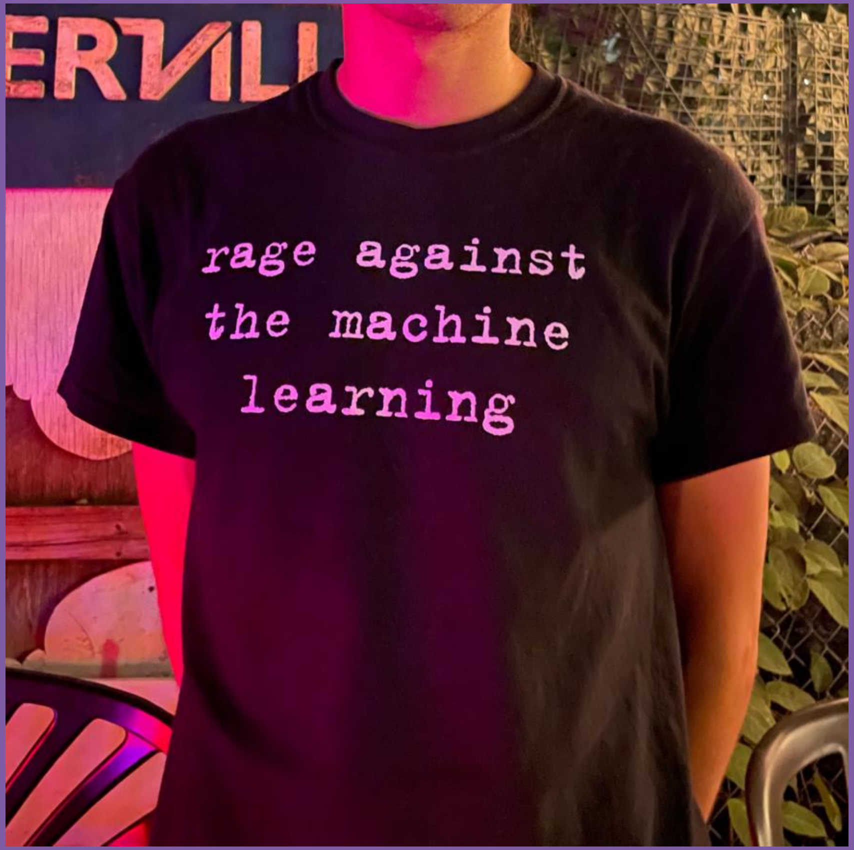
FH CAMPUS WIEN
UNIVERSITY OF APPLIED SCIENCES

Naïve Bayes (cont.)
K-NN
K-Means

LIAD MAGEN

How was the LDA assignment?



FH CAMPUS WIEN
UNIVERSITY OF APPLIED SCIENCES

Agenda

- Recap:
  - Linear Regression
  - Logistic Regression
- Naïve Bayes (continued)

- K-NN vs. K-Means

Recap

- > What is bag-of-words?
- > What is TF/IDF?
  What does it do?
  What is the logical thought behind it?

- > Can I use **Linear Regression** to detect if an email is spam?
- > Can I use **Logistic Regression** to determine the correct translation of the English preposition “in” to French (dans, en, à, au bout de, …)
  - > To which canonical learning type(s) does this problem belong?
  - > Which activation function is used?

Naïve Bayes

> The Naive Bayes classifier is a simple but surprisingly effective probabilistic text classifier that builds on Bayes’ rule.

> It is called ‘naive’ because it makes strong (unrealistic) independence assumptions about probabilities.

> It uses a representation of texts as bags of words, that is, it does not pay attention to word order.

Naive Bayes classification rule, informally

Nigerian
Prince
Inheritance

?

