CORK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

Machine Learning

Machine Learning

Lecture: Bayesian Classification

Ted Scully

Contents

Document Classification

- Naive Bayes is a very successful and effective approach to learning to classify text documents.
- In document classification **each word is treated as an feature**.
- Document Classification
  - Spam Filtration
  - Author Identification
  - Sentiment Analysis (movie review, product reviews, important applications)

ML Workflow for Document Classification

Document Classification

- A Bayesian classifier will typically either adopt a **bag** of words or **set** of words approach.
  - (Bernoulli model) **Set of words**, counts the number of documents where a word occurs
  - (Multinomial Model) **Bag of words**, counts the total occurrences of a word across all documents.

- When classifying a test document, the Bernoulli model uses **binary occurrence** information, ignoring the number of occurrences of a word in a document , whereas the multinomial model keeps track of multiple occurrences in a single document.
- The models also differ in how **non-occurring terms** are used in classification. They do not impact the classification decision in the multinomial model; but in the Bernoulli model the probability of non-occurrence is factored in when computing probabilities

Calculating Prior Probabilities

$$c_{MAP} = argmax_{c \in C} \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

- The first thing we need to do is calculate the prior probabilities (that is, the probability of the class). This calculation is the same for both multinomial and binomial.

$$P(c) = \frac{\text{Number of documents of class c}}{Total\ Number\ of\ documents}$$

Naïve Bayes - Multinomial Model

$$c_{MAP} = argmax_{c \in C} \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

- Calculation of the probabilities in the multinomial model as are follows (notice we use laplace smoothing here):

  - $$P(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

count(w, c) is the number of occurrences of the word w in all documents of class c.

count(c) The total number of words in all documents of class c (**including duplicates**).

|V| The number of words in the vocabulary, which is all unique words irrespective of class.

Exercise

- The table below shows a very simple training set containing 4 documents and the words contained within those documents.
- It also contains the class of each of the document.

- Objective is to classify the new Test as either class Comp or class Politics.
  - We will use **laplace** for calculating the Multinomial probabilities
  - We will use simple **+1 smoothing** for calculating the Bernoulli probabilities

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

$$c_{MAP} = argmax_{c \in C} \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

$$P(Comp) = \frac{3}{4}$$

$$P(Politics) = \frac{1}{4}$$

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

$$c_{MAP} = argmax_{c \in C} \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software Java	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

$$P(w \mid c) = \frac{count(w, c) + 1}{count(c) + |V|}$$

Notice we use Laplace smoothing here

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software Java	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

$$P(w \mid c) = \frac{count(w,c) + 1}{count(c) + |V|}$$

Notice we use Laplace smoothing here

$$P(Cloud \mid Comp) = \frac{5+1}{9+6}$$

$$P(Java \mid Comp) = \frac{2+1}{9+6}$$

$$P(Software \mid Comp) = \frac{1+1}{9+6}$$

$$P(Spring \mid Comp) = \frac{1+1}{9+6}$$

$$P(Referendum \mid Comp) = \frac{0+1}{9+6}$$

$$P(Election \mid Comp) = \frac{0+1}{9+6}$$

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software Java	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

$$P(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

Notice we use Laplace smoothing here

$$P(Cloud \mid Politics) = \frac{0+1}{3+6}$$

$$P(Java \mid Politics) = \frac{0+1}{3+6}$$

$$P(Software \mid Politics) = \frac{1+1}{3+6}$$

$$P(Spring \mid Politics) = \frac{0+1}{3+6}$$

$$P(Referendum \mid Politics) = \frac{1+1}{3+6}$$

$$P(Election \mid Politics) = \frac{1+1}{3+6}$$

	Doc	Words	Class
Test	5	Java Software Java Election	?

$$P(Cloud \mid Comp) = \frac{6}{15}$$

$$P(Java \mid Comp) = \frac{3}{15}$$

$$P(Software \mid Comp) = \frac{2}{15}$$

$$P(Spring \mid Comp) = \frac{2}{15}$$

$$P(Election \mid Comp) = \frac{1}{15}$$

$$P(Referendum \mid Comp) = \frac{1}{15}$$

$$P(Cloud \mid Politics) = \frac{1}{9}$$

$$P(Java \mid Politics) = \frac{1}{9}$$

$$P(Software \mid Politics) = \frac{2}{9}$$

$$P(Spring \mid Politics) = \frac{1}{9}$$

$$P(Election \mid Politics) = \frac{2}{9}$$

$$P(Referendum \mid Politics) = \frac{2}{9}$$

$$P(Comp) = \frac{3}{4} \qquad P(Politics) = \frac{1}{4}$$

	Doc	Words	Class
Test	5	Java Software Java Election	?

$$c_{MAP} = argmax_{c \in C} \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

	Doc	Words	Class
Test	5	Java Software Java Election	?

$$P(c \mid W) = \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

$P(Comp \mid Test) =$ log(3/4) + log(3/15) + log(2/15) + log (3/15) + log (1/15) = **-3.57**

$P(Politics \mid Test) =$ log(1/4) + log(1/9) + log(2/9)+log(1/9)+log(2/9)= **-3.81**

Classify the document as being of class Comp

Naïve Bayes:
Text Classification for Multinomial

Examples are a set of training documents.

