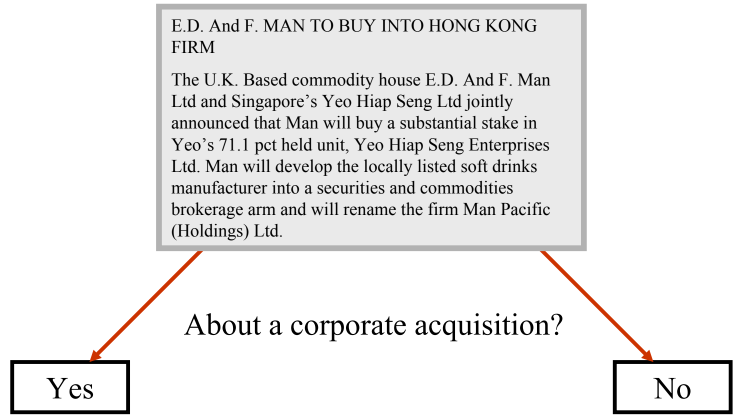
Text Classification

Thorsten Joachims
Cornell University

Text Classification

Definition: **Assign pieces of text to predefined categories based on content.**



E.D. And F. MAN TO BUY INTO HONG KONG FIRM

The U.K. Based commodity house E.D. And F. Man Ltd and Singapore's Yeo Hiap Seng Ltd jointly announced that Man will buy a substantial stake in Yeo's 71.1 pct held unit, Yeo Hiap Seng Enterprises Ltd. Man will develop the locally listed soft drinks manufacturer into a securities and commodities brokerage arm and will rename the firm Man Pacific (Holdings) Ltd.

About a corporate acquisition?

Yes

No

Text Classification Tasks

- **Types of text**
  - Documents (typical)
  - Paragraphs
  - Sentences
  - WWW-Sites
- **Different types of categories**
  - By topic
  - By function
  - By author
  - By style

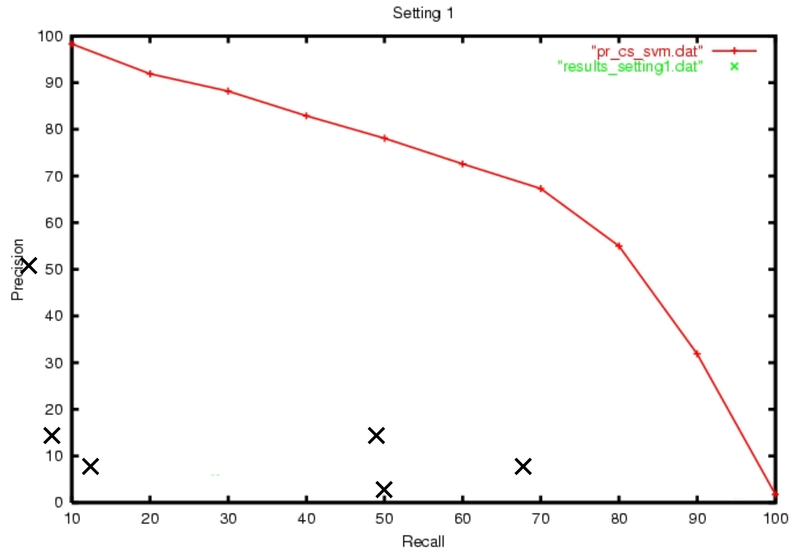
Text Classification Applications

- **Help-Desk Support**
  - Who is an appropriate expert for a particular problem?
- **Information Filtering Agent**
  - Which news articles are interesting to a particular person?
- **Relevance Feedback**
  - What are other documents relevant for a particular query?
- **Knowledge Management**
  - Organizing a document database by semantic categories.
- **Focused Crawling**
  - Find all the WWW pages on a particular topic.

