Probabilistic Classification

Naïve Bayes

Machine Learning II

Master in Business Analytics and Big Data

acastellanos@faculty.ie.edu

D. Hume

"We can rely only in what we learn from experience."

Rev. T. Bayes

"We modify our opinions with objective information: Initial Beliefs + Recent Data = A new and improved belief."

P. S. Laplace

"The probability of an hypothesis equals our initial estimate of its probability, times the probability of each new piece of information, divided by the sum of all probabilities of the data in all possible hypothesis ."

Independent Events

I usually run 3 days a week.
When I listen to music, I select rock music 4 out of every 5 times.

P(*running today*) = 0.42; # 3 days / 7 days = 0.42

P(*listens to rock*) = 0.8; ***Chain rule***

P(*listens to rock*, *running today*) = P(*running*) · P(*listens to rock* | *running*)

P(*listens to rock* | *running*) = P(*listens to rock*) …as they're independent

P(*listens to rock*, *running today*) = *0.8 · 0.42 = 0.34* ***Joint Probability***

Dependent Events

20% of the times I listen to music, I select Metallica.
But, if I'm listening to Rock music, the chance that I selected Metallica is 50%

P(*listens to rock*) = 0.8

P(*listens to Metallica*) = 0.2

$\boxed{\text{P(}\textit{listens to Metallica}\text{ | }\textit{listens to rock}\text{) = 0.5}}$

P(*rock*, *Metallica*) = P(*rock*) · P(*Metallica* | *rock*) = 0.8 · 0.5 = 0.4 **Dependent**

I need to know the dependent probability to compute the joint probability.

Bayes Theorem

"The probability of an hypothesis equals our initial estimate of its probability, times the probability of each new piece of information, divided by the sum of all probabilities of the data in all possible hypothesis ."

Likelihood
How probable is the evidence given that our hypothesis is true?

Prior
How probable was our hypothesis before observing the evidence?

$$P(H \mid e) = \frac{P(e \mid H)\,P(H)}{P(e)}$$

Posterior
How probable is our hypothesis given the observed evidence?

Marginal
How probable is the new evidence under all possible hypotheses?

Bayes Theorem

Likelihood
How probable is the evidence given that our hypothesis is true?

Prior
How probable was our hypothesis before observing the evidence?

$$P(H \mid e) = \frac{P(e \mid H)\, P(H)}{P(e)}$$

Posterior
How probable is our hypothesis given the observed evidence?

Marginal
How probable is the new evidence under all possible hypotheses?

Think Bayes!

You make a medical test and the result is **positive** for a rare disease, which is present only in **0,3% of the population**

The medical test is 99% effective

What is the probability that you have that rare disease?

Think Bayes!

Frequentist approach

99% because is the accuracy of the test

Bayesian approach

$$P(Disease|positive) = \frac{P(positive|Disease) \cdot P(Disease)}{P(postive)}$$

Think Bayes!

$$P(Disease|positive) = \frac{P(positive|Disease) \cdot P(Disease)}{P(postive)}$$

$P(Disease)$: This is the probability the disease, that is 0,003 (0,3%)

$P(positive|Disease)$: This is related to the effectiveness of our test: 0,99

$P(postive)$: The probability of the test giving a positive result. It includes when the test is correct (the test is positive and I have the disease) and when it's wrong.

$$P(postive) = P(TP) + P(FP) = 0{,}99 * 0{,}003 + 0{,}1 * 0{,}997 = 0{,}0129$$

$$P(Disease|positive) = \frac{0{,}99 \cdot 0{,}003}{0{,}0129} = 0{,}23\ (23\%)$$

Naïve Bayes Classification

- "Naïve" comes from "too simple to be true"
- Suppose we want to build a text classifier based on Naïve Bayes:
  - The text is made of a series of words $(x_1,\ldots,x_n)$
  - The classifier must be able to distinguish between two classes k_1 and k_2, for a given text X.
  - The text will be assigned to the class for which:

$$\text{MAP} = \max(\, P\,(k_1 \mid x_1, \ldots, x_n), P\,(k_2 \mid x_1, \ldots, x_n)\,)$$

This is called the "**Maximum A Posteriori**" (MAP) probability.