V is the set of classes (ex. Spam / NotSpam)

Learn_naive_Bayes_text($Examples$, V)

1. collect all words that occur in $Examples$
   $Vocabulary \leftarrow$ all distinct words in $Examples$
2. calculate the required $P(v_j)$ and $P(w_k|v_j)$ probability terms
   For each target value v_j in V do
   - $docs_j \leftarrow$ subset of $Examples$ for which the target value is v_j
   - $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
   - $Text_j \leftarrow$ a single document created by concatenating all members of $docs_j$
   - $n \leftarrow$ total number of words in $Text_j$ (counting duplicate words multiple times)
   - for each word w_k in $Vocabulary$
     - $n_k \leftarrow$ number of times word w_k occurs in $Text_j$
     - $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$

Document Classification

- Classify_naive_Bayes_text(newDoc)
  - We take in an unseen document *newDoc*, we extract all words from the document and store in *allWords* (the same word may appear multiple time)
  - Return V_{NB}, where:

$$V_{NB} = \underset{v_j \in V}{\mathrm{arg}max} \quad logP(v_j) + \sum_{x \in allWords} \log P(x \mid v_j)$$

Naïve Bayes - Bernoulli Model

$$c_{MAP} = argmax_{c \in C} \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

- Calculation of the probabilities in the Bernoulli model as are follows (notice we use plus one smoothing here):

- $$P(w \mid c) = \frac{countDocs(w,c)+1}{countDocs(c)+2}$$

countDocs(w, c) is the number of documents of class c where the word w occurs.

countDocs(c) The total number of documents of class c.

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software Java	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

In this example we will use +1 smoothing.

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software Java	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

Notice we use +1 smoothing here

$$P(Cloud \mid Comp) = \frac{3+1}{3+2}$$

$$P(Java \mid Comp) = \frac{2+1}{3+2}$$

$$P(Software \mid Comp) = \frac{1+1}{3+2}$$

$$P(Spring \mid Comp) = \frac{1+1}{3+2}$$

$$P(Referendum \mid Comp) = \frac{0+1}{3+2}$$

$$P(Election \mid Comp) = \frac{0+1}{3+2}$$

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software Java	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

$$P(Cloud \mid Politics) = \frac{0+1}{1+2}$$

$$P(Java \mid Politics) = \frac{0+1}{1+2}$$

$$P(Software \mid Politics) = \frac{1+1}{1+2}$$

$$P(Spring \mid Politics) = \frac{0+1}{1+2}$$

Notice we use +1 smoothing here

$$P(Referendum \mid Politics) = \frac{1+1}{1+2}$$

$$P(Election \mid Politics) = \frac{1+1}{1+2}$$

	Doc	Words	Class
Test	5	Java Software Java Election	?

$$P(Cloud \mid Comp) = \frac{4}{5}$$

$$P(Java \mid Comp) = \frac{3}{5}$$

$$P(Software \mid Comp) = \frac{2}{5}$$

$$P(Spring \mid Comp) = \frac{2}{5}$$

$$P(Election \mid Comp) = \frac{1}{5}$$

$$P(Referendum \mid Comp) = \frac{1}{5}$$

$$P(Cloud \mid Politics) = \frac{1}{3}$$

$$P(Java \mid Politics) = \frac{1}{3}$$

$$P(Software \mid Politics) = \frac{2}{3}$$

$$P(Spring \mid Politics) = \frac{1}{3}$$

$$P(Election \mid Politics) = \frac{2}{3}$$

$$P(Referendum \mid Politics) = \frac{2}{3}$$

	Doc	Words	Class
Test	5	Java Software Java Election	?

$$P(c \mid W) = \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

When classifying a new document in Bernoulli we go through every word in the vocabulary and we incorporate the probability of the word occurring and the word not occurring given the class.

The probability of a word occurring given the class is $P(w \mid c)$. Note that the probability of a word w not occurring given the class c is $1 - P(w \mid c)$

	Doc	Words	Class
Test	5	Java Software Java Election	?

$$P(c \mid W) = \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

$P(Comp \mid Test)$ = log(3/4) +

$P(Politics \mid Test)$ = log(1/4) +

	Doc	Words	Class
Test	5	Java Software Java Election	?

$$P(c \mid W) = \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

$P(Comp \mid Test)$ = log(3/4) + log(1-(4/5))+log(3/5)+log(2/5)+log(1-(2/5))+log(1/5)+log(1-(1/5)) = -2.46

$P(Politics \mid Test)$ = log(1/4) + log(1-(1/3))+log(1/3)+log(2/3)+log(1-(1/3))+log(2/3)+log(1-(2/3)) = -2.26

Classify the document as being of class Politics

Bernoulli v's Multinomial Model

- Empirical evaluations tend to report that the multinomial model typically outperforms the Bernoulli model as the **vocabulary size** increases and when used in classifying **large documents**.

- Please note that this is not always the case and it can be dependent on the data you use and the appropriate choice of features (pre-processing steps such as stop word removal etc.).