Spam

Nigerian
Prince
Inheritance

Score(spam) =
P(spam) P(Nigerian | spam) P(Prince | spam)
P(Inheritance | spam)

70%

Not Spam

Nigerian
Prince
Inheritance

Score(not spam) =
P(not spam) P(Nigerian | not spam)
P(Prince | not spam) P(Inheritance | not spam)

30%

Naive Bayes classification rule, informally

Nigerian
Prince
Inheritance

Spam

Not Spam

Nigerian
Prince
Inheritance

Score(spam) =
P(spam) P(Nigerian | spam) P(Prince | spam) P(Inheritance | spam)

70%

Nigerian
Prince
Inheritance

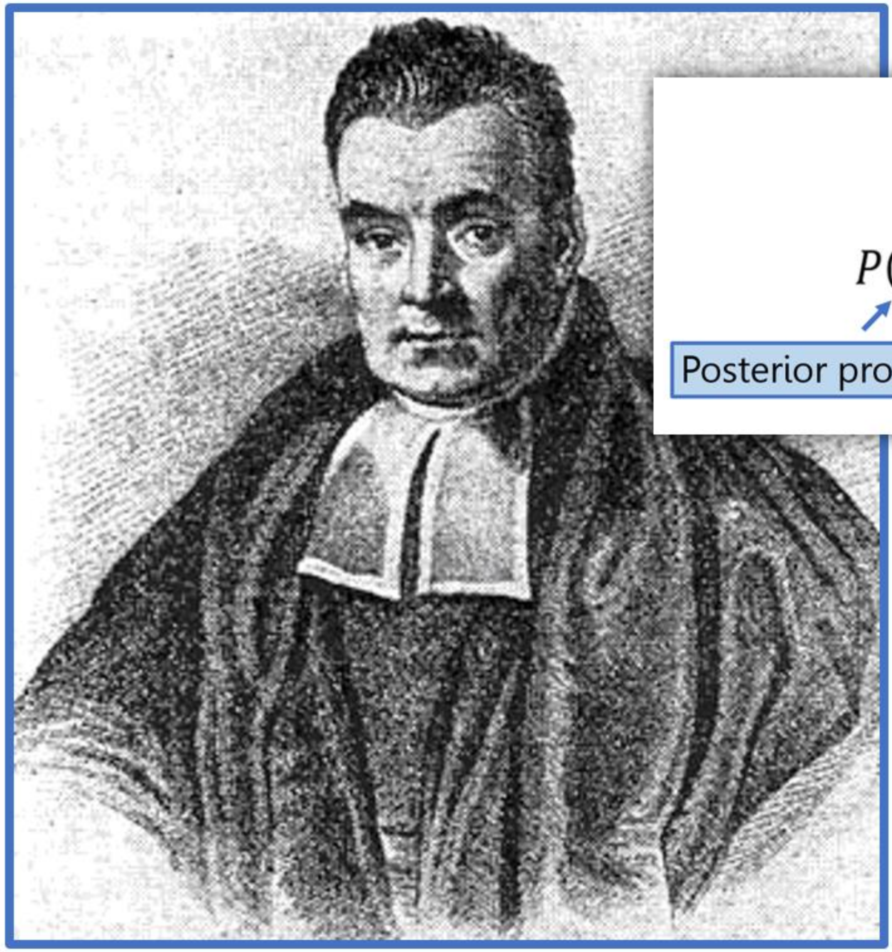
Score(not spam) =
P(not spam) P(Nigerian | not spam) P(Prince | not spam) P(Inheritance | not spam)

30%

Bayes' Rule

- For classification, we would like to know $P(\text{class} \mid \text{document})$.
  $P(\text{spam} \mid \text{email-words})$
- But a Naive Bayes classifier contains $P(\text{document} \mid \text{class})$.
  $P(\text{words} \mid \text{spam})$
- The classifier uses Bayes' rule to convert between the two.
  $P(\text{spam} \mid \text{email-words}) \propto P(\text{class})\, P(\text{document} \mid \text{class})$

Bayes' Rule



Likelihood

Prior probability

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Posterior probability

Evidence

Formal Definition of Bayes Rule

C a set of possible classes

V a set of possible words; the model's **vocabulary**

$P(c)$ probabilities that specify how likely it is for a document to belong to class c (one probability for each class)

$P(w|c)$ probabilities that specify how likely it is for a document to contain the word w, given that the document belongs to class c (one probability for each class–word pair)

Classification using Naïve Bayes Rule

argmax → Choose the class c that maximizes the probability:

$$\hat{c} = \arg\max P(c) = \arg\max P(doc|class)P(class)$$

$P(doc|class)$→ calculating the probability for every word in our vocabulary $w \in V$ and multiplying all the probabilities:

$$\hat{c} = \arg \max_{c \in Classes} P(c) \prod_{w \in V} P(w_i|c)$$

FH CAMPUS WIEN
UNIVERSITY OF APPLIED SCIENCES

Classification using Naïve Bayes Rule - Problems

> If the vocabulary is large – looping on all of it takes too long.

