Text classification I (Naïve Bayes)

CE-324: Modern Information Retrieval

Sharif University of Technology

M. Soleymani

Fall 2015

Outline

- Text classification
  - definition
  - relevance to information retrieval
- Naïve Bayes classifier

Formal definition of text classification

- Document space X
  - Docs are represented in this (typically high-dimensional) space
- Set of classes $C = \{c_1, \dots, c_K\}$
  - Example: $C = \{\text{spam, non-spam}\}$
- Training set: a set of labeled docs. Each labeled doc $\langle d, c \rangle \in X \times C$

- Using a learning method, we find a classifier $\gamma(.)$ that maps docs to classes: $\gamma: X \to C$

Examples of using classification in IR systems

- Language identification (classes: English vs. French etc.)
- Automatic detection of spam pages (spam vs. non-spam)
- Automatic detection of secure pages for safe search
- Topic-specific or vertical search – restrict search to a "vertical" like "related to health" (relevant to vertical vs. not)
- Sentiment detection: is a movie or product review positive or negative (positive vs. negative)

- Exercise: Find examples of uses of text classification in IR

Bayes classifier

- Bayesian classifier is a probabilistic classifier:

$$c = \underset{k}{\operatorname{argmax}}\, P(C_k|d)$$
$$c = \underset{k}{\operatorname{argmax}}\, P(d|C_k)P(C_k)$$

- $d = \langle t_1, \dots, t_{L_d} \rangle$

- There are too many parameters $P(\langle t_1, \dots, t_{L_d} \rangle | C_k)$
  - One for each unique combination of a class and a sequence of words.
  - We would need a very, very large number of training examples to estimate that many parameters.

Naïve bayes assumption

- Naïve bayes assumption:

$$P(d|C_k) = P\left(\langle t_1, \dots, t_{L_d}\rangle \middle| C_k\right) \propto \prod_{i=1}^{L_d} P(t_i|C_k)$$

  - L_d: length of doc d (number of tokens)
  - $P(t_i|C_k)$: probability of term t_i occurring in a doc of class C_k
  - $P(C_k)$: prior probability of class C_k.

Naive Bayes classifier

- Since log is a monotonic function, the class with the highest score does not change.

$$c = \underset{k}{\operatorname{argmax}}\, P(d|C_k)P(C_k) = \underset{k}{\operatorname{argmax}}\, P(C_k) \prod_{i=1}^{L_d} P(t_i|C_k)$$

$$c = \underset{k}{\operatorname{argmax}}\ \log P(C_k) + \sum_{i=1}^{L_d} \log P(t_i|C_k)$$

$\log P(t_i|C_k)$: a weight that indicates how good an indicator t_i is for C_k

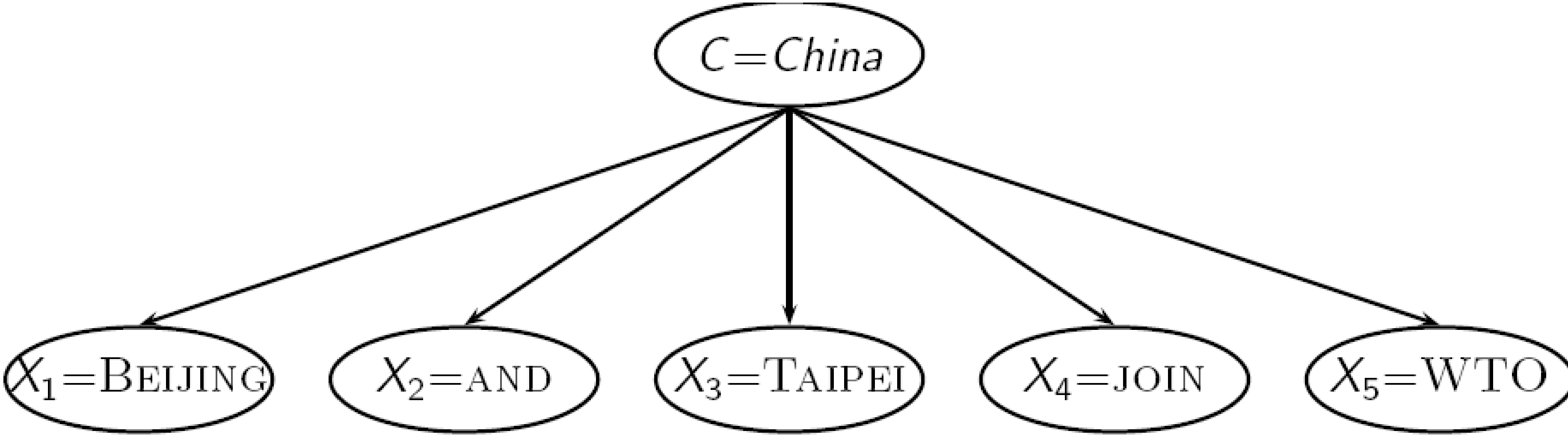
log(xy) = log(x) + log(y)