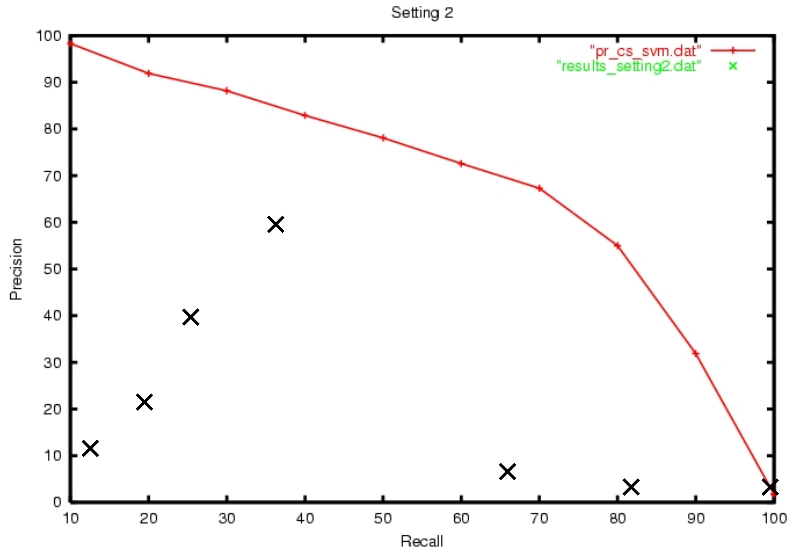
Why Learn Text Classifiers

- **Classifying documents by hand is costly and does not scale well**
  - e.g. browse all WWW pages to filter out those about job announcements
- **Humans are not really good at constructing text classification rules**
  - It is hard to write good queries
- **Sometimes there is no expert available**
  - e.g. rules for routing email
- **Often training data is cheap and plenty**
  - e.g. clickthrough from users, existing databases

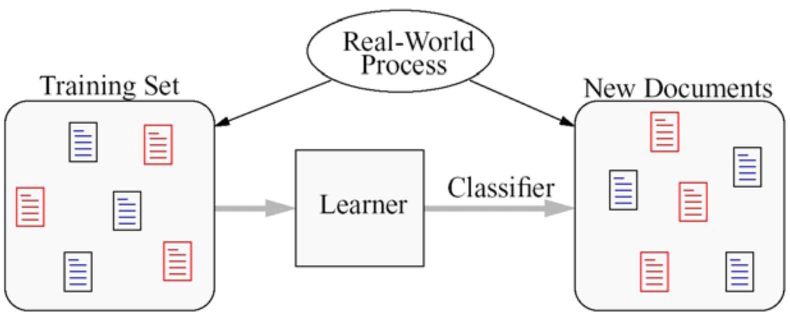
Human vs. ML: Setting 1



Human vs. ML: Setting 2



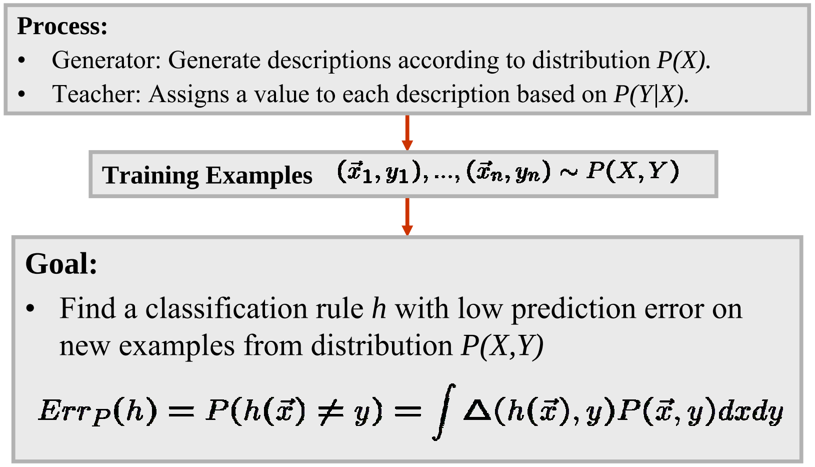
Learning Setting



Goal:

- Learner uses training set to find classifier with low prediction error.

