CS630 Representing and Accessing Digital Information

Text Classification: Intro and Naïve Bayes

Thorsten Joachims
Cornell University

Text Classification Example

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

- **Assign pieces of text to predefined categories based on content**
- **Types of text**
  - Documents (typical)
  - Paragraphs
  - Sentences
  - WWW-Sites
- **Different types of categories**
  - By topic
  - By function
  - By author
  - By style

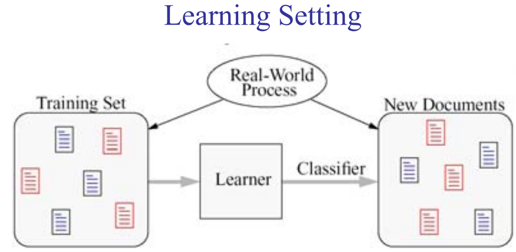
Text Classification Applications

- **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

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

↓

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

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

Prediction Error and Loss Function

- **Prediction error**
  - Also generalization error or true error
  - Probability of making an error on a new example drawn from the same distribution *P(X,Y)*
  - Equivalent: Expected value of loss function

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

- **Loss function**
  - Assigns amount of "penalty" when making a mistake
  - Zero/One-Loss:

$$\Delta(h(\vec{x}), y) = \begin{cases} 0 & \text{if } h(\vec{x}) = y \\ 1 & \text{else} \end{cases}$$

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

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

Bayes' Rule

- **If you know *P(Y=1 | X)* and *P(Y=-1 | X)*, the optimal classification is**

$$h(\vec{x}) = \begin{cases} 1 & \text{if } P(Y=1|X=\vec{x}) > P(Y=-1|X=\vec{x}) \\ -1 & \text{else} \end{cases}$$

- **Will minimize prediction error**

Bayes' Theorem

- **It is possible to "switch" conditioning according to the following rule**

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- **Note that**

$$P(B) = \sum_{a \in A} P(B|A=a)P(A=a)$$

Bayes Rule/Theorem for Classification

- **Need conditional probability**

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

  to apply Bayes's rule.
- **Use Bayes' theorem to get**

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

- **Equivalence**

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

Unigram Model for Text

- **What is the probability of seeing a document in class +1 vs. class –1**
  - Need to estimate $P(X=x \mid Y=1)P(Y=1)$ and $P(X=x \mid Y=-1) P(Y=-1)$
- **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

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

  - 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**

Pros and Cons for Naïve Bayes

- **Pros:**
  - Explicit theoretical foundation
  - Relatively effective
  - Very simple
  - Fast in learning and classification
- **Cons:**
  - Multinomial model / independence assumption clearly wrong for text
  - Performs worse than other methods in practice
  - on some datasets it really fails badly