Text Classification and Naïve Bayes

These slides are based on:

Dan Jurafsky and James H. Martin, Speech and Language Processing (3rd ed. draft)
https://web.stanford.edu/~jurafsky/slp3/
(Chapter 7)

These slides are an edited version of Jurafsky's slides:
https://web.stanford.edu/~jurafsky/NLPCourseraSlides.html

Examples of text classification problems

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 spam?

From Jack <huixinsoft17@foxmail.com>
Subject lab furniture info from Kerric
To [redacted]

Reply | Reply All | Forward | Archive | Junk | Delete | More
05:48

Dear My Friends,

Good day!

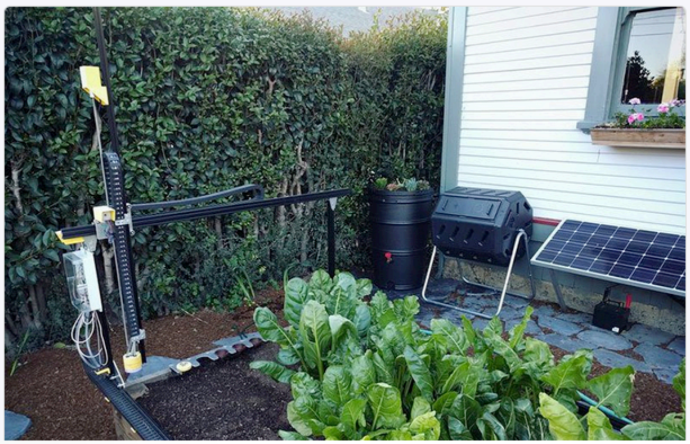
Kerric Laboratory Equipment Research & Development Manufacturer Co.,Ltd is established since 1999, is the leading manufacturer & supplier of the LABORATORY FURNITURE and relevant accessory for school, college, university and chemical or biology industry.

Our products Range Includes:
Laboratory Bench (All Steel, Steel Wood)
Laboratory Cabinet(All Steel, Aluminum Wood)
Fume Hood(All steel, PP)
Laboratory Rack/Shelf(All Steel, Steel Wood)

What is the topic of this post?

Liked 33 times

Smithsonian Magazine @SmithsonianMag · Jul 6
The **FarmBot** Genesis brings precision agriculture to your own backyard via @ModFarm. smithmag.co/iNSxKN



Cristiano Ronaldo @Cristiano · Jul 8
Time to focus. Counting down the hours with my @TAGHeuer . See you on Sunday on the field. #dontcrackunderpressure

?

- Agriculture
- Robotics
- Sport
- Religion
- Psychology
- …

Text Classification

Assigning subject categories:

- Sentiment analysis
- Spam detection
- Language identification
- Authorship identification
- Topic identification
- …

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 and dictionary methods

- Rules based on combinations of words or 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

Text Classification: Supervised Machine Learning

Classification Methods: 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 $(d_1,c_1),\ldots,(d_m,c_m)$
- *Output:*
  - a learned classifier $\gamma: d \rightarrow c$

Classification Methods: Supervised Machine Learning

- Any kind of classifier
  - Naïve Bayes
  - Logistic regression
  - Support-vector machines
  - k-Nearest Neighbors
  - …

The bag of words (BOW)

$\gamma($

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.

$)=c$

BOW: important words

$\gamma($

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.

$) = c$

BOW: using a subset of words

γ(

```
x love xxxxxxxxxxxxxxxx sweet
xxxxxxx satirical xxxxxxxxxx
xxxxxxxxxxx great xxxxxxx
xxxxxxxxxxxxxxxxxxx fun  xxxx
xxxxxxxxxxxxx whimsical xxxx
romantic xxxx  laughing
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxx recommend xxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx several xxxxxxxxxxxxxxxxx
xxxxx  happy xxxxxxxxx again
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxx
```

)=c

BOW: word counts

$\gamma($

great	2
love	2
recommend	1
laugh	1
happy	1
...	...