Learning Setting



Process:

- Generator: Generate descriptions according to distribution *P(X)*.
- Teacher: Assigns a value to each description based on *P(Y|X)*.

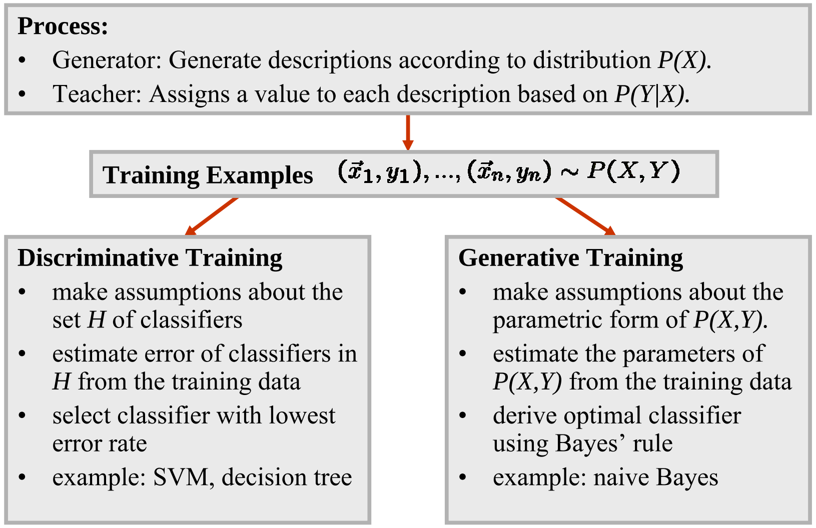
Training Examples $(\vec{x}_1, y_1), ..., (\vec{x}_n, y_n) \sim P(X,Y)$

Goal:

- Find a classification rule *h* with low prediction error on new examples from distribution *P(X,Y)*

$$Err_P(h) = P(h(\vec{x}) \neq y) = \int \Delta(h(\vec{x}), y) P(\vec{x}, y) dx dy$$

Generative vs. Discriminative Training



Process:

- Generator: Generate descriptions according to distribution *P(X)*.
- Teacher: Assigns a value to each description based on *P(Y|X)*.

Training Examples $(\vec{x}_1, y_1), ..., (\vec{x}_n, y_n) \sim P(X,Y)$

Discriminative Training

- make assumptions about the set *H* of classifiers
- estimate error of classifiers in *H* from the training data
- select classifier with lowest error rate
- example: SVM, decision tree

Generative Training

- make assumptions about the parametric form of *P(X,Y)*.
- estimate the parameters of *P(X,Y)* from the training data
- derive optimal classifier using Bayes' rule
- example: naive Bayes

Unigram Model for Text

- **What is the probability of seeing a document in class +1 vs. class –1**
  - Need to estimate *P(Y=1| X=x) ~ P(X=x | Y=1)P(Y=1)* and *P(Y=-1| X=x) ~ P(X=x | Y=-1) P(Y=-1)*
- **Assume that words are drawn randomly from class dependent lexicons (with replacement)**
- **Result**
  - l_x is the total number of words in the document *x*
  - w_i is the *i*-th word in the document

$$P(X = \vec{x}|Y = 1) = \prod_{i=1}^{l_x} P(W = w_i|Y = 1)$$

$$P(X = \vec{x}|Y = -1) = \prod_{i=1}^{l_x} P(W = w_i|Y = -1)$$

Naïve Bayes' Classifier for Text

- **Multinomial model for each class**

$$P(X = \vec{x}|Y) = \prod_{i=1}^{l_x} P(W = w_i|Y)$$

- **Prior probabilities**

$$P(Y)$$

- **Classification rule:**
  - predict class +1 if

$$P(Y = 1|X = \vec{x}) > P(Y = -1|X = \vec{x})$$

$$\Longleftrightarrow$$

$$P(Y = 1) \prod_{i=1}^{l_x} P(W = w_i|Y = 1) > P(Y = -1) \prod_{i=1}^{l_x} P(W = w_i|Y = -1)$$

