COMP 472: Artificial Intelligence
Machine Learning
Naive Bayes Classification
Application to Spam Filtering

- Russell & Norvig: Sections 12.2 to 12.6

Today

1. Introduction to ML
2. Naïve 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

$$H_{NB} = \underset{H_i}{\operatorname{argmax}} \frac{P(H_i) \times P(E \mid H_i)}{P(E)} = \underset{H_i}{\operatorname{argmax}}\ P(H_i) \times P(E \mid H_i) = \underset{H_i}{\operatorname{argmax}}\ P(H_i) \times P(< a_1, a_2, a_3, \ldots, a_n > \mid H_i) = \underset{H_i}{\operatorname{argmax}}\ P(H_i) \times \prod_{j=1}^{n} P(a_j \mid H_i)$$

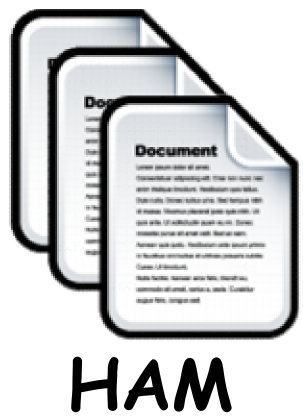
$$\boxed{H_{NB} = \underset{H_i}{\operatorname{argmax}}\ P(H_i) \times \prod_{j=1}^{n} P(a_j \mid H_i)}$$

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,…
- Given

  给你一组事先分好类别的邮件，可以使spam/ham,也可以是sports/recreation/politics这种
  - training set of documents already classified into the correct category

垃圾邮件



好邮件

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

有个1000字10000字都是很正常的

<airplane=0, banana=1, cat=5, duck=4, ..., zoo=0, class=SPAM>

<airplane=2, banana=0, cat=0, duck=8, ..., zoo=3, class=SPAM>

...

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

<airplane=1, banana=3, cat=5, duck=0, ..., zoo=6, class=HAM> 但是这种计算频率的严格上来说只能算是multinomial naive bayes classifier

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

标准的naive bayes classifier只用10表示是否出现过这个word

Naïve Bayes Algorithm

// 1. training

for all classes c_i // ex. ham or spam 决定conditional probability

for all words w_j **in the vocabulary**

compute $P(w_j \mid c_i) = \dfrac{count(w_j, c_i)}{\sum_j count(w_j, c_i)}$ ci的例子看第8页，总之就是不是挑选所有的，而是挑选特定H

for all classes c_i

compute $P(c_i) = \dfrac{\overset{\text{spam的数量}}{count(documents\ in\ c_i)}}{\underset{\text{training set的总数量}}{count(all\ documents)}}$ 决定prior probability

P(ham) P(spam)

// 2. testing a new document D

for all classes c_i // ex. ham or spam

score(c_i) = $P(c_i)$ 先选好一个priority假设

for all words w_j **in the D**

score(c_i) = **score**(c_i) x $P(w_j \mid c_i)$

hypothesis的几率连乘在ci hypothesis下各种条件的几率

choose c^* = **with the greatest score**(c_i)

	w_1	w_2	w_3	w_4	w_5	w_6
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

- Dataset

training set

  - c1: SPAM
    - doc1: "cheap meds for sale"
    - doc2: "click here for the best meds"
    - doc3: "book your trip"

SPAM

  - c2: HAM
    - doc4: "cheap book sale, not meds"
    - doc5: "here is the book for you"

HAM

- Question:
  - doc6: “the cheap book”
  - should it be classified as HAM or SPAM?

Example 1

Assume

vocabulary = {best, book, cheap, sale, trip, meds}

If not in vocabulary, ignore word

1. Training:

conditionality

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

prior

- P(SPAM) = 3/5 P(HAM) = 2/5

注意prior分母是总共的，而conditionality的分母是对应的class,这样bayes公式才有意义

2. Testing: "the cheap book"

- Score(HAM)= P(HAM) x P(cheap|HAM) x P(book|HAM)
- Score(SPAM)= P(SPAM) x P(cheap|SPAM) x P(book|SPAM)

HAM的几率 × cheap的几率 × book的几率

Be Careful: Smooth Probabilities

- normally: $$P(w_i \mid c_j) = \frac{(\text{frequency of } w_i \text{ in } c_j)}{\text{total number of words in } c_j}$$
- what if we have a $P(w_i|c_j) = 0$...?
  - ex. the word "dumbo" never appeared in the class SPAM?
  - then P("dumbo"| SPAM) = 0

有的字符只出现过0次，我们如果连乘就没意义了

- 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)

additive smoothing

  - ex: add-1 smoothing:

