epidemic
A new virus called H1N1 causes Swine Flu.

Doc-4
Category: Epidemic
Bird flu is caused by a virus called H5N1. The disease is of concern to humans, who have no immunity against it.

Example

Assume that red terms are the selected features:

Doc-1
Category: Computers
The sales of laptops in 2009 was high as many OS were released

Doc-2
Category: Computers
Many OS provide varying level of securities for laptops as they tend to switch networks. This makes the laptops more secure from computer viruses

Doc-3
Category: Epidemic
A new virus called H1N1 causes Swine Flu.

Doc-4
Category: Epidemic
Bird flu is caused by a virus called H5N1. The disease is of concern to humans, who have no immunity against it.

Example: Naïve Bayes Text Classifier

Task: Classify D5: *"A deadly virus called H1N1 was detected in various parts of the world"*

- **P(Computers|D5)** = P(Computers) P(Virus|Computers) P(H1N1|Computers)
- **P(Epidemic|D5)** = P(Epidemic) P(Virus|Epidemic) P(H1N1|Epidemic)

P(Epidemic|D5) > P(Computers|D5)

Thus, class of D5 is Epidemics

Vector Space Classification

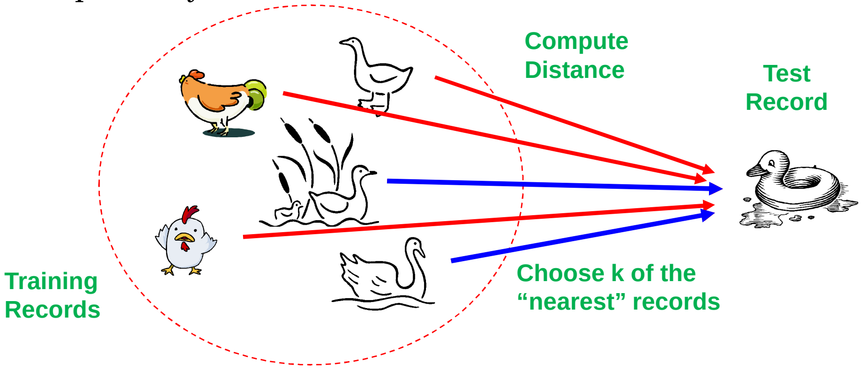
- Documents represented as a vector with generally *tfidf* of terms
- Generally classification decisions are based on a similarity/distance measure
  - Centroids [averages] play a role
- Sample algorithms:
  - Rocchio
  - K Nearest Neighbor (kNN)
  - SVM

Only this one is covered in this course!

Nearest Neighbor Classifiers

***Slide from**: Tan, Steinback, Kumar, 2004*

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



K-Nearest Neighbor Classifier

- No model is built (lazy learner) a priori
  (Classification done based on raw training data)
- The class of a document will be the class of the majority class of the *k* nearest neighbor (*majority voting*)
- The relatedness/nearness of two documents can be quantified in terms of *similarity* (eg. *Cosine measure*) or *distance (eg. Euclidean distance*)
  - Different weight for different features
  - Feature values can be normalized to prevent different handling (may prefer different handling!)
- Sensitivity to value of *K*
  - Picked empirically, domain knowledge

Distance/Similarity Measures

Euclidean Distance:

$$dist(d_i, d_j) = \sqrt{(|d_{i1} - d_{j1}|^2 + |d_{i2} - d_{j2}|^2 + \ldots + |d_{ip} - d_{jp}|^2)}$$

Cosine Similarity:

$$Sim(d_i, d_j) = \frac{\sum_{k=1}^{t} d_{ik} \, x \, d_{jk}}{\sqrt{\sum_{k=1}^{t} (d_{ik})^2 \sum_{k=1}^{t} (d_{jk})^2}}$$

Term weight:

$$w_{ij} = \frac{(\log tf_{ij} + 1.0) * idf_j}{\sum_{j=1}^{t} [(\log tf_{ij} + 1.0) * idf_j]^2}$$

