# COMP 472: Artificial Intelligence Machine Learning port 2 Naive Bayes Classification Application to Spam Filtering video #3

Russell & Norvig: Sections 12.2 to 12.6

# Today

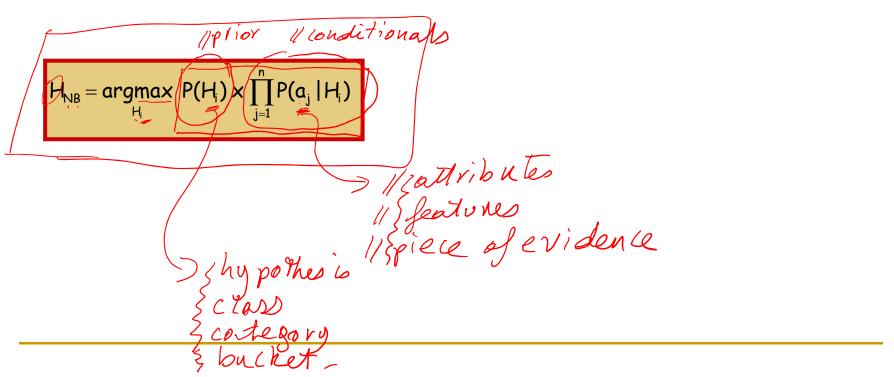
- Introduction to ML
- 2. Naive Bayes Classification
  - a. Application to Spam Filtering



- 3. Decision Trees
- 4. (Evaluation
- 5. Unsupervised Learning)
- 6. Neural Networks
  - a. Perceptrons
  - b. Multi Layered Neural Networks

### Recall

$$H_{NB} = \underset{H_{i}}{\operatorname{argmax}} \quad \frac{P(H_{i}) \times P(E \mid H_{i})}{P(E)} = \underset{H_{i}}{\operatorname{argmax}} \quad P(H_{i}) \times P(E \mid H_{i}) = \underset{H_{i}}{\operatorname{argmax}} \quad P(H_{i}) \times P(< a_{1}, a_{2}, a_{3}, ..., a_{n} > |H_{i}) = \underset{H_{i}}{\operatorname{argmax}} \quad P(H_{i}) \times \prod_{j=1}^{n} P(a_{j} \mid H_{i})$$

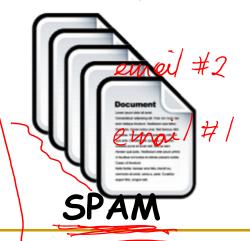


# Application of Naive Bayes Classification: Spam Filtering

- Task: classify e-mails (documents) into a pre-defined class.
  - ex: spam / ham
  - = ex: sports, recreation, politics, war, economy,... positive vautval

## Given

 training set of documents already classified into the correct category

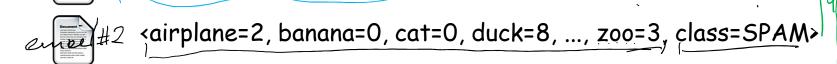




## e-mail Representation

- each e-mail is represented by a vector of feature/value>
  - 👝 feature = actual words in the e-mail
  - value = number of times that word appears in the e-mail

\(\frac{10000}{\text{words}}\) words \(\text{words}\) \(\text{airplane=0}\) banano=1, \(\text{cat=5}\), \(\text{duck=4}\), \(\text{duck=4}\), \(\text{vords}\)



<airplane=1, banana=1, cat=5, duck=8, ..., zoo=3, class=HAM>

<airplane=1, banana=3, cat=5, duck=0, ..., zoo=6, class=HAM>

Strictly speaking, what this is called a <u>Multinomial</u> Naïve Bayes classifier, because we use the <u>frequency</u> of words, as opposed to just using binary values for the presence/absence of words.