Classification using Bayes Theorem

$$P(H \mid e) = \frac{P(e \mid H)\, P(H)}{P(e)}$$

Given a document and the words present on it ($x_1,\ldots,x_n$), we can use Bayes to compute the probability that it belongs to class k_1

$$P(k_1 | x_1, \ldots, x_n) = \frac{P(x_1, \ldots, x_n | k_1)\, P(k_1)}{P(x_1, \ldots, x_n)}$$

Classification using Bayes Theorem

$$P(k_1|x_1,\dots,x_n) = \frac{P(x_1,\dots,x_n|k_1)\,\boldsymbol{P(k_1)}}{P(x_1,\dots,x_n)}$$

Classification using Bayes Theorem

k_1

k_2

$\mathrm{P}(k_1) = 5/8 = 0.625$

$\mathrm{P}(k_2) = 3/8 = 0.375$

Classification using Bayes Theorem

$$P(k_1|x_1,\ldots,x_n) = \frac{\boldsymbol{P(x_1,\ldots,x_n|k_1)}\,P(k_1)}{P(x_1,\ldots,x_n)}$$

Naïve Bayes assumption

Assumption: the probabilities of words appearing in the text are **independent** from one another:

$$P(\boldsymbol{x_1}, \ldots, \boldsymbol{x_n} \mid k_1) =$$

$$P(x_1, \ldots, x_n) | P(k_1) =$$

$$(P(x_1)P(x_2) \ldots, P(x_n)) | \, P(k_1) =$$

$$\boldsymbol{P(x_1 \mid k_1) \cdot \cdots \cdot P(x_n \mid k_1)}$$

Naïve Bayes assumption

Assumption: the probabilities of words appearing in the text are **independent** from one another:

$$P(x_1, \dots, x_n \mid k_1) = P(x_1 \mid k_1) \cdot \dots \cdot P(x_n \mid k_1)$$

This is a frequency count problem, very easy to solve

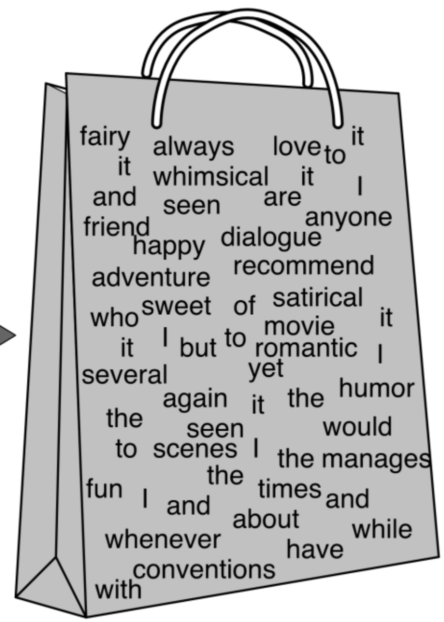
$$P(x_i|k_j) = \frac{count(x_i, k_j)}{\sum count(x, k_j)}$$

fraction of times word x_i appears
among all word counts in documents of topic k_j

Representation of *documents*: Bag of Words

k_1

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!



$$P(it|k_1) = \frac{6}{32} = 0{,}1875$$

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

$$P(fairy|k_1) = \frac{1}{32} = 0{,}03125$$

Classification using Bayes Theorem

$$P(k_1|x_1,\dots,x_n) = \frac{P(x_1,\dots,x_n|k_1)\,P(k_1)}{\boldsymbol{P(x_1,\dots,x_n)}}$$

