Naïve Bayes for Text Classification

Phạm Quang Nhật Minh
Aimesoft JSC
minhpham0902@gmail.com

January 7, 2023

Lecture Contents

- What is text classification?
- Naïve Bayes text classification model
- Text classification evaluation
  - Binary classification
  - Evaluation with more than two classes

Is this spam?


```
Received: from 192.168.1.100 ([65.202.85.3]) by pacific-carrier-annex.mit.edu
          (8.9.2/8.9.2) with SMTP id AAA06179;
          Mon, 11 Jun 2001 00:39:32 -0400 (EDT)
From: [some forged email address]
Message-ID: <200106110439.AAA06179@pacific-carrier-annex.mit.edu>
Subject: I am as shocked as you!
Date: Sun, 10 Jun 01 00:32:35 Pacific Daylight Time
X-Priority: 3
X-MSMailPriority: Normal
Importance: Normal
MIME-Version: 1.0
Content-Type: multipart/mixed;
              boundary="----=_NextPart_000_018C_01BD9940.715D52A0"
```

```
<HTML>
<BODY>

<FONT face="MS Sans Serif">
<FONT size=2> <BR>
<BR>
Some of the most beautiful women in the world bare it all for you.Denise Richard
s, Britney  Spears, Jessica Simpson, and many more.<A HREF="http://216.130.166.1
88/index.html">CLICK HERE FOR NUDE CELEBS<A/><BR>
<BR>
</FONT></FONT></BODY></HTML>
```

Spam=**True**/False

Positive or negative movie review?

👎 Unbelievably disappointing

👍 Full of zany characters and richly applied satire, and some great plot twists

👍 this is the greatest screwball comedy ever filmed

👎 It was pathetic. The worst part about it was the boxing scenes.

Is this Tweet about a flooding event?

1 Fish Creek flooded after a week of rain.

1 Just won't stop raining in Dover, Delaware today. Think we've had 4 inches already.

0 this used to be the road, before the flood. now it's just the short cut to the river.

Text Classification Tasks

- Assigning subject categories, topics, or genres
- Spam detection
- Authorship identification
- Age/gender identification
- Language identification
- Sentiment analysis
- …

Text Classification: definition

- *Input*:
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
- *Output*:
  - a predicted class $c \in C$

Classification Methods: Hand-coded rules

- Rules based on combination of words and other features
  - spam: black-list-address OR (“dollars” AND “have been selected”)
- Accuracy can be high
  - If rules carefully refined by expert
- But building and maintaining these rules is expensive

Supervised Machine Learning

- *Input*:
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
  - A training set of *m* hand-labeled documents $D = \{(d_1, c_1), \ldots, (d_m, c_m)\}$

- *Output*:
  - A learned classifier $\gamma: d \rightarrow c$

Lecture Contents

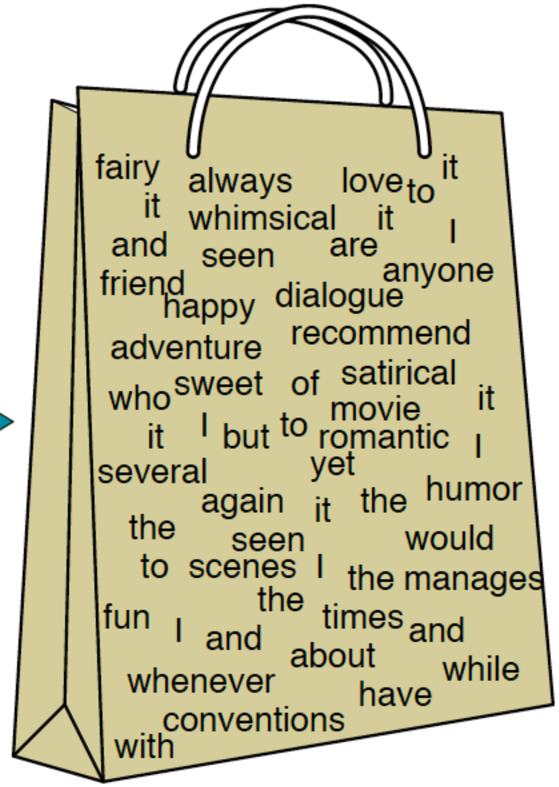
- What is text classification?
- Naïve Bayes text classification model
- Text classification evaluation
  - Binary classification
  - Evaluation with more than two classes

Naïve Bayes Intuition

- Simple ("naïve") classification method based on Bayes rule
- Relies on very simple representation of document
  - Bag of words

The bag of words representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

The bag of words representation

$\gamma\Big($

seen	2
sweet	1
whimsical	1
recommend	1
happy	1
...	...