  - else, predict class -1

Estimating the Parameters

- **Count frequencies in training data**
  - *n*: number of training examples
  - *pos/neg*: number of positive/negative training examples
  - *TF(w,y)*: number of times word w occurs in class y
  - l_y: number of words occurring in documents in class y
- **Estimating P(Y)**
  - Fraction of positive / negative examples in training data

$$\hat{P}(Y=1)=\frac{pos}{n} \qquad \hat{P}(Y=-1)=\frac{neg}{n}$$

- **Estimating P(W|Y)**
  - Smoothing with Laplace estimate

$$\hat{P}(W=w|Y=y)=\frac{TF(w,y)+1}{l_y+2}$$

Assumptions of Naïve Bayes

- **Words occur independently given the class according to one multinomial distribution per class**
- **Each document is in exactly one class**
- **Word probabilities do not depend on the document length**

Test Collections

- **Reuters-21578**
  - Reuters newswire articles classified by topic
  - 90 categories (multi-label)
  - 9603 training documents / 3299 test documents (ModApte)
  - ~27,000 features
- **WebKB Collection**
  - WWW pages classified by function (e.g. personal HP, project HP)
  - 4 categories (multi-class)
  - 4183 training documents / 226 test documents
  - ~38,000 features
- **Ohsumed MeSH**
  - Medical abstracts classified by subject heading
  - 20 categories from "disease" subtree (multi-label)
  - 10,000 training documents/ 10,000 test documents
  - ~38,000 features

Example: Reuters Article (Multi-Label)

Categories: COFFEE, CRUDE

KENYAN ECONOMY FACES PROBLEMS, PRESIDENT SAYS

The Kenyan economy is heading for difficult times after a boom last year, and the country must tighten its belt to prevent the balance of payments swinging too far into deficit, President Daniel Arap Moi said.

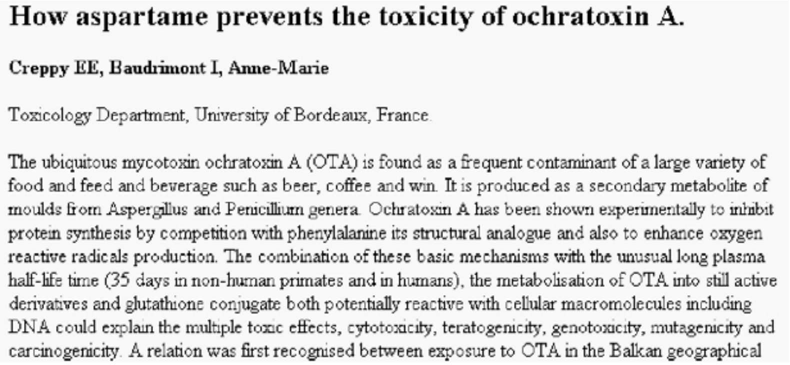
In a speech at the state opening of parliament, Moi said high coffee prices and cheap oil in 1986 led to economic growth of five pct, compared with 4.1 pct in 1985. The same factors produced a two billion shilling balance of payments surplus and inflation fell to 5.6 pct from 10.7 pct in 1985, he added.

"But both these factors are no longer in our favour ... As a result, we cannot expect an increase in foreign exchange reserves during the year," he said.

…

Example: Ohsumed Abstract

Categories: Animal, Blood_Proteins/Metabolism, DNA/Drug_Effects, Mycotoxins/Toxicity, …



How aspartame prevents the toxicity of ochratoxin A.