Naïve Bayes assumption

$$\left.\begin{aligned} P(k_1|x_1,\dots,x_n) &= \frac{P(x_1,\dots,x_n|k_1)\,P(k_1)}{P(x_1,\dots,x_n)} \\ P(k_2|x_1,\dots,x_n) &= \frac{P(x_1,\dots,x_n|k_2)\,P(k_2)}{P(x_1,\dots,x_n)} \end{aligned}\right\}$$

Same denominator, so, it can be **removed**

$$MAP = \max\{\, P(x_1,\dots,x_n|k_1)\cdot P(k_1), P(x_1,\dots,x_n|k_2)\cdot P(k_2)\,\}$$

Naïve Bayes: implementation

To decide (CLASSIFY) which class (among two, in this example) a text belongs to, we must compute

$$\max\left(\underbrace{P(k_1)P(x_1|k_1)\cdot\ \ldots\ \cdot P(x_n|k_1)}_{\text{Probability of these events belong to class } k_1},\ \underbrace{P(k_2)P(x_1|k_2)\cdot\ \ldots\ \cdot P(x_n|k_2)}_{\text{Probability of these events belong to class } k_2}\right)$$

And that's it! No training, etc., just compute probabilities and classify

Problems

1) What happens with words **not** seen before?
2) The product of small probabilities result in really small values (underflow)

$$\max\left(P(k_1)P(x_1 \mid k_1)\cdot \ldots \cdot P(x_n \mid k_1),\ P(k_2)P(x_1 \mid k_2)\cdot \ldots \cdot P(x_n \mid k_2)\right)$$

Add-one Simplification

- Maximum likelihood estimate is based on

$$P(k_j) = \frac{doccount(k_j)}{N_{docs}}$$

Add-one approach to avoid zeroed probabilities

$$P(x_i|k_j) = \frac{count(x_i, k_j)}{\sum count(x, k_j)} \sim \frac{count(x_i, k_j) + 1}{\sum count(x, k_j) + |X|}$$

Laplace Smoothing

Laplace smoothing, which computes the **sum of logs** to avoid underflow errors (probabilities with extremely low values).

$$\max \left[\underbrace{\log P(k_1) + \sum_{i=1}^{n} \log P(x_i|k_1)}, \underbrace{\log P(k_2) + \sum_{i=1}^{n} \log P(x_i|k_2)} \right]$$

Example

Mandrill App

appWords.txt (~500Kb)

```
[blog] using nullmailer and mandrill for your ubuntu linux server outboud mail  http://bit.ly/zjhok7  #plone
[blog] using postfix and free mandrill email service for smtp on ubuntu linux server  http://bit.ly/11hmdzz  #plone
@aalbertson there are several reasons emails go to spam mind submitting a request at http://help.mandrill.com  with additional details
@adrienneleigh i just switched it over to mandrill  let's see if that improve the speed at which the emails are sent.
@ankeshk +1 to @mailchimp we use mailchimp for marketing emails and their mandrill app for txn emails.. @sampad @abhijeetmk @hiway
@biggoldring that error may occur if unsupported auth method used can you email us via http://help.mandrill.com  so we can get details
@bluehayes mind sending us some details about your account via http://help.mandrill.com    things look correct here but we may need some detail
@cemsisman it can vary  but if sending really low volumes  may not be worth it can offer detail – submit request at http://help.mandrill.com
@compactcode have you checked out mandrill (@mandrillapp)  it's a transactional email service that runs .. https://longreply.com/r/66c91ea4
```

otherWords.txt (~500Kb)

```
¿en donde esta su remontada mandrill
.@katie_phd alternate  'reproachful mandrill' cover of @davidquammen's spillover
.@theophani can i get "drill" in there  it would be a picture of a mandrill holding a drill somethin.
“@chrisjboyland baby mandrill paignton zoo 29th april 2013 http://youtu.be/qpjoffylxgg a via @youtube” this is just so cute
“@missmya #nameanamazingband mandrill ” mint condition  maroon 5  the fray.
“fat city strut” by mandrill is my new jam http://t.thisismyjam.com/siftdigital/_5jl4tyj …
【soul train #22】1973年 mandrill 中古盤屋でインパクトのあるジャケットを良く見かけたが音を聴いたのはこの番組（95年）が初めて。少しチカーノが入
@alicegreennn_ but how come you didn't have mandrill
```