Estimating parameters

- Estimate $\hat{P}(C_k)$ and $\hat{P}(t_i|C_k)$ from training data
  - N_k: number of docs in class C_k
  - $T_{i,k}$: number of occurrence of t_i in training docs from class C_k (includes multiple occurrences)

- $\hat{P}(C_k) = \frac{N_k}{N}$
- $\hat{P}(t_i|C_k) = \frac{T_{i,k}}{\sum_{j=1}^{M} T_{j,k}}$

Problem with estimates: Zeros



$$
\begin{aligned}
P(China|d) \quad \propto \quad & P(China) \cdot P(\text{BEIJING}|China) \cdot P(\text{AND}|China) \\
& \cdot P(\text{TAIPEI}|China) \cdot P(\text{JOIN}|China) \cdot {\color{red} P(\text{WTO}|China)}
\end{aligned}
$$

d: $BEIGING\ AND\ TAIPEI\ JOIN\ WTO$

$$P(WTO|China) = 0$$

Problem with estimates: Zeros

- For doc d containing a term t that does not occur in any doc of a class $c \Rightarrow \hat{P}(c|d) = 0$
  - Thus d cannot be assigned to class c

- We use

$$\hat{P}(t|c) = \frac{T_{t,c} + 1}{\left(\sum_{t' \in V} T_{t',c}\right) + |V|}$$

Instead of

$$\hat{P}(t|c) = \frac{T_{t,c}}{\sum_{t' \in V} T_{t',c}}$$

Naïve Bayes: summary

- Estimate parameters from the training corpus using add-one smoothing

- For a new doc $d = t_1, \dots, t_{L_d}$, for each class, compute $\log P(C_k) + \sum_{i=1}^{L_d} \log P(t_i|C_k)$

- Assign doc d to the class with the largest score

Naïve Bayes: example

	docID	words in document	in $c = China$?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

- Training phase:
  - Estimate parameters of Naive Bayes classifier
- Test phase
  - Classifying the test doc

Naïve Bayes: example

$C = China$

- Estimating parameters
  - $\hat{P}(C) = \frac{3}{4}, \hat{P}(\bar{C}) = \frac{1}{4}$
  - $\hat{P}(CHINESE|C) = \frac{5+1}{8+6} = \frac{6}{14}$ $\quad \hat{P}(CHINESE|\bar{C}) = \frac{1+1}{3+6} = \frac{2}{9}$
  - $\hat{P}(TOKYO|C) = \frac{0+1}{8+6} = \frac{1}{14}$ $\quad \hat{P}(TOKYO|\bar{C}) = \frac{1+1}{3+6} = \frac{2}{9}$
  - $\hat{P}(JAPAN|C) = \frac{0+1}{8+6} = \frac{1}{14}$ $\quad \hat{P}(JAPAN|\bar{C}) = \frac{1+1}{3+6} = \frac{2}{9}$

- Classifying the test doc:
- $\hat{P}(C|d) \propto \frac{3}{4} \times \left(\frac{6}{14}\right)^3 \times \frac{1}{14} \times \frac{1}{14} \approx 0.0003$
- $\hat{P}(\bar{C}|d) \propto \frac{1}{4} \times \left(\frac{2}{9}\right)^3 \times \frac{2}{9} \times \frac{2}{9} \approx 0.0001$