Evaluation Metrics

	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	TP	FN
	Class=No	FP	TN

$$\text{Precision (p)} = \frac{tp}{tp+fp}$$

$$\text{Recall (r)} = \frac{tp}{tp+fn}$$

$$\text{F1- measure (F1)} = \frac{2rp}{r+p}$$

Macro-Averaging

- Macro-average:
  - Equal weight to each category

$$\text{Macro-Precision} = \frac{\text{Precision(A)} + \text{Precision(B)} + \text{Precision(C)}}{3}$$

$$\text{Macro-Recall} = \frac{\text{Recall(A)} + \text{Recall(B)} + \text{Recall(C)}}{3}$$

$$\text{Macro-F1 Measure} = \frac{\text{F1 Measure(A)} + \text{F1 Measure(B)} + \text{F1 Measure(C)}}{3}$$

Micro-Averaging

- Micro-average:
  - Equal weight to each sample (record, document)

$$\text{Micro-Precision} = \frac{TP_A + TP_B + TP_C}{TP_A + TP_B + TP_C + FP_A + FP_B + FP_C}$$

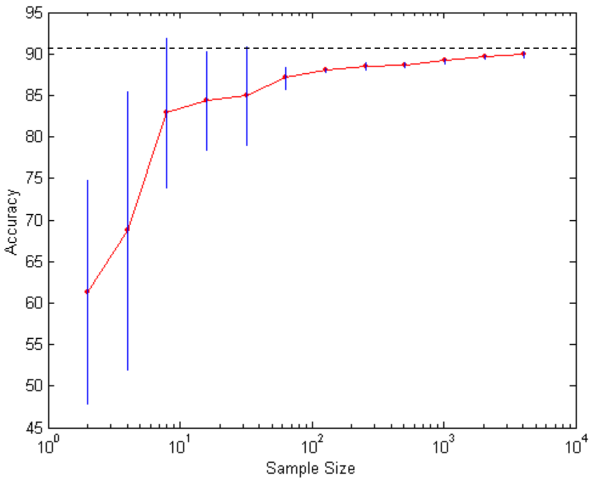
$$\text{Micro-Recall} = \frac{TP_A + TP_B + TP_C}{TP_A + TP_B + TP_C + FN_A + FN_B + FN_C}$$

$$\text{Micro-F1 Measure} = \frac{2 * \text{Micro-Precision} * \text{Micro-Recall}}{\text{Micro} - \text{Precision} + \mathit{Micro} - \text{Recall}}$$

Performance Factors

- Performance of a model may depend on other factors besides the learning algorithm:
  - Class distribution
  - Cost of misclassification
  - Size of training and test sets
  - Good coverage

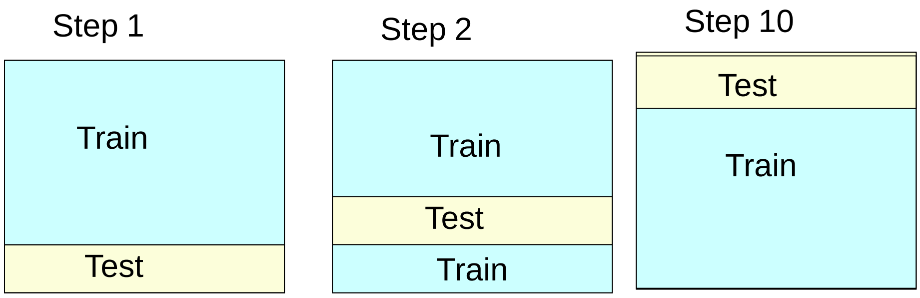
Learning Curve



- Learning curve shows how accuracy changes with varying sample size

10-fold cross validation

- Training data: 90%
- Test data: 10%
- Each run will result in a particular classification rate.
- Average the ten classification rates for a final 10-fold cross validation classification rate.