Creppy EE, Baudrimont I, Anne-Marie

Toxicology Department, University of Bordeaux, France.

The ubiquitous mycotoxin ochratoxin A (OTA) is found as a frequent contaminant of a large variety of food and feed and beverage such as beer, coffee and win. It is produced as a secondary metabolite of moulds from Aspergillus and Penicillium genera. Ochratoxin A has been shown experimentally to inhibit protein synthesis by competition with phenylalanine its structural analogue and also to enhance oxygen reactive radicals production. The combination of these basic mechanisms with the unusual long plasma half-life time (35 days in non-human primates and in humans), the metabolisation of OTA into still active derivatives and glutathione conjugate both potentially reactive with cellular macromolecules including DNA could explain the multiple toxic effects, cytotoxicity, teratogenicity, genotoxicity, mutagenicity and carcinogenicity. A relation was first recognised between exposure to OTA in the Balkan geographical

Multi-Class / Multi-Label

- **Cannot learn multi-label rules directly**
  - Most classifiers assume that each document is in exactly one class
  - Many classifiers can only learn binary classification rules
- **Most common solution: Multi-Label**
  - Learn one binary classifier for each label
  - Attach all labels, for which some classifier says positive
- **Most common solution: Multi-Class**
  - Learn one binary classifier for each label
  - Put example into the class with the highest probability (or some approximation thereof)

Performance Measures

- **Precision/Recall Break-Even Point**
  - Intersection of PR-curve with the identity line
- **Macro-averaging**
  - First compute the measure, then compute average
  - Results in average over tasks
- **Micro-averaging**
  - First average the elements of the contingency table, then compute the measure
  - Results in average over each individual classification decision

Experimental Results

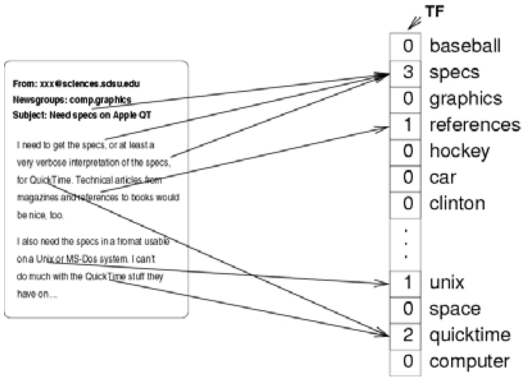
Reuters Newswire
- 90 categories
- 9603 training doc.
- 3299 test doc.
- ~27000 features

WebKB Collection
- 4 categories
- 4183 training doc.
- 226 test doc.
- ~38000 features

Ohsumed MeSH
- 20 categories
- 10000 training doc.
- 10000 test doc.
- ~38000 features

microaveraged precision/recall breakeven-point [0..100]	**Reuters**	**WebKB**	**Ohsumed**
Naive Bayes	72.3	82.0	62.4
Rocchio Algorithm	79.9	74.1	61.5
C4.5 Decision Tree	79.4	79.1	56.7
k-Nearest Neighbors	82.6	80.5	63.4
SVM	**87.5**	**90.3**	**71.6**

Representing Text as Attribute Vectors



From: xxx@sciences.sdsu.edu
Newsgroups: comp.graphics
Subject: Need specs on Apple QT

I need to get the specs, or at least a very verbose interpretation of the specs, for QuickTime. Technical articles from magazines and references to books would be nice, too.

I also need the specs in a fromat usable on a Unix or MS-Dos system. I can't do much with the QuickTime stuff they have on...