$\Rightarrow \quad \hat{c} = C$

Naïve Bayes: training

$\text{TrainMultinomialNB}(\mathbb{C}, \mathbb{D})$

$V \leftarrow \text{ExtractVocabulary}(\mathbb{D})$

$N \leftarrow \text{CountDocs}(\mathbb{D})$

for each $c \in \mathbb{C}$

do $N_c \leftarrow \text{CountDocsInClass}(\mathbb{D}, c)$

$\quad prior[c] \leftarrow N_c / N$

$\quad text_c \leftarrow \text{ConcatenateTextOfAllDocsInClass}(\mathbb{D}, c)$

$\quad$ **for each** $t \in V$

$\quad$ **do** $T_{ct} \leftarrow \text{CountTokensOfTerm}(text_c, t)$

$\quad$ **for each** $t \in V$

$\quad$ **do** $condprob[t][c] \leftarrow \frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$

return $V, prior, condprob$

Naïve Bayes: test

$\textsc{ApplyMultinomialNB}(\mathbb{C}, V, \mathit{prior}, \mathit{condprob}, d)$

$W \leftarrow \textsc{ExtractTokensFromDoc}(V, d)$
for **each** $c \in \mathbb{C}$
do $\mathit{score}[c] \leftarrow \log \mathit{prior}[c]$
    for **each** $t \in W$
    do $\mathit{score}[c] += \log \mathit{condprob}[t][c]$
return $\arg\max_{c \in \mathbb{C}} \mathit{score}[c]$

Time complexity of Naive Bayes

mode	time complexity
training	$\Theta(\lvert\mathbb{D}\rvert L_{ave} + \lvert\mathbb{C}\rvert\lvert V\rvert)$
testing	$\Theta(L_a + \lvert\mathbb{C}\rvert M_a) = \Theta(\lvert\mathbb{C}\rvert M_a)$

Generally: $|\mathbb{C}||V| < |D|L_{ave}$

- D: training set, V: vocabulary, $\mathbb{C}$: set of classes
- L_{ave}: average length of a training doc
- L_a: length of the test doc
- M_a: number of distinct terms in the test doc

- Thus: Naive Bayes is linear in the size of the training set (training) and the test doc (testing).
  - This is optimal time.

Why does Naive Bayes work?

- The independence assumptions do not really hold of docs written in natural language.

- Naive Bayes can work well even though these assumptions are badly violated.

- Classification is about predicting the correct class and not about accurately estimating probabilities.
  - Naive Bayes is terrible for correct estimation . . .
  - but it often performs well at choosing the correct class.

Naive Bayes is not so naive

- Naive Bayes has won some bakeoffs (e.g., KDD-CUP 97)
- A good dependable baseline for text classification (but not the best)
  - Optimal if independence assumptions hold (never true for text, but true for some domains)
  - More robust to non-relevant features than some more complex learning methods
  - More robust to concept drift (changing of definition of class over time) than some more complex learning methods
- Very fast
- Low storage requirements

Reuters collection

symbol	statistic	value
N	documents	800,000
L	avg. # word tokens per document	200
M	word types	400,000

type of class	number	examples
region	366	UK, China
industry	870	poultry, coffee
subject area	126	elections, sports

Evaluating classification

- Evaluation must be done on test data that are independent of the training data
  - training and test sets are disjoint.
- Measures: Precision, recall, F1, accuracy
  - F1 allows us to trade off precision against recall (harmonic mean of P and R).

Precision P and recall R

	actually in the class	actually in the class
predicted to be in the class	tp	fp
Predicted not to be in the class	fn	tn

Precision P = tp/(tp + fp)

Recall R = tp/(tp + fn)

Averaging: macro vs. micro

- We now have an evaluation measure (F1) for one class.

- But we also want a single number that shows aggregate performance over all classes
  - Macroaveraging
    - Compute F1 for each of the C classes
    - Average these C numbers
  - Microaveraging
    - Compute TP, FP, FN for each of the C classes
    - Sum these C numbers (e.g., all TP to get aggregate TP)
    - Compute F1 for aggregate TP, FP, FN

Comparision

	NB	Rocchio	kNN	trees	SVM
earn	96	93	97	98	98
acq	88	65	92	90	94
money-fx	57	47	78	66	75
grain	79	68	82	85	95
crude	80	70	86	85	89
trade	64	65	77	73	76
interest	65	63	74	67	78
ship	85	49	79	74	86
wheat	70	69	77	93	92
corn	65	48	78	92	90
micro-avg (top 10)	82	65	82	88	92
micro-avg-D (118 classes)	75	62	n/a	n/a	87

Evaluation measure: F1

Resources

- Chapter 13 of IIR