Evaluation Dataset

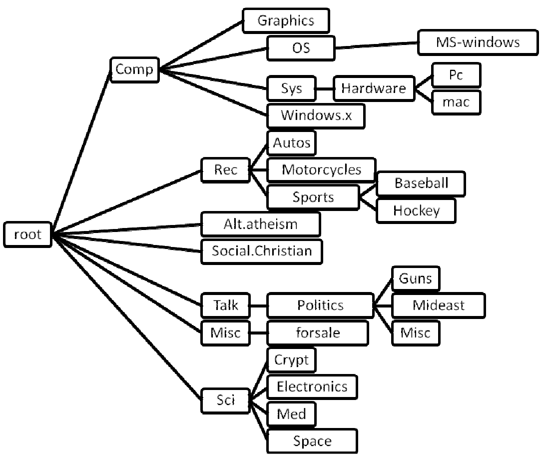
- Manual labeling needs excessive effort
- Available Web directory: *Yahoo directory* & *dmoz ODP (Open Directory Project)*
- Several other sources available – nowadays *Wikipedia*
- Problem– not one given benchmark!
- Not one given domain!

Some of the Text Classification Benchmark Datasets

Datasets	No. of documents	No. of Categories	Size of dataset	Domain
Reuters 21578	21,578	108 Categories (we used top 10)	28 MB	News Articles
20 News Group	20,000	20 categories	61 MB	News Articles
WebKB	8,282	7 categories	43 MB	Web Pages (University websites)
OHSUMED	54,710 (Total) 39,320 (Subset)	4,308 (we used top 50)	382 MB	Bio-medical Documents
GENOMICS (TREC 05)	4.5 million (Total) 591,689 (Subset)	20,184 (we used top 50)	15.5 GB	Bio-medical Documents

More benchmark datasets exist!

Sample Dataset:
20 Newsgroups Hierarchy

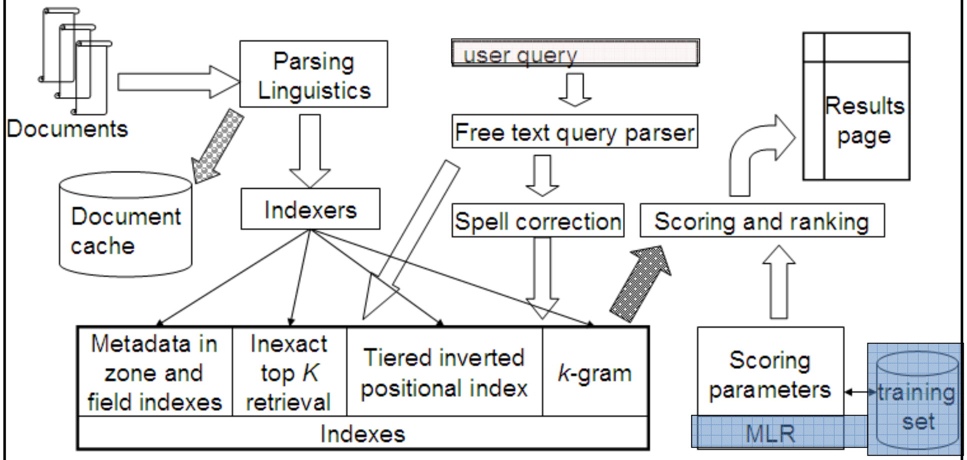


Learning to Rank

Reference: T. Liu, "Learning to Rank for Information Retrieval", Foundations & Trends in Information Retrieval, 2009

Putting it all together (borrowed from:

©D. Manning, P. Raghavan, H. Schutze, Introduction to Information retrieval, p 135, Cambridge University Press., 2008.



Learning to Rank

- Retrieval models need tuning parameters
  - Not a trivial task
  - may lead to overfitting
- Not one retrieval model outcome may suffice for ranking, a combination maybe helpful
  - Thus, using ML to automatically
    - Tune parameters
    - Combine ranking features

"Learning-to-rank" methods are those ranking methods that use ML for ranking!