对于每一个出现频率都add1,但实际上1有点太大了有时候结果不是很好，0.5或者更小的都是很正常的

$$P(w_i \mid c_j) = \frac{(\text{frequency of } w_i \text{ in } c_j) + 1}{\text{total number of words in } c_j + \text{size of vocabulary}}$$

我们不仅要加一个fake 1,还要加size of vocabulary(所有备选词)

Smoothing

add-1 smoothing

- Assume:
  - vocabulary V = {ball, heat, kitchen, referee, stove, the, ... }
  - |V| = 100 这个就是size of vocabulary
- Training set:

实际上orginal data set

c1: COOKING	c2: SPORTS
doc_1: ... stove... kitchen... the... heat doc_2: ... kitchen... pasta... stove... ... doc_{100000}: ... stove...heat... ball...	doc_1: ... ball... heat... doc_2: ... the... referee... player... ... doc_{75000}: goal... injury ...

我们假装:ball:1,heat:1.............100个

也就是说在原来的基础上加上了100extra word

original+我们的100extra word

Be Careful: Use Logs

- if we really do the product of probabilities…
  - $\text{argmax}_{cj}\ P(c_j) \prod P(w_i|c_j)$ 但是实际上连乘0.xx，会导致数字飞快下降
  - we soon have numerical underflow…
  - ex: 0.01 x 0.02 x 0.05 x …

所以我们用连加Log来代替，我们只追求比较大小，用连加log不影响the ranking of hypothesis结果

- so instead, we add the log of the probs
  - $\text{argmax}_{cj}\ \log(P(c_j)) + \sum \log(P(w_i|c))$
  - ex: log(0.01) + log(0.02) + log(0.05) + …

结果是负数，假设结果是-3，另外结果是-4，还是我大，我更有可能

log的base无所谓，log 2 log 3, log 4

Example 2

- Training set:

c1: COOKING	c2: SPORTS
doc_1: … stove… kitchen… the… heat doc_2: … kitchen… pasta… stove… … doc_{100000}: … stove…heat… ball…	doc_1: … ball… heat… doc_2: … the… referee… player… … doc_{75000}: goal… injury …

- Assume:
  - vocabulary V = {ball, heat, kitchen, referee, stove, the, … }
  - |V| = 100
  - 500,000 words in Cooking
  - 300,000 words in Sports
  - 100,000 docs in Cooking
  - 75,000 docs in Sports

Example 2

- Training – Unsmoothed / Smoothed probs:
  - P(ball|COOKING) = $\frac{10,000}{500,000}$ $\frac{??}{??}$ 10000+1/5000000+100 P(ball|SPORTS) = $\frac{10,000}{300,000}$ $\frac{??}{??}$
  - P(heat|COOKING) = $\frac{255}{500,000}$ $\frac{??}{??}$ 杯水车薪，还是很小，所以用log P(heat|SPORTS) = $\frac{1,800}{300,000}$ $\frac{??}{??}$
  - P(kitchen|COOKING) = $\frac{2,600}{500,000}$ $\frac{??}{??}$ P(kitchen|SPORTS) = $\frac{0}{300,000}$ $\frac{??}{??}$
  - P(referee|COOKING) = $\frac{0}{500,000}$ $\frac{??}{??}$ P(referee|SPORTS) = $\frac{1,50}{30\ ,000}$ $\frac{??}{??}$
  - P(stove|COOKING) = $\frac{3,600}{500,000}$ $\frac{??}{??}$ P(stove|SPORTS) = $\frac{4}{300,000}$ $\frac{??}{??}$
  - P(the|COOKING) = $\frac{400,000}{500,000}$ $\frac{??}{??}$ P(the|SPORTS) = $\frac{19,000}{300,000}$ $\frac{??}{??}$
  - …
  - P(COOKING) = $\frac{100,000}{175,000}$ P(SPORTS) = $\frac{75,000}{175,000}$
- Testing: "the referee hit the ~~blue~~ ~~bird~~"
  - Score(COOKING)= log($\frac{100,000}{175,000}$) + log(P(the|COOKING)) + log(P(referee|COOKING)) + log(P(hit|COOKING)) + log(P(the|COOKING))
  - Score(SPORTS)= log($\frac{75,000}{175,000}$) + log(P(the|SPORTS)) + log(P(referee|SPORTS)) + log(P(hit|SPORTS)) + log(P(the| SPORTS))

Today

1. Introduction to ML
2. Naïve 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

Up Next

1. Introduction to ML
2. Naïve 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