# Naïve Bayes Algorithm

```
// 1. training
                                                                 // conditionals
for all classes c_i // ex. ham or spam
      for all words w_i in the vocabulary
                       P(w_j \mid c_i) = \frac{count(w_j, c_i)}{\sum count(w_j, c_i)}
                                                                 1/ P (Hi) = 0.0
P (ham) 7 (spam) = 0.4
for all classes c;
     compute P(c_i) = \frac{count(documents in c_i)}{count}
                            count(all documents)
// 2. testing a new document D
for all classes c<sub>i</sub> // ex. ham or spam
       score(c_i) = P(c_i)
      for all words w_i in the D
            score(c_i) = score(c_i) \times P(w_i | c_i)
choose (c_i) = with the greatest score (c_i)
                        W<sub>1</sub>
                                  W<sub>2</sub>
                                             W<sub>3</sub>
                                                        W4
                                                                   W5
                                                                               W<sub>6</sub>
      c1: SPAM
                     p(w_1|c_1)
                               p(w_2|c_1)
                                          p(w_3|c_1)
                                                     p(w_4|c_1)
                                                                 p(w_5|c_1)
                                                                             p(w_6|c_1)
      c2: HAM
                     p(w_1|c_2)
                               p(w_2|c_2)
                                          p(w_3|c_2)
                                                     p(w_4|c_2)
                                                                p(w_5|c_2)
                                                                             p(w_6|c_2)
```

# Example 1

vocabulary = > best, book, cheap, sale, Trip, meds? p (speen) = 3/5

Dataset

c1: SPAM

doc1: "cheap meds for sale"

doc2: "click here for the best meds"

doc3: "book your trip"

doc4: "cheap book sale, not meds"

"here is the book for you"

HAM

Question:

👊 doc6: "the cheap book"

should it be classified as HAM or SPAM?







# Example 1

a word in an emeil is

#### Assume

vocabulary = {best, book, cheap, sale, trip, meds} 12=6 words

If not in vocabulary, ignore word

Training:

1. Training:

```
P(best|SPAM) = 1/7
P(book|SPAM) = 1/7
P(book|HAM) = 0/5
P(cheap|SPAM) = 1/7
P(cheap|HAM) = 1/5
P(sale|SPAM) = 1/7
P(trip|SPAM) = 1/7
P(trip|SPAM) = 1/7
P(meds|SPAM) = 1/7
P(meds|HAM) = 1/5
                                                                                                      P(HAM) = 2/5
```

priors 70 P(SPAM) = 3/5

2. Testing: "the cheap book" the

Score(HAM)=P(HAM) x P(cheap|HAM) x P(book|HAM) X O

Score(SPAM)=P(SPAM) x P(cheap|SPAM) x P(book|SPAM) X Y

## Be Careful: Smooth Probabilities

- normally:  $P(w_i | c_j) = \frac{(frequency of w_i in c_j)}{total number of words in c_i}$
- what if we have a  $P(w_i|c_i) = 0...?$ 
  - ex. the word "dumbo" never appeared in the class SPAM?
  - then P("dumbo" | SPAM) = 0
- so if a text contains the word "dumbo", the class SPAM is completely ruled out!
- to solve this: we assume that every word always appears at least once (or a smaller value) for smoothing
  - ex: add-1 smoothing:

(frequency of winc)+1 total number of words in citize of vocabulary

# Smoothing - add-1 smoothing

Assume: vocabulary V = {ball, heat, kitchen, referee, stove, the, |V| = 1<u>00</u> Training set: c1: COOKING c2: SPORTS original doc1: ... ball... heat... doc1: ... stove ... kitchen ... the ... heat doc2: ... the ... referee ... player ... doc2: ... kitchen... pasta... stove. doc<sub>100000</sub>: ¿... stove...heat... ball... doc<sub>75000</sub>: goal... injury ... ball heat kitchem 100 extre words 100 extra words new smoothed value for the conditional probs will be based on 1 + 1 T

