# Machine Learning for natural language processing

## Classification: Naive Bayes


Laura Kallmeyer

Heinrich-Heine-Universität Düsseldorf


Summer 2016

# Introduction

- Classification = supervised method for classifying an input, given a finite set of possible classes.
- Today: Generative classifier that builds a model for each class.

Jurafsky & Martin (2015), chapter 7, and Manning et al. (2008), chapter 13

# Table of contents

# Motivation

In the following, we are concerned with **text classification**: the task of classifying an entire text by assigning it a label drawn from some finite of labels.

Common text categorization tasks:

- **sentiment analysis**
- **spam detection**
- **authorship attribution**

Some classifiers operate with hand-written rules. Our focus is, however, on supervised machine learning.

**Generative** classifiers (e.g., naive Bayes) build a model for each class while **discriminative** classifiers learn useful features for discriminating between the different classes.

# Multinomial naive Bayes classifier

Intuition: Represent a text document as a *bag-of-words* keeping only frequency information but ignoring word order.

**Example**

It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife. However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered the rightful property of some one or other of their daughters.

$\rightsquigarrow$

| | |
|---|---|
| a | 6 |
| of | 6 |
| is | 3 |
| truth | 2 |
| that | 2 |
| man | 2 |
| or | 2 |
| it | 1 |
| universally | 1 |
| acknowledged | 1 |
| … | |

# Multinomial naive Bayes classifier

Naive Bayes returns the class $\hat{c}$ out of the set $C$ of classes wich has the maximum posterior probability given the document $d$:

$$\hat{c} = \arg\max_{c \in C} P(c|d)$$

**Reminder: Bayes' rule**

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

$$\hat{c} = \arg\max_{c \in C} P(c|d) = \arg\max_{c \in C} \frac{P(d|c)P(c)}{P(d)} = \arg\max_{c \in C} P(d|c)P(c)$$

- $P(c)$: prior probability of the class $c$
- $P(d|c)$: likelihood of the document $d$ given the class $c$
- $P(d|c)P(c) = P(d, c)$: joint probability of class and document

# Multinomial naive Bayes classifier

We represent $d$ as a set of features $f_1, \ldots, f_n$ and make the **naive Bayes assumption** that

$$P(f_1, f_2, \ldots, f_n | c) = P(f_1 | c) \ldots P(f_n | c)$$

Each word $w_1, w_2, \ldots, w_{|d|}$ in the document $d$ is a feature:

$$\hat{c} = \arg\max_{c \in C} P(c) \prod_{i=1}^{|d|} P(w_i | c)$$

As usual, we calculate in log space:

$$\hat{c} = \arg\max_{c \in C} \left(\log P(c) + \sum_{i=1}^{|d|} \log P(w_i | c)\right)$$

# Training the classifier

First try: Maximum likelihood estimates, based on frequencies in the training data.

Our training data consists of $N_{doc}$ documents, each of which is in a unique class $c \in C$. $N_c$ is the number of documents belonging to class $c$. $C(w, c)$ gives the number of times word $w$ occurs in a document from class $c$.

$$\hat{P}(c) = \frac{N_c}{N_{doc}}$$

$$\hat{P}(w|c) = \frac{C(w, c)}{\sum_{w'} C(w', c)}$$

# Training the classifier

## Example

Classes $A$ and $B$, documents to be classified are all $d \in \{a, b\}^*$.

Training data:

| $d$ | $c$ | $d$ | $c$ |
|---|---|---|---|
| $aa$ | $A$ | $ba$ | $A$ |
| $ab$ | $A$ | $bb$ | $B$ |

(Note that without any smoothing, this example does not allow to calculate in log space because $\log P(a|B) = \log 0$ is not defined.)

$P(A) = 0.75, P(B) = 0.25$
$P(a|A) = \frac{4}{6} = \frac{2}{3}, P(b|A) = \frac{2}{6} = \frac{1}{3}, \quad P(a|B) = \frac{0}{2} = 0, P(b|B) = \frac{2}{2} = 1$

Classification of new documents:

| $d$ | $P(d|A)$ | $P(d|A)P(A)$ | $P(d|B)$ | $P(d|B)P(B)$ | class |
|---|---|---|---|---|---|
| $aaba$ | 0.1 | 0.075 | 0 | 0 | $A$ |
| $aaa$ | 0.3 | 0.225 | 0 | 0 | $A$ |
| $bbba$ | 0.02 | 0.015 | 0 | 0 | $A$ |
| $bbbb$ | 0.01 | 0.0075 | 1 | 0.25 | $B$ |

# Training the classifier

Problems:

- Unseen combinations of $w$ and $c$.
- Unknown words.