$) = c$

Text Classification: Naïve Bayes

Naïve Bayes Intuition

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

Bayes' Rule applied to documents and classes

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

likelihood — prior

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

posterior

Bayes theorem

Naïve Bayes Classifier

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(c \mid d)$$

MAP is "maximum a posteriori" = most likely class

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(d \mid c) P(c)}{P(d)}$$

Bayes theorem

$$= \underset{c \in C}{\operatorname{argmax}} P(d \mid c) P(c)$$

Dropping the denominator

Naïve Bayes Classifier

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(d \mid c) P(c)$$

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

Document d represented as features x1..xn

Features could be:

- words (binary value)
- word counts
- word frequencies (tf)
- tf-idf

Naïve Bayes Classifier

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

$O(|X|^n \bullet |C|)$ parameters

How often does this class occur?

Could only be estimated if a very, very large number of training examples was available.

We can just count the relative frequencies in a corpus

Multinomial NB: Independence Assumptions

- **Bag of words assumption**: Assume that position of words doesn't matter (the **exchangeability** of random variables)

$$P(x_1, x_2, \ldots, x_n \mid c) = P(x_{\delta(1)}, x_{\delta(2)}, \ldots, x_{\delta(n)} \mid c)$$

- **Conditional Independence**: Assume the feature probabilities *P*(x_i| c_j) are independent given the class *c.*

$$P(x_1, \ldots, x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet P(x_3 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

Multinomial Naïve Bayes Classifier

Features: counts
Likelihood: $P(\boldsymbol{x}) = Multinomial(\boldsymbol{x})$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i \in V} P(x_i \mid c)$$

Boolean Multinomial Naïve Bayes

Features: binarized counts (0/1 values)
Likelihood: *P(**x**) = Multinomial(**x**)*

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i \in V} P(x_i \mid c)$$

- Boolean (aka binarized) multinomial NB good for polarity prediction
- Different from Bernoulli Naïve Bayes classifier

Bernoulli Naïve Bayes Classifier

Features: binarized counts (0/1 values)
Likelihood: *P(**x**) = MultivariateBernoulli(**x**)*

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}}\, P(c) \prod_{i \in V} \left(P(x_i \mid c) \right)^{x_i} \left(1 - P(x_i \mid c) \right)^{1 - x_i}$$

Learning parameters of Naïve Bayes

Learning parameters of Multinomial NB

- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}}$$

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

Parameter estimation

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

fraction of times word w_i appears among all words in documents of topic c_j

- Create mega-document for topic *j* by concatenating all docs in this topic
  - Use the frequency of *w* in mega-document

Problem with Maximum Likelihood

- What if we have seen no training documents with the word ***fantastic*** and classified in the topic **positive** (***thumbs-up)***?

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

- Zero probabilities cannot be conditioned away, no matter the other evidence!

$$c_{MAP} = \text{argmax}_c \, \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$

Laplace (add-1) smoothing for Naïve Bayes

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

Additive smoothing:

$$\hat{P}(w_i \mid c) = \frac{count(w_i, c) + \alpha}{\left(\sum_{w \in V} count(w, c) \right) + \alpha|V|}$$

Multinomial Naïve Bayes: Learning

- From training corpus, extract *Vocabulary*

- Calculate $P(c_j)$ terms
  - For each c_j in C do

    $docs_j \leftarrow$ all docs with class $= c_j$

$$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$

- Calculate $P(w_k \mid c_j)$ terms
  - $Text_j \leftarrow$ single doc containing all $docs_j$
  - For each word w_k in *Vocabulary*

    $n_k \leftarrow$ # of occurrences of w_k in $Text_j$

$$P(w_k \mid c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |Vocabulary|}$$

Summary: Naive Bayes surprisingly good

- Very Fast, low storage requirements
- Robust to Irrelevant Features

    Irrelevant Features cancel each other without affecting results

- Very good in domains with many equally important features

    Decision Trees suffer from *fragmentation* in such cases – especially if little data

- Optimal if the independence assumptions hold: If assumed independence is correct, then it is the Bayes Optimal Classifier for problem
- A good dependable baseline for text classification
    - **But we will see other classifiers that give better accuracy**