TF	
0	baseball
3	specs
0	graphics
1	references
0	hockey
0	car
0	clinton
⋮	
1	unix
0	space
2	quicktime
0	computer

TF IDF weighting heuristic: $x_i = TF(w_i,d) * IDF(w_i)$
$IDF(w_i) = \log(\#docs / \#docs\ containing\ w_i)$

Document length norm.: $x_i := x_i / ||x||$
$||x|| = \Sigma_j\, x_j^2$

=> Ignore ordering of words

K-Nearest Neighbor

- **Given:** $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n) \sim P(X,Y), \vec{x}_i \in \Re^N, y_i \in \{-1,1\}$
- **Preprocessing:**
  - Bring into vector space model representation (e.g. TFIDF)
- **Learning:**
  - None
- **Prediction rule**

$$h(\vec{x}') = sign\left(\sum_{i \in knn(\vec{x}')} y_i cos(\vec{x}_i, \vec{x}')\right)$$

Rocchio Algorithm

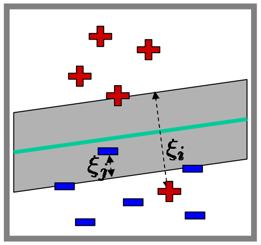
- **Given:** $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n) \sim P(X,Y), \vec{x}_i \in \Re^N, y_i \in \{-1,1\}$
- **Preprocessing:**
  - Split into set of positive / negative examples (ie. D_+ / D_-)
- **Training:**
  - Compute weight vector as weighted difference between prototypes
    $$\vec{w} = \frac{1}{|D_+|}\sum_{\vec{x}_i \in D_+} \vec{x}_i - \beta \frac{1}{|D_-|}\sum_{\vec{x}_i \in D_-} \vec{x}_i$$
  - Often: set negative elements of w vector to zero
- **Prediction rule**
  $$h(\vec{x}') = \begin{cases} 1 & \text{if } cos(\vec{w}, \vec{x}') > \theta \\ -1 & \text{else} \end{cases}$$

Support Vector Machines

- **Given:** $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n) \sim P(X,Y), \vec{x}_i \in \Re^N, y_i \in \{-1,1\}$
- **Training:**
  - Find linear classifier that minimizes training error and that has large margin

Soft-Margin OP (Primal):

$$\min_{\vec{w},\vec{\xi},b} \frac{1}{2}\vec{w}\cdot\vec{w} + C\sum_{i=1}^{n}\xi_i$$
$$s.t.\ y_1(\vec{w}\cdot\vec{x}_1 + b) \geq 1 - \xi_1 \wedge \xi_1 \geq 0$$
$$\ldots$$
$$y_n(\vec{w}\cdot\vec{x}_n + b) \geq 1 - \xi_n \wedge \xi_n \geq 0$$



  - C is a parameter that controls trade-off between margin and training error.
- **Prediction rule**
  $$h(\vec{x}') = \begin{cases} 1 & \text{if } (\vec{w}\cdot\vec{x}' + b) > 0 \\ -1 & \text{else} \end{cases}$$

Experimental Results

Reuters Newswir e
- 90 cate gories
- 9603 training doc.
- 3299 test doc.
- ~27000 features

WebKB Collection
- 4 cate gories
- 4183 training doc.
- 226 test doc.
- ~38000 features

Ohsumed MeSH
- 20 cate gories
- 10000 training doc.
- 10000 test doc.
- ~38000 features

microaveraged precision/recall breakeven-point [0..100]	**Reuters**	**WebKB**	**Ohsumed**
Naive Bayes	72.3	82.0	62.4
Rocchio Algorithm	79.9	74.1	61.5
C4.5 Decision Tree	79.4	79.1	56.7
k-Nearest Neighbors	82.6	80.5	63.4
SVM	**87.5**	**90.3**	**71.6**

Comparison of Methods

	Naïve Bayes	**Rocchio**	**C4.5**	**K-NN**	**SVM**
Simplicity (conceptual)	+	++	-	++	-
Efficiency at training	+	+	--	++	-
Efficiency at prediction	++	++	+	--	++
Handling many classes	+	+	--	++	-
Theoretical understanding	**o**	--	-	**o**	+
Prediction accuracy	-	**o**	-	+	++
Stability and robustness	-	-	--	+	++