Naive Bayes for Classification

Sentiment Analysis

- Sentiment analysis is the problem of extracting the author's tone from a piece of text.
- A simple example is deciding if a product review is positive or negative. Here are some short reviews of Amazon products, labelled with a 0 if they are negative or a 1 if they are positive.

```
So there is no way for me to plug it in here in the US unless I go by a
    converter. 0
Good case, Excellent value. 1
Great for the jawbone. 1
Tied to charger for conversations lasting more than 45 minutes.MAJOR
    PROBLEMS!! 0
The mic is great. 1
```

- We have three files each with 1000 labelled reviews, 500 of which are positive, 500 negative:
  - amazon reviews of products
  - yelp reviews of restaurants
  - imdb reviews of movies
- Our method will be *supervised learning* where we use a set of pre-labelled reviews to develop an algorithm that we can then apply to new, unlabelled reviews.
- Building a Spam filter is another example of this type of problem.

Bernoulli tests

- Building block: presence or absence of keywords. Each word is a "test."

	+	-	total
great	92	5	97
~**great**	408	495	903
total	500	500	1000

$$P(\mathbf{great}|+) = .184$$

$$P(\mathbf{great}) = .097$$

$$P(+|\mathbf{great}) = .948$$

$$P(+| \sim \mathbf{great}) = .452$$

$$P(+|\text{great}) = \frac{P(\text{great}|+)P(+)}{P(\text{great})}$$

	+	-	total
waste	0	14	14
~**waste**	500	486	986
total	500	500	1000

$$P(+|\mathbf{waste}) = 0$$
$$P(+| \sim \mathbf{waste}) = .51$$

Independence assumption

- We make the (false) assumption that each keyword gives an independent test.

$$P(\mathbf{great}, \mathbf{waste}|\pm) = P(\mathbf{great}|\pm)P(\mathbf{waste}|\pm)$$
$$P(\mathbf{great}, \sim \mathbf{waste}|\pm) = P(\mathbf{great}|\pm)P(\sim \mathbf{waste}|\pm)$$
$$\vdots$$

$$P(+|\mathbf{great}, \sim \mathbf{waste}) = \frac{P(\mathbf{great}|+)P(\sim \mathbf{waste}|+)P(+)}{P(\mathbf{great}, \sim \mathbf{waste})}$$

$$P(-|\mathbf{great}, \sim \mathbf{waste}) = \frac{P(\mathbf{great}|-)P(\sim \mathbf{waste}|-)P(-)}{P(\mathbf{great}, \sim \mathbf{waste})}$$

- Decision rule: compare probabilities. But only the numerator matters – this is called the "likelihood."

$$L(+|\mathbf{great}, \sim \mathbf{waste}) = (.184)(1)(.5) = .092$$

$$L(-|\mathbf{great}, \sim \mathbf{waste}) = (.01)(.028)(.5) = .00014$$

Feature vectors

- Given words $w_1, \ldots, w_k$, with probabilities $P(w_i|\pm)$, we imagine independent tests.
- The "naive" probabilities come from the training data:

$$P(w_i|\pm) = \frac{\text{number of } \pm \text{ reviews that include } w_i}{\text{total } \pm \text{ reviews}}$$

- All we need to know about a document is whether or not each of the key words appears.
- So a document can be replaced by a vector of 1/0 (called a "feature vector") where $f_i = 1$ if w_i appears, and 0 if it doesn't appear.

review $\rightsquigarrow$ [1 0 0 1 0 . . . 1]