Solution: loop only over the document words

> What if we encounter an out-of-vocabulary (ooV) word?

Solution: skip unknown words

> Multiplying small floats yields a too small number for calculations

Solution: Use Log Probabilities

Math Recap: Log rules

$$\log_b(MN) = \log_b(M) + \log_b(N)$$

$$\log_b\left(\frac{M}{N}\right) = \log_b(M) - \log_b(N)$$

$$\log_b(M^p) = p\log_b(M)$$

[Log rules: Justifying the logarithm properties (article) | Khan Academy]

Classification using Naïve Bayes – Log Probabilities

To avoid underflow when multiplying small numbers:

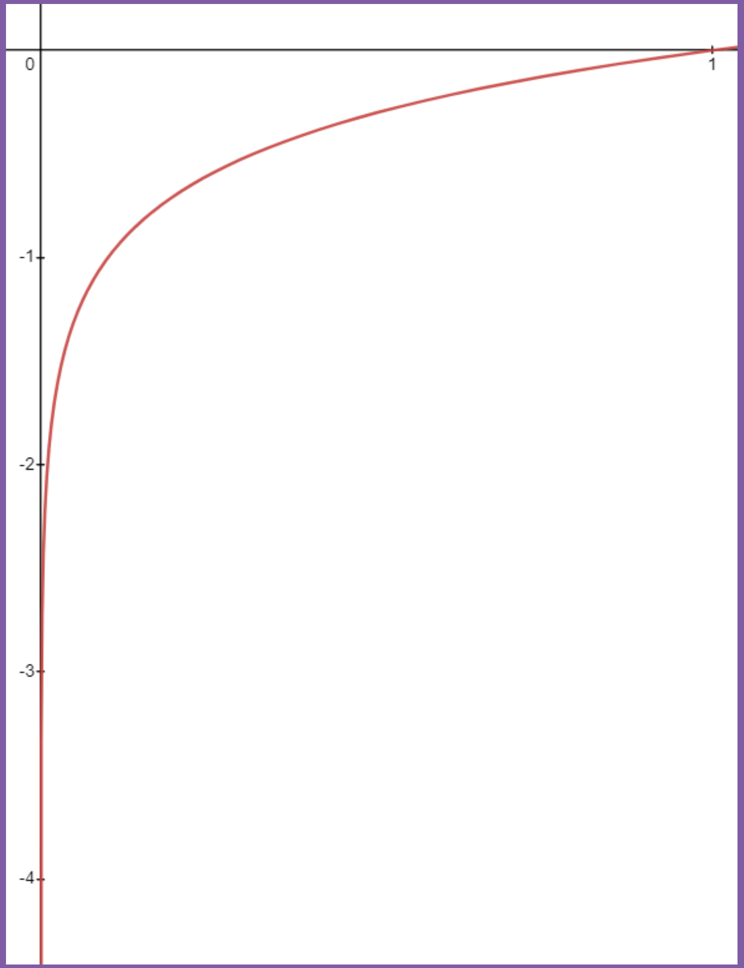
$$\hat{c} = \arg\max_{c \in Classes} P(c) \prod_{w \in V} P(w_i|c)$$

And to ease calculation and speed, we can take logs from both sides:

$$\log \hat{c} = argmax_{c \in Classes} \log P(c) + \sum_{w \in sentence} \log P(w_i|c)$$

Reminder: Log Probabilities

Log(x)



Training Naïve Bayes Classifier

> For training, we use the frequencies in our Training-Data

> $P(c) = \frac{|Doc \in C|}{|Docs|}$ ➔ The percentage of documents belong to class c