$\Big) = c$

Bayes' Rule Applied to Documents and Classes

- For a document *d* and a class *c*

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

Naïve Bayes Classifier

- The classifier returns the class $\hat{c}$ which has the maximum posterior probability (MAP) given the document

$$\hat{c} = \operatorname*{argmax}_{c \in C} P(c|d)$$

$$= \operatorname*{argmax}_{c \in C} \frac{P(d|c)P(c)}{P(d)}$$

$$= \operatorname*{argmax}_{c \in C} P(d|c)P(c)$$

Bayes Rule

Drop *P(x)* because *P(x)* is the same for all classes

Naïve Bayes Classifier

- Document *d* is represented as features $(x_1, \dots, x_n)$

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(d|c)P(c)$$
$$= \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \dots, x_n|c)P(c)$$

- It is too hard to compute $P(x_1, x_2, \dots, x_n|c)$
- How can we estimate it?

Naïve Bayes Independence Assumptions

$$P(x_1, x_2, \ldots, x_n | c)$$

- **Bag of Words** assumption: Assume position doesn't matter
- **Conditional Independence:** Assume the feature probabilities $P(x_i|c)$ are independent given the class c

$$P(x_1, x_2, \ldots, x_n | c) = P(x_1|c)P(x_2|c)\ldots P(x_n|c)$$

Multinomial Naïve Bayes Classifier

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n | c) P(c)$$

$$= \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i=1}^{n} P(x_i | c)$$

Naïve Bayes for Text Classification

positions ← all word positions in test documents

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i \in positions} P(w_i|c)$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} \log P(c) + \sum_{i \in positions} \log P(w_i|c)$$

Naïve Bayes is a linear classification model

Learning the Multinomial Naive Bayes Model

- Maximum likelihood estimation (MLE)

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

N_c is the number of documents in class *c* and *N* is the total number of documents

$$\hat{P}(w_i|c) = \frac{\text{count}(w_i, c)}{\sum_{w \in V} \text{count}(w, c)}$$

count(w, c) is the count of the number of word *w* occurs in documents of class *c* in the training data

Problem with Maximum Likelihood

- MLE estimate gets zero for a term-class combination that did not occur in the training data.
- E.g., what if we have seen no training documents with the word *fantastic*

$$\hat{P}(\text{“fantastic”}|\text{positive}) = \frac{\text{count}(\text{“fantastic”}, \text{positive})}{\sum_{w \in V} \text{count}(w, \text{positive})} = 0$$

Laplace (add-1) smoothing for Naïve Bayes

$$\hat{P}(w_i|c) = \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V}(\text{count}(w, c) + 1)}$$

$$= \frac{\text{count}(w_i, c) + 1}{\left(\sum_{w \in V} \text{count}(w, c)\right) + |V|}$$

Multinomial Naïve Bayes: Learning


```
function TRAIN NAIVE BAYES(D, C) returns log P(c) and log P(w|c)

for each class c ∈ C                # Calculate P(c) terms
   N_doc = number of documents in D
   N_c = number of documents from D in class c
   logprior[c] ← log N_c / N_doc
   V ← vocabulary of D
   bigdoc[c] ← append(d) for d ∈ D with class c
   for each word w in V                    #  Calculate P(w|c) terms
      count(w,c) ← # of occurrences of w in bigdoc[c]
      loglikelihood[w,c] ← log (count(w,c) + 1) / (Σ_{w' in V} (count(w',c) + 1))
return logprior, loglikelihood, V


function TEST NAIVE BAYES(testdoc, logprior, loglikelihood, C, V) returns best c

for each class c ∈ C
   sum[c] ← logprior[c]
   for each position i in testdoc
      word ← testdoc[i]
      if word ∈ V
         sum[c] ← sum[c] + loglikelihood[word,c]
return argmax_c sum[c]
```

Where:

$$logprior[c] \leftarrow \log \frac{N_c}{N_{doc}}$$

$$loglikelihood[w,c] \leftarrow \log \frac{count(w,c) + 1}{\sum_{w' \text{ in } V} (count(w',c) + 1)}$$

Worked Example (use add-1 smoothing)

	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	?