10

## Be Careful: Use Logs

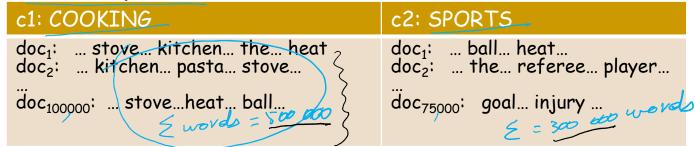
- if we really do the product of probabilities...
  - $\square$  argmax<sub>ci</sub>  $P(c_i) \prod P(w_i|c_i)$
  - we soon have numerical underflow...
  - $\Box$  ex:  $0.01 \times 0.02 \times 0.05 \times ...$
- so instead, we add the log of the probs | we kneet the seme the

  - = ex: log(0.01) + log(0.02) + log(0.05) + ...

     use the base (log) that you prefer

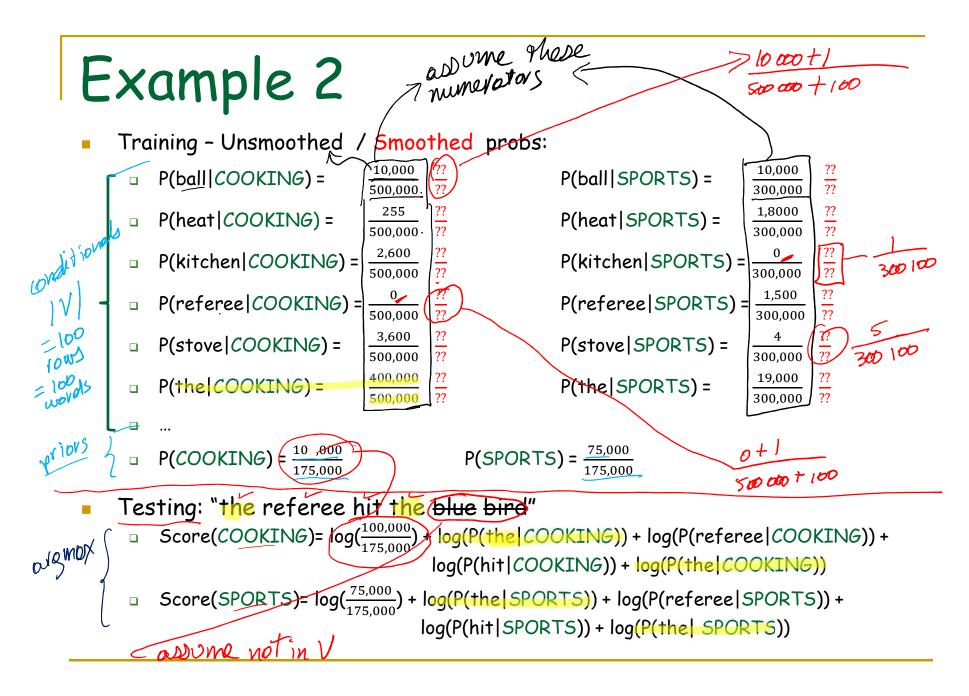
# Example 2

Training set:



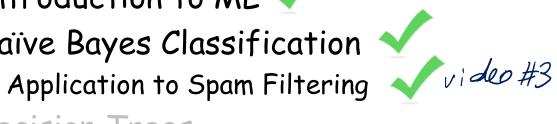
#### Assume:

- vocabulary V = {ball, heat, kitchen, referee, stove, the, ...}
- □ 500,000 words in Cooking
- □ 300,000 words in Sports
- □ 100,000 docs in Cooking —
- 75,000 docs in Sports



# Today

- Introduction to ML ~
- Naïve Bayes Classification



- 3. Decision Trees
- 4. (Evaluation
- Unsupervised Learning)
- Neural Networks
  - Perceptrons
  - Multi Layered Neural Networks

# Up Next

- 1. Introduction to ML
- 2. Naive Bayes Classification
  - a. Application to Spam Filtering
- 3. Decision Trees video #4
- 4. (Evaluation
- 5. Unsupervised Learning)
- 6. Neural Networks
  - a. Perceptrons
  - b. Multi Layered Neural Networks