1 in i^{th} pos $\longleftrightarrow$ w_i occurs in review

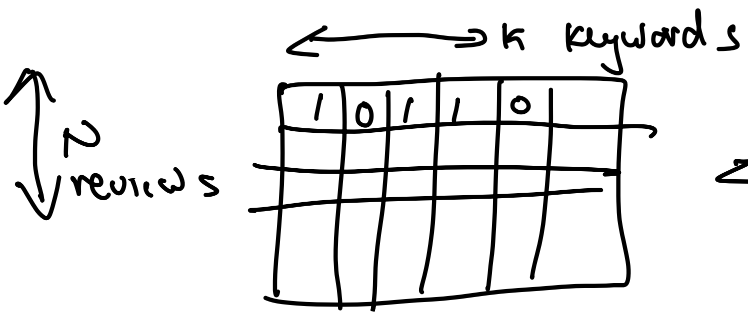
target $\longleftrightarrow$ 0, 1 $\quad$ −/+ review

Packaging up the data

- Our set of documents can be replaced by an $N \times k$ matrix with entries 0 and 1, with $x_{ij} = 1$ if the j^{th} word appears in the i^{th} document.
- Our labels form an $N \times 1$ column vector with entries 0 (for negative) or 1 (for positive) reviews.
- $Y^\intercal X$ is the sum of the rows of X corresponding to positive reviews; it is a $1 \times k$ vector whose entries count the number of times w_i occurs in a positive document.
- $(1 - Y)^\intercal X$ is a vector that counts the number of times w_i occurs in a negative document.
- $Y^\intercal Y = N_+$ is the number of positive documents, and $N_- = N - N_+$.
- The naive probabilities are

$$P_+ = \frac{1}{N_+} Y^\intercal X = \begin{bmatrix} P(w_1|+) & P(w_2|+) & \cdots & P(w_k|+) \end{bmatrix}.$$

$$P_- = \frac{1}{N_-} (1 - Y)^\intercal X = \begin{bmatrix} P(w_1|-) & P(w_2|-) & \cdots & P(w_k|-) \end{bmatrix}.$$



$\longleftrightarrow$ X matrix

1 if review is +

$$Y \quad \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\underset{1 \times N}{\begin{bmatrix} 1 & 0 & 1 & \cdots \end{bmatrix}} \underset{N \times k}{\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & \cdots \\ 1 & 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & \cdots \end{bmatrix}} = \underset{1 \times k}{}$$

sum of rows $\leftrightarrow$ + reviews

$f \longleftrightarrow$ partition counts of keywords

$[1, 0, 1, 1, \ldots]$

Likelihood

$$P(f|\pm) = \prod_{i:f_i=1} P(w_i|\pm) \prod_{i:f_i=0} (1 - P(w_i|\pm))$$ Prior

$$P(f|\pm) = \prod_{i=1}^{k} P(w_i|\pm)^{f_i} (1 - P(w_i|\pm))^{(1-f_i)}.$$ Prior

$P(f|+)$ $= P(w_1|+$

$=$ assuming $=$ #'s of $+, -$ reviews

- Log likelihood is simpler to work with

$$\log P(f|\pm) = \sum_{i=1}^{k} f_i \log P(w_i|\pm) + (1 - f_i)\log(1 - P(w_i|\pm))$$

Matrix form

$$\log P(X|\pm) = X\underbrace{(\log P_\pm)^\intercal}_{} + (1 - X)(\log(1 - P_\pm))^\intercal.$$

$X \longleftrightarrow$

$$\begin{array}{c} K \\ \begin{bmatrix} 1 & 0 & 1 & \cdots \\ 1 & 1 & 0 & \cdots \\ & & & \end{bmatrix} \\ N \times K \end{array} \quad \begin{array}{c} \begin{bmatrix} \log P(w_1|+) \\ \log P(w_2|+) \\ \vdots \end{bmatrix} \\ K \times 1 \end{array} \quad N \times 1$$

$\log P(X|\pm)$

$$\begin{bmatrix} P(f|+) \\ \\ \\ \end{bmatrix}$$

Bayes Theorem

$$\log P(\pm|f) = \log P(f|\pm) + \log P(\pm) - \log P(f)$$

$\log P(+|f) > \log P(-|f)$?

Decision rule

- a review is positive if $\log P(f|+) + \log P(+) > \log P(f|-) + \log P(-)$ and negative otherwise.