- $P(c) = 3/4 \qquad P(\bar{c}) = 1/4$
- $P(\text{Chinese}|c) = (5+1)/(8+6) = 6/14 = 3/7$
- $P(\text{Tokyo}|c) = P(\text{Japan}|c) = (0+1)/(8+6) = 1/14$
- $P(\text{Chinese}|\bar{c}) = (1+1)/(3+6) = 2/9$
- $P(\text{Tokyo}|\bar{c}) = P(\text{Japan}|\bar{c}) = (1+1)/(3+6) = 2/9$
- $P(c|d_5) \propto \frac{3}{4} \times \left(\frac{3}{7}\right)^3 \times \frac{1}{14} \times \frac{1}{14} \approx 0.0003$
- $P(\bar{c}|d_5) \propto \frac{1}{4} \times \left(\frac{2}{9}\right)^3 \times \frac{2}{9} \times \frac{2}{9} \approx 0.0001$
- Thus, the classifier assigns the test document to c = China.

Optimizing for sentiment analysis

- For tasks like sentiment, word occurrence seems to be more important than word frequency.
  - The occurrence of the word fantastic tells us a lot
  - The fact that it occurs 5 times may not tell us much more.
- Binary multinominal naive bayes, or binary NB
  - Clip our word counts at 1

Binary Naive Bayes

- The Binary NB model uses binary occurrence information, ignoring the number of occurrences.
- The multinomial NB model keeps track of multiple occurrences.

Binary NB model: Worked 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	?

	NB Counts		Binay Counts	
	c = yes	c = no	c = yes	c = no
Chinese	5	1	3	1
Beijing	1	0	1	0
Shanghai	1	0	1	0
Macao	1	0	1	0
Tokyo	0	1	0	1
Japan	0	1	0	1

Why Naive Bayes?

- The probability estimates of NB are of low quality, but its classification decisions are surprisingly good.

  Correct estimation implies accurate prediction, but accurate prediction does not imply correct estimation (by Manning, Christopher D.)

- Naive Bayes's main strength is its efficiency:

  Training and classification can be accomplished with one pass over the data.

- Naive Bayes is often used as a baseline in text classification research.

  It combines efficiency with good accuracy.

Agenda

- What is text classification?
- Naïve Bayes text classification model
- Text classification evaluation

Evaluation

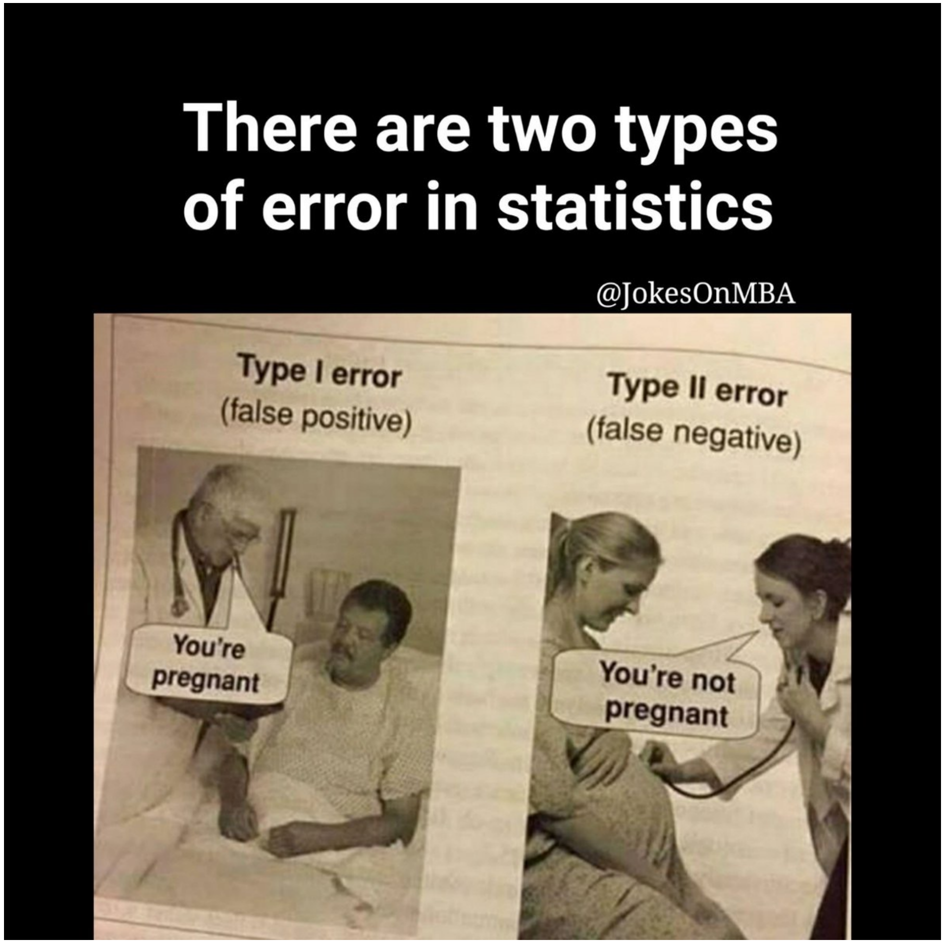
- Let's consider just binary text classification tasks
- Imagine you're the CEO of Delicious Pie Company
- You want to know what people are saying about your pies
- So you build a "Delicious Pie" tweet detector
  - Positive class: tweets about Delicious Pie Co
  - Negative class: all other tweets