Simplest solution for unseen $w$, $c$ combinations: add-one (Laplace) smoothing, commonly used in naive Bayes text categorization.

$$\hat{P}(w|c) = \frac{C(w,c)+1}{\sum_{w'}(C(w',c)+1)} = \frac{C(w,c)+1}{\sum_{w'} C(w',c)+|V|}$$

($V$ being the vocabulary.)

Standard solution for unknown words: simply remove them from the test document.

# Training the classifier

**Example from Jurafsky & Martin (2015), chapter 7**

| | $c$ | $d$ |
|---|---|---|
| Training | - | "just plain boring" |
| | - | "entirely predictable and lacks energy" |
| | - | "no surprises and very few laughs" |
| | + | "very powerful" |
| | + | "the most fun film of the summer" |
| Test | ? | $S$ = "predictable ~~with~~ no ~~originality~~" |

$$P(-) = \frac{3}{5}, \quad P(+) = \frac{2}{5}, \quad |V| = 20$$

$$P(S|-)P(-) = \frac{1+1}{14+20}\frac{1+1}{14+20}\frac{3}{5} = \frac{2\times 2\times 3}{34\times 34\times 5} = 0.002076$$

$$P(S|+)P(+) = \frac{1}{9+20}\frac{1}{9+20}\frac{2}{5} = \frac{2}{29\times 29\times 5} = 0.000476$$

# Evaluation

First consider the simple case of $|C| = 2$, i.e., we have only 2 possible classes, i.e., we label "is in $c$" or "is not in $c$".

The classifier is evaluated on human labeled data (**gold labels**). For each document, we have a gold label and a system label. Four possibilities:

| | gold positive | gold negative |
|---|---|---|
| system positive | true positive | false positive |
| system negative | false negative | true negative |

# Evaluation

The following evaluation metrics are used (t/fp/n = number of true/false positives/negatives):

1. **Precision**: How many of the items the system classified as positive are actually positive?

$$\textbf{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}$$

2. **Recall**: How many of the positives are classified as positive by the system?

$$\textbf{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}$$

3. **Accuracy**: How many of the classes the system has assigned are correct?

$$\textbf{Accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{fp} + \text{tn} + \text{fn}}$$

# Evaluation

The **F-measure** combines precision $P$ and recall $R$:

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

$\beta$ weights the importance of precision and recall:

- $\beta > 1$ favors recall;
- $\beta < 1$ favors precision;
- $\beta = 1$: both are equally important.

With $\beta = 1$, we get

$$F_1 = \frac{2PR}{P + R}$$

($F_1$ = Harmonic mean of $P$ and $R$.)

# Evaluation

## Example

Training data:

| $d$ | $c$ | $d$ | $c$ |
|---|---|---|---|
| $a$ | $A$ | $aa$ | $A$ |
| $b$ | $A$ | $bb$ | $B$ |

Test data:

| $d$ | $c$ | $d$ | $c$ |
|---|---|---|---|
| $ab$ | $A$ | $bba$ | $B$ |
| $ba$ | $A$ | $bbbb$ | $B$ |

$$\log P(A) = \log \tfrac{3}{4} = -0.12$$
$$\log P(B) = \log \tfrac{1}{4} = -0.6$$
$$\log P(a|A) = \log \tfrac{4}{6} = -0.18$$
$$\log P(b|A) = \log \tfrac{2}{6} = -0.48$$
$$\log P(a|B) = \log \tfrac{1}{4} = -0.6$$
$$\log P(b|B) = \log \tfrac{3}{4} = -0.12$$

Classes assigned to test data:

$ab$: $\log P(A) + \log P(a|A) + \log P(b|A) = -0.12 - 0.18 - 0.48 = -0.78$

$\log P(B) + \log P(a|B) + \log P(b|B) = -0.6 - 0.6 - 0.12 = -1.32$

$\Rightarrow$ class $A$ since $-0.78 > -1.32$

$bba$: $A$: $-(0.12 + 2 \cdot 0.48 + 0.18) = -1.26$

$B$: $-(0.6 + 2 \cdot 0.12 + 0.6) = -1.44$

$\Rightarrow$ class $A$

# Evaluation

## Example continued

Test data:

| $d$ | $c_{gold}$ | $c_{system}$ | $d$ | $c_{gold}$ | $c_{system}$ |
|---|---|---|---|---|---|
| $ab$ | $A$ | $A$ | $bba$ | $B$ | $A$ |
| $ba$ | $A$ | $A$ | $bbbb$ | $B$ | $B$ |

Evaluation with respect to class $A$ ($B$ = not $A$, i.e., $\neg A$):

| | gold $A$ | gold $\neg A$ |
|---|---|---|
| system $A$ | 2 | 1 |
| system $\neg A$ | 0 | 1 |

$P = \frac{2}{2+1} = 0.67$ $\quad A = \frac{3}{4} = 0.75$

$R = \frac{2}{2+0} = 1$ $\quad F_1 = \frac{2 \cdot \frac{2}{3}}{\frac{2}{3}+1} = 0.8$

Evaluation with respect to class $B$:

| | gold $B$ | gold $\neg B$ |
|---|---|---|
| system $B$ | 1 | 0 |
| system $\neg B$ | 1 | 2 |

$P = \frac{1}{1} = 1$ $\quad A = \frac{3}{4} = 0.75$

$R = \frac{1}{2} = 0.5$ $\quad F_1 = \frac{2 \cdot \frac{1}{2}}{\frac{1}{2}+1} = 0.67$

# Evaluation

Evaluation for classifiers with more than 2 classes but a unique class for each document (multinomial classification):
Results can be represented in a *confusion matrix* with one column for every gold class and one row for every sytsem class.

We can compute precision and recall for every single class $c$ as before based on a separate contingency matrix for that class.

The contingency tables can be pooled into one combined contingency table.

Two ways of combining this into an overall evaluation:

1. **Macroaveraging**: average P/R over all classes.
2. **Microaveraging**: compute P/R from the pooled contingency table.

# Evaluation

## Example

We classify documents as to whether they are from the 18th, 19th or 20th century. Our test set comprises 600 gold labeled documents.

Possible confusion matrix:

| | | *gold labels* | | |
|---|---|---|---|---|
| | | 18th | 19th | 20th |
| *system* | 18th | 150 | 35 | 0 |
| *labels* | 19th | 20 | 110 | 5 |
| | 20th | 10 | 10 | 260 |

Separate contingency tables:

| 18th | yes | no |
|---|---|---|
| yes | 150 | 35 |
| no | 30 | 385 |

| 19th | yes | no |
|---|---|---|
| yes | 110 | 25 |
| no | 45 | 420 |

| 20th | yes | no |
|---|---|---|
| yes | 260 | 20 |
| no | 5 | 315 |

Pooled table:

| | yes | no |
|---|---|---|
| y | 520 | 80 |
| n | 80 | 1120 |

# Evaluation

## Example continued

| 18 | yes | no |
|---|---|---|
| y | 150 | 35 |
| n | 30 | 385 |

| 19 | yes | no |
|---|---|---|
| y | 110 | 25 |
| n | 45 | 420 |

| 20 | yes | no |
|---|---|---|
| y | 260 | 20 |
| n | 5 | 315 |

| | yes | no |
|---|---|---|
| y | 520 | 80 |
| n | 80 | 1120 |

Single class P and R:

18th: $P_{18th} = \frac{150}{150+35} = 0.81, \quad R_{18th} = \frac{150}{150+30}$

19th: $P_{19th} = \frac{110}{110+25} = 0.81, \quad R_{19th} = \frac{110}{110+45}$

20th: $P_{20th} = \frac{260}{260+20} = 0.93, \quad R_{20th} = \frac{260}{260+5}$

Macroaverage P: $\frac{0.81+0.81+0.93}{3} = 0.85$

Microaverage P: $\frac{520}{520+80} = 0.87$

Microaverage is dominated by the more frequent classes.

## References


Jurafsky, Daniel & James H. Martin. 2015. Speech and language processing. an introduction to natural language processing, computational linguistics, and speech recognition. Draft of the 3rd edition.

Manning, Christopher D., Prabhakar Raghavan & Hinrich Schütze. 2008. *Introduction to information retrieval.* Cambridge University Press.