test.csv

2 columns, **20** rows

2 KB, Comma-Separated, UTF-8

APP	just love @mandrillapp transactional email service - http://mandrill.com sorry @sendgrid and @mailjet #timetomoveon
APP	@rossdeane mind submitting a request at http://help.mandrill.com with account details if you haven't already glad to take a look
APP	@veroapp any chance you'll be adding mandrill support to vero
APP	@elie__ @camj59 jparle de relai smtp 1 million de mail chez mandrill / mois comparé à 1 million sur lite sendgrid y a pas photo avec mailjet
APP	would like to send emails for welcome password resets payment notifications etc what should i use was looking at mailgun/mandrill
APP	from coworker about using mandrill i would entrust email handling to a pokemon.
APP	@mandrill realised i did that about 5 seconds after hitting send
APP	holy shit it's here http://www.mandrill.com/
APP	our new subscriber profile page activity timeline aggregate engagement stats and mandrill integratio #bjcbranding http://bit.ly/13wau5c
APP	@mandrillapp increases scalability (http://bit.ly/14myvuh) then decreases pricing (http://bit.ly/13uja7s) #selfinducedcannibalization
OTHER	the beets rt @missmya #nameanamazingband mandrill
OTHER	rt @luissand0val fernando vargas mandrill mexican pride mma
OTHER	photo oculi-ds mandrill by natalie manuel http://tmblr.co/zjqanxhdswlr
OTHER	@mandrill me neither we can be :sadpanda together :(
OTHER	@mandrill n / (k * (n - k)) where n = 5 and k = 4 it has been a long time but i think that is it
OTHER	megaman x - spark mandrill acapella http://youtu.be/hyx9-kwyjdi @youtubeさんから
OTHER	@angeluserrare1 storm eagle ftw nomás no dejes que se le acerque spark mandrill xd
OTHER	gostei de um vídeo @youtube http://youtu.be/xzny7zimtni aspark ... mandrill's stage on guitar (mega man x)
OTHER	what is 2-year-old mandrill jj thinking in this pic http://ow.ly/jfrqf re-tweet with your caption.

Frequency count

Spark/Scala

```
val appWordsFile = sc.textFile("appWords.txt")
val othWordsFile = sc.textFile("otherWords.txt")

val toRemove = "\"\'.;_:,´^)(*+-\\/*][?&%!".toSet
val appWords = appWordsFile.flatMap(_.split(' ')).map(
_.filterNot(toRemove) )
val appCount = appWords.map(w => (w,1)).reduceByKey(_ + _).sortBy( _._1 )

val otherWords = othWordsFile.flatMap(_.split(' ')).map(
_.filterNot(toRemove) )
val otherCount = otherWords.map(word => (word,1)).reduceByKey(_ +
_).sortBy( _._1 )
```

2 columns, **500+** rows

9 KB, Comma-Separated, UTF-8

#atl	30
#atlanta	30
#bjcbranding	30

R

```
appFile   <- read.csv("appFreqs.csv", header=F)
otherFile <- read.csv("otherFreqs.csv", header=F)
appTotal <- sum(appFile$V2)
otherTotal <- sum(otherFile$V2)
appFreqs <- cbind(appFile, freq=log((appFile$V2/appTotal)))
otherFreqs <- cbind(otherFile,
freq=log((otherFile$V2/otherTotal)))
```

$\log P(x_i|k_1)$

appFreqs.csv

	V1	V2	freq
1	#atl	30	-7.705262
2	#atlanta	30	-7.705262
3	#bjcbranding	30	-7.705262

This function gives me a word frequency in the data frame

```r
freq  <- function(word, frame) {
  val <- frame[which(frame$V1 == word),]$freq
  if(length(val) == 0) 1/log(sum(frame$V2))
  else val
}
```

Returns the `freq` column, which is the $\log P(x_i|k_1)$.