Naïve Bayes in Spam Filtering

- SpamAssassin Features:
  - Mentions Generic Viagra
  - Online Pharmacy
  - Mentions millions of (dollar) ((dollar) NN,NNN,NNN.NN)
  - Phrase: impress ... girl
  - From: starts with many numbers
  - Subject is all capitals
  - HTML has a low ratio of text to image area
  - One hundred percent guaranteed
  - Claims you can be removed from the list
  - 'Prestigious Non-Accredited Universities'
  - http://spamassassin.apache.org/tests_3_3_x.html

Evaluation metrics: precision, recall, accuracy,

The 2-by-2 contingency table

Actual class:

Predicted class:

	positive	negative
positive	tp	fp
negative	fn	tn

true negative

Precision and recall

- **Precision**: % of predicted positive items that are correctly classified $$P = \frac{tp}{tp + fp}$$
- **Recall**: % of actual positive items that are correctly classified $$R = \frac{tp}{tp + fn}$$
- **Accuracy**: % of all items that are correctly classified $$A = \frac{tp + tn}{n + p}$$

Actual class:

Predicted class:

	positive	**negative**
positive	tp	fp
negative	fn	tn

Confusion matrix c_{ij}

		Predicted		
		Cat	Dog	Rabbit
Actual class	Cat	5	3	0
	Dog	2	3	1
	Rabbit	0	2	11

$c_{33} = 11$
i.e., 11 rabbits correctly classified as rabbits

$c_{32} = 2$
i.e., 2 rabbits incorrectly classified as dogs

Per class evaluation measures

Recall:
Fraction of docs in class *i* classified correctly:

$$R = \frac{c_{ii}}{\sum_j c_{ij}}$$

		Predicted		
		Cat	Dog	Rabbit
Actual class	Cat	5	3	0
	Dog	2	3	1
	Rabbit	0	2	11

Precision:
Fraction of docs assigned class *i* that are actually about class *i*:

$$P = \frac{c_{ii}}{\sum_j c_{ji}}$$

		Predicted		
		Cat	Dog	Rabbit
Actual class	Cat	5	3	0
	Dog	2	3	1
	Rabbit	0	2	11

Accuracy: (1 - error rate)
Fraction of docs classified correctly:

$$Accuracy = \frac{\sum_i c_{ii}}{\sum_j \sum_i c_{ij}}$$

		Predicted		
		Cat	Dog	Rabbit
Actual class	Cat	5	3	0
	Dog	2	3	1
	Rabbit	0	2	11

A combined measure: F

- A combined measure that assesses the P/R tradeoff is F measure (weighted harmonic mean):

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}}$$

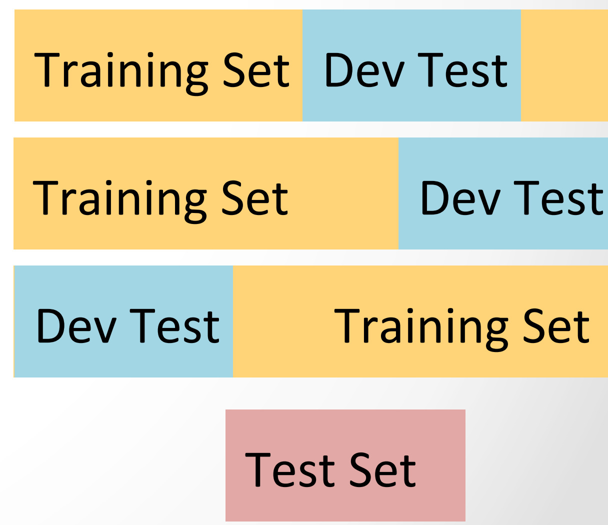
- The harmonic mean is a very conservative average
- We typically use balanced F measure with $\alpha = 0.5$
    - Namely, F1= $2PR/(P+R)$

Cross-validation

Data:

- Score 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
  - compute score metric for each split
  - average score over all splits
- Grid search over hyperparameters
  - repeat previous step for various values of hyperparameters to find the best ones
  - choose hyperparameters that maximize score

CV, 3 splits (folds):



Text Classification and Naïve Bayes

These slides are based on:

Dan Jurafsky and James H. Martin, Speech and Language Processing (3rd ed. draft)
https://web.stanford.edu/~jurafsky/slp3/
(Chapter 7)

These slides are an edited version of Jurafsky's slides:
https://web.stanford.edu/~jurafsky/NLPCourseraSlides.html