57

Learning to Rank

- For a given query *q*, its related document *d* is represented as a feature vector $x = \Phi(q, d)$

(Φ is a feature extractor)

Typical features: qtf, BM25, PageRank, link info, …

- Learning process based on training data

(training data is documents, user feedback, log, etc…..)

58

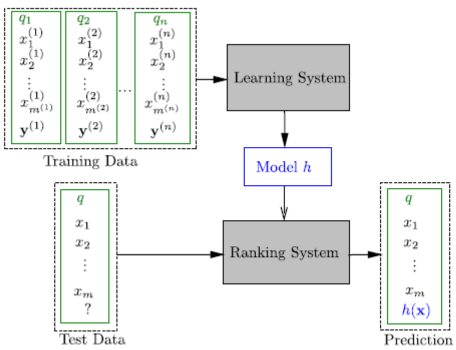
Learning to Rank: Sample Learning Features (Trec)

1 Term frequency (TF) of body
2 TF of anchor
3 TF of title
4 TF of URL
5 TF of whole document
6 Inverse document frequency (IDF) of body
7 IDF of anchor
8 IDF of title
9 IDF of URL
10 IDF of whole document
11 TF*IDF of body
12 TF*IDF of anchor
13 TF*IDF of title
14 TF*IDF of URL
15 TF*IDF of whole document
16 Document length (DL) of body
17 DL of anchor
18 DL of title
19 DL of URL
20 DL of whole document
21 BM25 of body
22 BM25 of anchor
23 BM25 of title
24 BM25 of URL
25 BM25 of whole document
26 LMIR.ABS of body
27 LMIR.ABS of anchor
28 LMIR.ABS of title
29 LMIR.ABS of URL
30 LMIR.ABS of whole document
31 LMIR.DIR of body
32 LMIR.DIR of anchor
33 LMIR.DIR of title
34 LMIR.DIR of URL
35 LMIR.DIR of whole document
36 LMIR.JM of body
37 LMIR.JM of anchor
38 LMIR.JM of title
39 LMIR.JM of URL
40 LMIR.JM of whole document
41 Sitemap based term propagation
42 Sitemap based score propagation
43 Hyperlink base score propagation: weighted in-link
44 Hyperlink base score propagation: weighted out-link
45 Hyperlink base score propagation: uniform out-link
46 Hyperlink base feature propagation: weighted in-link
47 Hyperlink base feature propagation: weighted out-link
48 Hyperlink base feature propagation: uniform out-link
49 HITS authority
50 HITS hub
51 PageRank
52 HostRank
53 Topical PageRank
54 Topical HITS authority
55 Topical HITS hub
56 Inlink number
57 Outlink number
58 Number of slash in URL
59 Length of URL
60 Number of child page
61 BM25 of extracted title
62 LMIR.ABS of extracted title
63 LMIR.DIR of extracted title
64 LMIR.JM of extracted title

T. Liu, "Learning to Rank for Information Retrieval", Foundations & Trends in Information Retrieval, 2009

59

Learning to Rank: Framework



Different approaches exist -- based on how to perform the learning (*input, output, scoring functions,….*)!

T. Liu, "Learning to Rank for Information Retrieval", Foundations & Trends in Information Retrieval, 2009

60

References used to prepare this set of slides:

- *Nazli Goharian & Saket Mengle slides for Text Classification lecture, 2009*

Other references:

- *ACM Computing Surveys article by X. Qi and B. Davison, 2009*
- *Introduction to Information Retrieval, Manning, Raghavan and Schütze, 2008*
- *Introduction to Data Mining, Tan, Steinbach, Kumar, Addison Wesley, 2006*
- *Data Mining Concepts and Techniques, Han, Kamber, Pei, Morgan Kaufmann, 2011*
- *Machine Learning in Automated Text Categorization, F. Sebastiani, 2002*
- *T. Liu, "Learning to Rank for Information Retrieval", Foundations & Trends in Information Retrieval, 2009*