This computes the prior probability of each class

```r
appPrior = log(length(appFile) / (length(appFile) + length(otherFile))) # Number of Tweets in app File / Total nu
mber of Tweets (sum of the number of tweets in both files)
otherPrior = log(length(otherFile) / (length(appFile) + length(otherFile))) # Number of Tweets in other File / To
tal number of Tweets (sum of the number of tweets in both files)
```

Naïve Bayes, working!

```
for(j in 1:nrow(test))
{
  tweet <- as.character(test[j,2])                # Extract the content of the tweet
  wordsInThisTweet <- strsplit(tweet, " ")[[1]]   # Extract the words into a list.
  appProb = as.double(0.0)
  otherProb = as.double(0.0)

  # For every word in this tweet, sum its frequency value.
  for(word in wordsInThisTweet) {
    appProb   <- sum(appProb,   freq(as.character(word), appFreqs))
    otherProb <- sum(otherProb, freq(as.character(word), otherFreqs))
  }

  posteriorAppPob = appProb * appPrior
  posteriorOtherPob = otherProb * otherPrior

  # Categorize according to the score obtained from every subset (App tweets, and Other tweets)
  if(posteriorAppPob > posteriorOtherPob) {
    pred[j] <- "APP"
  } else {
    pred[j] <- "OTHER"
  }
}
```

Correct matching rate: 85% !

Model Evaluation

- Use confusion matrix, as usual in classification problems
- Evaluate with Precision & Recall,

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

$$Precission = \frac{TP}{TP+FP} \qquad Recall = \frac{TP}{TP+FN}$$

- or F1-Score.

$$F1 = \frac{2 \cdot precission \cdot recall}{precission + recall}$$

Model evaluation: More than 2 classes

- TRAINING
  - For each class c
    - Build a different classifier to distinguish c from all other classes, c'
- TEST
  - Given test document d,
    - Evaluate it for membership in each class using each classifier.
    - d belongs to the one class with maximum score
- EVALUATION
  - For each pair of classes $<c_1, c_2>$ how many documents from c_1 were incorrectly assigned to c_2?

Multiclass performance metrics

- **Recall:**
  - Fraction of docs in class i classified correctly $\quad \frac{c_{i,i}}{\sum_j c_{i,j}}$
- **Precision:**
  - Fraction of docs assigned class i that are actually about class i $\quad \frac{c_{i,i}}{\sum_j c_{j,i}}$
- **Accuracy: (1 - error rate)**
  - Fraction of docs classified correctly $\quad \frac{\sum_i c_{i,i}}{\sum_j \sum_i c_{i,j}}$

Micro- vs. Macro-Averaging

- In **multi-class Bayes**, the performance of the different classifiers obtained is combined using:
  - **Macro**averaging: Compute performance for each class, then average

$$(0.5 + 0.9)/2 = 0.7$$

  - **Micro**averaging: Collect decisions for all classes, compute contingency table, evaluate

$$100/120 = 0.83$$

Class 1

	Truth: yes	Truth: no
Classifier: yes	10	10
Classifier: no	10	970

Class 2

	Truth: yes	Truth: no
Classifier: yes	90	10
Classifier: no	10	890

Micro Ave. Table

	Truth: yes	Truth: no
Classifier: yes	100	20
Classifier: no	20	1860

Summary for Naïve Bayes

- **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.
- Optimal if the **independence assumptions** hold:
  - If assumed independence of events is correct, then Bayes is the optimal classifier for problem
  - If not, **remember**: "All models are wrong but some are useful".
- Try more complicated models if it does not perform well