> $P(w_i|c) = \frac{|w_i \in C|}{\sum_{w \in V} |w \in c|}$ ➔ The percentage of the word occurrence in this class

This technique is called Maximum Likelihood Estimation (MLE)

Training Naïve Bayes Classifier – Unseen words

> If we encounter a word in a class, which was not there before: $P(w_i|c) = 0$

> But then the whole naïve bag-of-words multiplication will be… Zero

Solution: add-one smoothing (aka *Laplace Smoothing*):

$$\log P(w_i|c) = \log \frac{|w_i \in C| + 1}{\sum_{w \in V} |w \in c| + 1}$$

> "Hallucinating" an additional word occurrence

Training Naïve Bayes Classifier – Additive Smoothing

> Instead of adding 1 extra occurrence, we can also add any *k* extra occurrences: **Additive Smoothing**

> *k* can be any *positive* number – even fractions (0.0001)

> We can tune *k* during training

Advanced Smoothing Techniques

> Additive smoothing and add-one smoothing often works well in the context of text classification.

> In other contexts, more advanced smoothing techniques work considerably better than additive smoothing.

e.g., Witten–Bell smoothing, Kneser–Ney smoothing for ngram-models

Naïve Bayes - Generative

> The bayes formula:

$$\hat{c} = \arg\max P(c|d) = \arg\max P(d|c)P(c)$$

Likelihood Prior

If the prior is known (category/class is given - $P(c) = 1$), we can generate words based on the Likelihood: $P(d|c)$

Reminder:
Discriminative → $P(\mathbf{class}|document)$ or

Generative → $P(\mathbf{document}|class)$

FH CAMPUS WIEN
UNIVERSITY OF APPLIED SCIENCES

Naïve Bayes - Generative

On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes

Andrew Y. Ng
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720

Michael I. Jordan
C.S. Div. & Dept. of Stat.
University of California, Berkeley
Berkeley, CA 94720

nips01-discriminativegenerative.pdf (stanford.edu)

Naïve Bayes – Final Notes

> A linear & probabilistic classifier
> Implementation exists in NLTK & Scikit-Learn (SKL)
> On SKL one can choose the distribution. The best performance is normally the *Multinomial* Naive Bayes
A Comparison of Event Models for Naive Bayes Text Classification: binomial.dvi (washington.edu)

A Comparison of Event Models for Naive Bayes Text Classification

Andrew McCallum[‡†]
mccallum@justresearch.com
[‡]Just Research
4616 Henry Street
Pittsburgh, PA 15213

Kamal Nigam[†]
knigam@cs.cmu.edu

[†]School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Naïve Bayes – Final Notes

- Can act as a *discriminative* or *generative* model
- Still widely used in Linguistics for classification

- Good summary/deeper dive: The Optimality of Naive Bayes – sections: Abstract & Naive Bayes and Augmented Naive Bayes (you can safely ignore the augmented Naïve Bayes)

The Optimality of Naive Bayes

Harry Zhang
Faculty of Computer Science
University of New Brunswick
Fredericton, New Brunswick, Canada E3B 5A3
email: hzhang@unb.ca

Dive deeper

- > Bayes theorem, the geometry of changing beliefs – YouTube
- > 1.9. Naive Bayes — scikit-learn 1.1.3 documentation
- > 6. Learning to Classify Text (nltk.org)
- > Dan Jourafsky’s book – Chapter 4: https://web.stanford.edu/~jurafsky/slp3/4.pdf

FH CAMPUS WIEN
UNIVERSITY OF APPLIED SCIENCES

K-NN // K-Means

K-NN // K-Means – Class Assignment

- Assignment:
  - Two groups:
    - Group 1 - K-NN
    - Group 2 - K-Means

  - Read and learn your topic.
    - Use any source of information
    - Make use of the guiding questions

  - Prepare a presentation (~30min) to teach it to the other group
    - Add visual aids to illustrate your points
    - Make sure everyone gets to present an equal part
    - Be ready for Q&A