The 2-by-2 confusion matrix

gold standard labels

		gold positive	gold negative	
system output labels	system positive	**true positive**	**false positive**	**precision** $= \frac{\text{tp}}{\text{tp+fp}}$
	system negative	**false negative**	**true negative**	
		recall $= \frac{\text{tp}}{\text{tp+fn}}$		**accuracy** $= \frac{\text{tp+tn}}{\text{tp+fp+tn+fn}}$

Two types of errors in statistics



Evaluation: Accuracy

- Why don't we use **accuracy** as our metric?
- Imagine we saw 1 million tweets
  - 100 of them talked about Delicious Pie Co.
  - 999,900 talked about something else
- We could build a dumb classifier that just labels every tweet "not about pie"
  - It would get 99.99% accuracy!!! Wow!!!!
  - But useless! Doesn't return the comments we are looking for!
  - That's why we use **precision** and **recall** instead

Evaluation: Precision

% of items the system detected (i.e., items the system labeled as positive) that are in fact positive (according to the human gold labels)

$$\textbf{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

Evaluation: Recall

% of items actually present in the input that were correctly identified by the system.

$$\textbf{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Why Precision and Recall

Our dumb pie-classifier

Just label nothing as "about pie"

Accuracy=99.99%

but

Recall = 0

(it doesn't get any of the 100 Pie tweets)

Precision and recall, unlike accuracy, emphasize true positives:

finding the things that we are supposed to be looking for.

A combined measure: F

- F measure: a single number that combines P and R:

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

- We almost always use balanced F_1 (i.e., $\beta = 1$)

$$\mathrm{F}_1 = \frac{2PR}{P + R}$$

Why harmonic means?

- Classifier1: P:0.53, R:0.36
- Classifier2: P:0.01, R:0.99

Harmonic	Average
0.429	0.445
0.019	0.500

Agenda

- What is text classification?
- Naïve Bayes text classification model
- Text classification evaluation
  - Evaluation with more than two classes

Confusion Matrix for 3-class classification

gold labels

system output		urgent	normal	spam	
	urgent	8	10	1	**precision**$_u$ $= \frac{8}{8+10+1}$
	normal	5	60	50	**precision**$_n$ $= \frac{60}{5+60+50}$
	spam	3	30	200	**precision**$_s$ $= \frac{200}{3+30+200}$
		recall$_u$ $= \frac{8}{8+5+3}$	**recall**$_n$ $= \frac{60}{10+60+30}$	**recall**$_s$ $= \frac{200}{1+50+200}$	

How to combine P/R from 3 classes to get one metric

Macroaveraging:

compute the performance for each class, and then average over classes

Microaveraging:

collect decisions for all classes into one confusion matrix

compute precision and recall from that table.

Macroaveraging and Microaveraging

Class 1: Urgent

	true urgent	true not
system urgent	8	11
system not	8	340

$$\text{precision} = \frac{8}{8+11} = .42$$

Class 2: Normal

	true normal	true not
system normal	60	55
system not	40	212

$$\text{precision} = \frac{60}{60+55} = .52$$

Class 3: Spam

	true spam	true not
system spam	200	33
system not	51	83

$$\text{precision} = \frac{200}{200+33} = .86$$

Pooled

	true yes	true no
system yes	268	99
system no	99	635

$$\text{microaverage precision} = \frac{268}{268+99} = \mathbf{.73}$$

$$\text{macroaverage precision} = \frac{.42+.52+.86}{3} = \mathbf{.60}$$

Development Test Sets and Cross-validation

Training set	Development Test Set	Test Set

- Metric: P/R/F1 or Accuracy
- Unseen test set
  - avoid overfitting ("tuning to the test set")
  - more conservative estimate of performance
- Cross-validation over multiple splits
  - *k*-fold cross validation or multiple train/test splits

K-fold cross validation

- Break up data into 10 folds
  - (Equal positive and negative inside each fold?)
- For each fold
  - Choose the fold as a temporary test set
  - Train on 9 folds, compute performance on the test fold
- Report average performance of the 10 runs